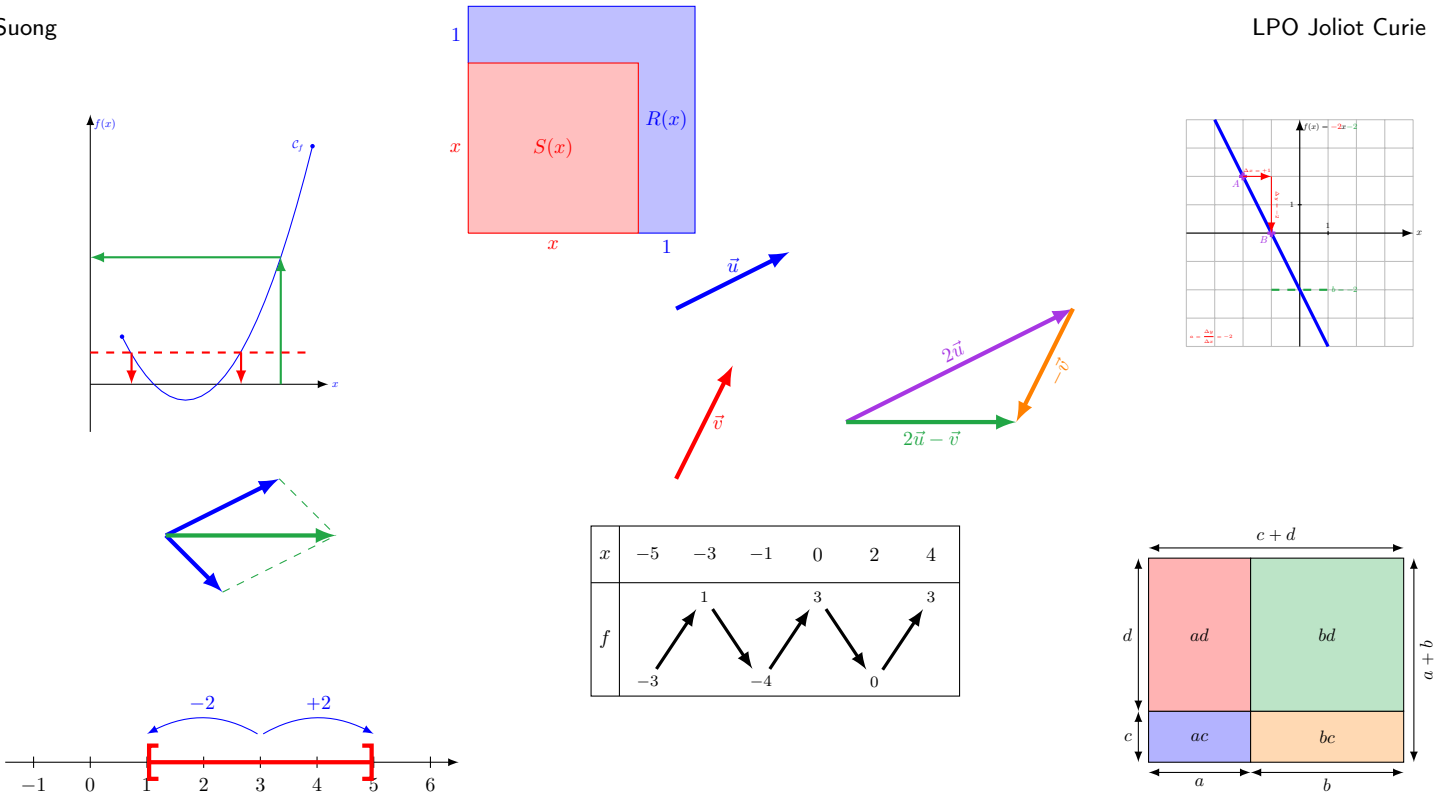
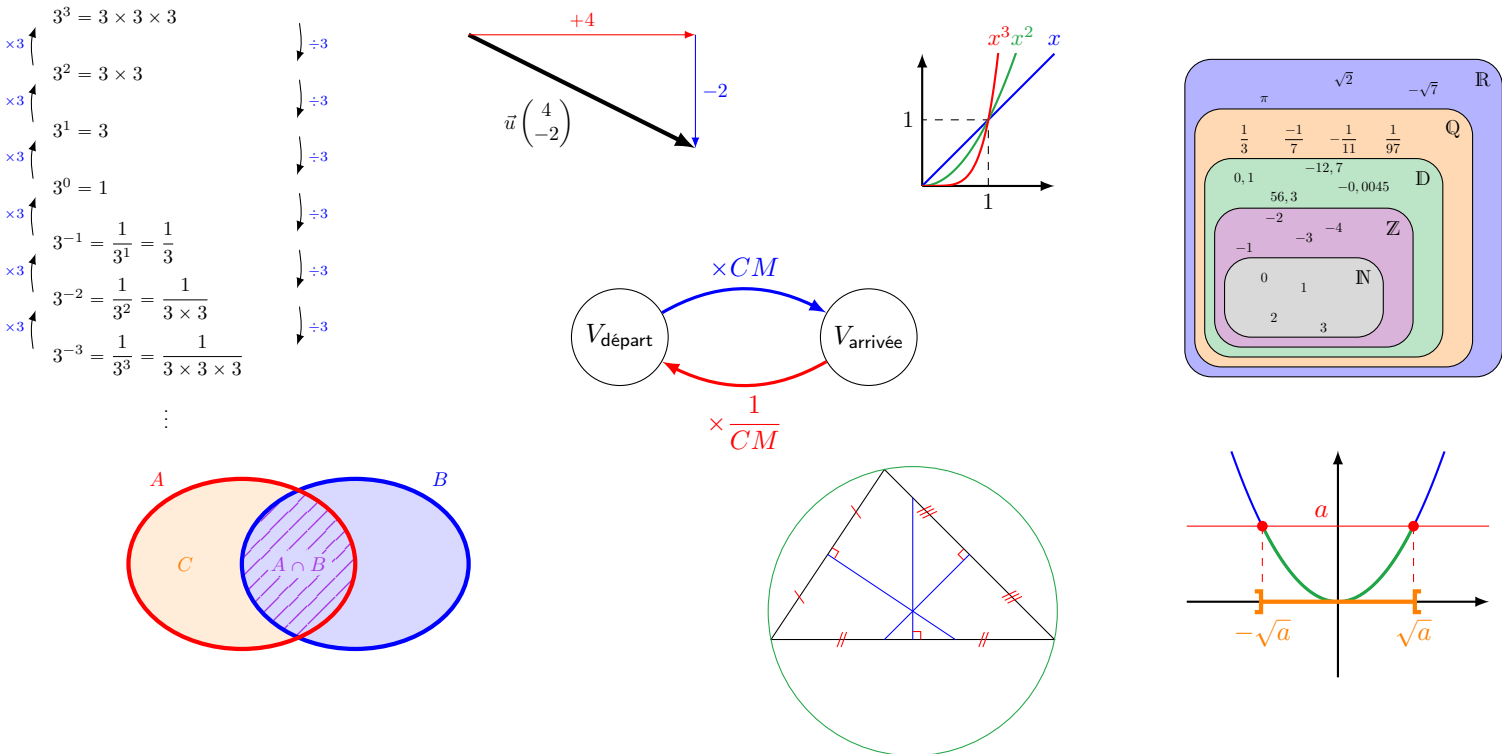




Sciences Numériques & Technologies

Seconde GT



Cours de SNT

Thème d'étude I Les données et leur traitement

1	Les données : structure, stockage, utilisations	9
1	Activités d'introduction	9
2	Données et bases de données	11
3	Stockage des données	12
4	Confidentialité des données	14
5	Activités de fin de chapitre	16
2	Formats et tailles de fichiers	18
1	Les fichiers informatiques	18
2	Les formats données structurées	21

Thème d'étude II Internet et le Web

3	Internet : un moyen pour échanger	35
1	Réseaux informatiques	35
2	Internet : un réseau de réseaux	42
4	Le Web : une manière d'échanger	61

Thème d'étude III Imagerie numérique

5	Les images numériques	65
1	Les images comme grilles de pixels	65
2	Représentations numériques de couleurs	66
3	Le fonctionnement des écrans	71
6	Autour de la photographie	75
1	Photographie chimique	75
2	Photographie électronique	76
3	Traitement d'images	88
4	Les enjeux sociétaux liés aux images	88

Thème d'étude IV Localisation et cartographie

7	Comment (se) repérer ?	99
1	Coordonnées sphériques et GPS	99
2	Les premiers outils de localisation	103
3	La localisation par satellite	105
4	Les centrales inertielles	108
5	Représentation des données géographiques	109
6	Annexes	121
8	Pourquoi (se) repérer ?	135
1	Repérer les lieux	135
2	Repérer les données	135
9	Les algorithmes et les positions	136
1	Les algorithmes d'itinéraire	136
2	La sensibilité des données de position	140
3	Annexes	141

Thème d'étude V Réseaux sociaux

10	Les réseaux sociaux	145
-----------	----------------------------	------------

Thème d'étude VI Objets connectés

11	Informatique embarquée	149
-----------	-------------------------------	------------

Thème d'étude VII Intelligences artificielles

12	L'histoire des IA	152
13	Les enjeux modernes	153
14	Vers un usage pertinent	154

Algorithmique et programmation

1	Algorithmique	158	4	(Bonus) Liste des mots-clés en Python . .	184
	1 Les variables informatiques	159		Exercices	185
	2 Les structures conditionnelles	160	4	Python : fonctions et modules	187
	3 Les structures itératives	161		1 Les fonctions	187
	4 Les fonctions	163		2 Les modules Python	189
	Exercices	164		3 (Bonus) Apprendre à lire les erreurs . . .	192
				Exercices	193
2	Programmer en Python	174	5	Python : boucles et conditionnement . . .	195
	1 Les interfaces	175		1 Les structures itératives	195
	2 Les logiciels	176		2 Les structures conditionnelles	196
				Exercices	199
3	Python : variables et types	178	6	Python : les listes	203
	1 Affectation en Python	178			
	2 Les différents types	179	7	Les langages du web	204
	3 Entrer du texte en Python	183			

SNT : pourquoi, et pour quoi ?

La SNT est un enseignement de culture générale. On a tendance à penser qu'elle ne s'adresse qu'aux futurs scientifiques, voire uniquement à la spécialité NSI, mais ce n'est pas le cas. Dans les sciences non informatiques, elles aident à développer une pensée structurée et rigoureuse. Les parcours littéraires peuvent exploiter les questions d'éthique. Les parcours économiques s'intéresseront aux enjeux sociétaux. Les parcours médicaux s'intéresseront aux problèmes de confidentialité et de protection des données. Les parcours non scolaires s'appuieront sur la maîtrise d'un ordinateur, d'un navigateur, d'une suite bureautique. Quel que soit le parcours de formation qui sera envisagé pour les années suivantes, la SNT apporte des éléments essentiels.

Et les IA, et internet en général ? À une époque où l'on peut accéder à n'importe quelle information en quelques secondes *via* nos smartphones, et où tout le monde peut avoir accès à un modèle de langage pouvant vraisemblablement donner toutes les réponses voulues en un éclair, quel intérêt d'une discipline culturelle, quel intérêt des professeurs, qui répondent aux questions des élèves ?

L'intérêt des professeurs, c'est de fournir une source d'information fiable. D'innombrables ressources sur le Web, et maintenant les IA sont capables d'expliquer énormément de choses en quelques minutes, voire secondes, et de manière de plus en plus satisfaisante. Mais les IA ne sont pas fiables, car on ne peut pas vérifier les sources de leurs informations, et toutes les sources du Web ne le sont pas nécessairement non plus. Faut-il bannir YouTube, TikTok, et surtout les IA des méthodes d'apprentissage ? Au-delà d'une question écologique pour cette dernière, la réponse est non. L'intérêt d'une discipline culturelle se trouve là : pour éviter de croire les paroles de n'importe quelle source et de finir par apprendre des choses fausses, on peut, et on doit apprendre un minimum de notion dans un peu tous les domaines, pour avoir suffisamment de connaissances pour se dire "tiens, cette information n'est pas vraisemblable, je vais la vérifier". Pour résumer, tout apprentissage culturel a vocation à développer l'esprit critique. Voici une analogie : vous discutez avec un interlocuteur dans une autre langue que vous ne connaissez pas. Vous recevez pour cela l'aide d'un traducteur : comment vous assurer que votre traducteur ne déforme pas complètement vos propos et ceux de votre interlocuteur ? Une connaissance minimale de l'autre langue vous permettrait de repérer des points clés de la vraisemblance de sa traduction.

Thème d'étude I

Les données et leur traitement

Programme

Les données structurées et leur traitement

Introduction

Les données constituent la matière première de toute activité numérique. Afin de permettre leur réutilisation, il est nécessaire de les conserver de manière persistante. Les structurer correctement garantit que l'on puisse les exploiter facilement pour produire de l'information. Cependant, les données non structurées peuvent aussi être exploitées, par exemple par les moteurs de recherche.

Repères historiques

- ▶ 1930 : utilisation des cartes perforées, premier support de stockage de données ;
- ▶ 1956 : invention du disque dur permettant de stocker de plus grandes quantités de données, avec un accès de plus en plus rapide ;
- ▶ 1970 : invention du modèle relationnel (E. L. Codd) pour la structuration et l'indexation des bases de données ;
- ▶ 1979 : création du premier tableur, VisiCalc ;
- ▶ 2009 : Open Government Initiative du président Obama ;
- ▶ 2013 : charte du G8 pour l'ouverture des données publiques.

Les données et l'information

Une donnée est une valeur décrivant un objet, une personne, un événement digne d'intérêt pour celui qui choisit de la conserver. Par exemple, le numéro de téléphone d'un contact est une donnée. Plusieurs descripteurs peuvent être utiles pour décrire un même objet (par exemple des descripteurs permettant de caractériser un contact : nom, prénom, adresse et numéro de téléphone).

Une collection regroupe des objets partageant les mêmes descripteurs (par exemple, la collection des contacts d'un carnet d'adresses). La structure de table permet de présenter une collection : les objets en ligne, les descripteurs en colonne et les données à l'intersection. Les données sont alors dites structurées.

Pour assurer la persistance des données, ces dernières sont stockées dans des fichiers. Le format CSV (Comma Separated Values, les données avec des séparateurs) est un format de fichier simple permettant d'enregistrer une table. À tout fichier sont associées des métadonnées qui permettent d'en décrire le contenu. Ces métadonnées varient selon le type de fichier (date et coordonnées de géolocalisation d'une photographie, auteur et titre d'un fichier texte, etc.).

Les données comme les métadonnées peuvent être capturées et enregistrées par un dispositif matériel ou bien renseignées par un humain. Elles sont de différents types (numériques, textes, dates) et peuvent être traitées différemment (calcul, tri, affichage, etc.). Certaines collections typiques sont utilisées dans des applications et des formats standardisés leur sont associés : par exemple le format ouvert vCard (extension .vfc) pour une collection de contacts.

Une base de données regroupe plusieurs collections de données reliées entre elles. Par exemple, la base de données d'une bibliothèque conserve les données sur les livres, les abonnés et les emprunts effectués.

Les algorithmes et les programmes

La recherche dans des données structurées a d'abord été effectuée selon une indexation préalable faite par l'homme. Des algorithmes ont ensuite permis d'automatiser l'indexation à partir de textes, d'images ou de sons.

Une table de données peut faire l'objet de différentes opérations : rechercher une information précise dans la collection, trier la collection sur une ou plusieurs propriétés, filtrer la collection selon un ou plusieurs tests sur les valeurs des descripteurs, effectuer des calculs, mettre en forme les informations produites pour une visualisation par les utilisateurs.

La recherche dans une base comportant plusieurs collections peut aussi croiser des collections différentes sur un descripteur commun ou comparable.

Les machines

Les fichiers de données sont stockés sur des supports de stockage : internes (disque dur ou SSD) ou externes (disque, clé USB), locaux ou distants (cloud). Ces supports pouvant subir des dommages entraînant des altérations ou des destructions des données, il est nécessaire de réaliser des sauvegardes.

Des recherches dans les fichiers se font à l'intérieur même des ordinateurs, soit sur la base de leurs métadonnées, soit sur la base d'une indexation (à la manière des moteurs de recherche sur le Web).

Les grandes bases de données sont souvent implémentées sur des serveurs dédiés (machines puissantes avec une importante capacité de stockage sur disques). Ces centres de données doivent être alimentés en électricité et maintenus à des températures suffisamment basses pour fonctionner correctement.

Impacts sur les pratiques humaines

L'évolution des capacités de stockage, de traitement et de diffusion des données fait qu'on assiste aujourd'hui à un phénomène de surabondance des données et au développement de nouveaux algorithmes capables de les exploiter.

L'exploitation de données massives (Big Data) est en plein essor dans des domaines aussi variés que les sciences, la santé ou encore l'économie. Les conséquences sociétales sont nombreuses tant en termes de démocratie, de surveillance de masse ou encore d'exploitation des données personnelles.

Certaines de ces données sont dites ouvertes (OpenData), leurs producteurs considérant qu'il s'agit d'un bien commun. Mais on assiste aussi au développement d'un marché de la donnée où des entreprises collectent et revendent des données sans transparence pour les usagers. D'où l'importance d'un cadre juridique permettant de protéger les usagers, préoccupation à laquelle répond le règlement général sur la protection des données (RGPD).

Les centres de données (datacenter) stockent des serveurs mettant à disposition les données et des applications les exploitant. Leur fonctionnement nécessite des ressources (en eau pour le refroidissement des machines, en électricité pour leur fonctionnement, en métaux rares pour leur fabrication) et génère de la pollution (manipulation de substances dangereuses lors de la fabrication, de la destruction ou du recyclage). De ce fait, les usages numériques doivent être pensés de façon à limiter la transformation des écosystèmes (notamment le réchauffement climatique) et à protéger la santé humaine.

Contenus	Capacités attendues
Données	Définir une donnée personnelle. Identifier les principaux formats et représentations de données.
Données structurées	Identifier les différents descripteurs d'un objet. Distinguer la valeur d'une donnée de son descripteur. Utiliser un site de données ouvertes, pour sélectionner et récupérer des données.
Traitement de données structurées	Réaliser des opérations de recherche, filtre, tri ou calcul sur une ou plusieurs tables.
Métadonnées	Retrouver les métadonnées d'un fichier personnel.
Données dans le nuage (cloud)	Utiliser un support de stockage dans le nuage. Partager des fichiers, paramétrer des modes de synchronisation. Identifier les principales causes de la consommation énergétique des centres de données ainsi que leur ordre de grandeur.

Exemples d'activités

- Consulter les métadonnées de fichiers correspondant à des informations différentes et repérer celles collectées par un dispositif et celles renseignées par l'utilisateur.

-
- ▶ Télécharger des données ouvertes (sous forme d'un fichier au format CSV avec les métadonnées associées), observer les différences de traitements possibles selon le logiciel choisi pour lire le fichier : programme Python, tableur, éditeur de textes ou encore outils spécialisés en ligne.
 - ▶ Explorer les données d'un fichier CSV à l'aide d'opérations de tri et de filtre, effectuer des calculs sur ces données, réaliser une visualisation graphique des données.
 - ▶ À partir de deux tables de données ayant en commun un descripteur, montrer l'intérêt des deux tables pour éviter les redondances et les anomalies d'insertion et de suppression, réaliser un croisement des données permettant d'obtenir une nouvelle information.
 - ▶ Illustrer, par des exemples simples, la consommation énergétique induite par le traitement et le stockage des données.

1 Les données : structure, stockage, utilisations

Table des matières

1	Activités d'introduction	9
2	Données et bases de données	11
3	Stockage des données	12
4	Confidentialité des données	14
A	Les données sensibles	14
B	Les données ouvertes	14
5	Activités de fin de chapitre	16

1 – Activités d'introduction

Les activités sont tirées de <https://ressources.forge.apps.education.fr/snt/Donnees/presentation>. Vous pouvez consulter ces ressources pour trouver des corrections.

Activité 1

- 1► Déterminer 5 critères permettant à une personne extérieure d'identifier directement n'importe qui dans la classe.
- 2► Proposer plusieurs manières de présenter ces informations. Quels sont leurs avantages et inconvénients ?

Activité 2

Regardez la vidéo [MOOC SNT / Données, comment les manipuler ?](#) de l'Inria - Médiation Scientifique sur YouTube, et répondez aux questions suivantes :

- 1► Qu'est-ce que l'informatique ?
- 2► Qu'est-ce qu'une donnée ?
- 3► Pourquoi avoir plusieurs données pour définir un objet ?
- 4► Pourquoi structurer des données ?
- 5► Comment structurer des données ?
- 6► Que sont les métadonnées, et à quoi servent-elles ?
- 7► Que sont les bases de données, et à quoi servent-elles ?

Correction

- 1► Qu'est ce que l'informatique :
 - C'est une science qui traite des de la manipulation et du traitement de l'information
 - Utilise des ordinateurs et des logiciels
 - Comprend le stockage, la récupération, la transformation et la gestion des données.
- 2► Qu'est ce qu'une donnée ?

- ▶ Une donnée c'est une valeur décrivant un objet ou une information.
 - ▶ Les données peuvent être de différents type texte, nombre, dates. . .
 - ▶ Pour identifier des objets de manière unique il est nécessaire, d'avoir plusieurs données de différentes catégories.
- 3▶ Pourquoi avoir plusieurs données pour définir un objet ?
- ▶ Un seul données ne suffit généralement pas pour identifier un objet
 - ▶ Pour identifier un chanson on a par exemple besoin de son titre, son auteur, sa durée. Seul la durée ne suffirait pas
- 4▶ Pourquoi structurer les données ?
- ▶ Structurer les données est nécessaire pour organiser les information de manière a les rendre utilisable
 - ▶ Pour faciliter le traitement par des programmes
- 5▶ Comment structurer les données ?
- ▶ On regroupe généralement les données dans un tableau
 - ▶ Chaque ligne correspond à un objet
 - ▶ Chaque colonne correspond à un descripteur de ces objets
- 6▶ Que sont les métadonnées ? À quoi servent-elles ?
- ▶ Ce sont des informations qui décrivent un fichier numérique.
 - ▶ Elle servent à fournir des détails sur le fichier (origine, date, format, ...)
- 7▶ Que sont les bases de données ? À quoi servent-elles ?
- ▶ Les bases de données sont un ensemble de données structurées et organisées
 - ▶ Elle servent a gérer et manipuler efficacement des informations (tirer, analyser, récupérer les données)

Activité 3

Dans votre quotidien ou celui de vos proches, quels applications, sites ou comportements génèrent des données ?
À quoi servent ces données ?

Activité 4

Voici une liste d'événements :

- ▶ Utilisation des cartes perforées comme premier support de stockage de données :
- ▶ Invention du disque dur
- ▶ Invention du modèle relationnel pour la structuration et l'indexation des bases de données
- ▶ Loi informatique et liberté
- ▶ Création du premier tableur
- ▶ Open Gouvernement Initiative du président B. Obama
- ▶ Charte du G8 pour l'ouverture des données publiques
- ▶ Adoption du RGPD

1▶ Choisissez 5 événements dans la liste ci-dessus, et placez-les sur une frise chronologique. Pour chaque événement, vous placerez un autre événement qui s'est produit la même année (la page Wikipédia de l'année en question pourra vous être utile pour cela).

2▶ Sur une demi feuille blanche :

- ▶ Inscrivez au recto un QCM : en question, le nom d'un événement, et en réponses possibles, des dates (ou l'inverse : en question, une date, et en réponses possibles, des événements)
- ▶ Au verso, donnez la réponse et créez une mini fiche qui détaille l'événement (courte description, que s'est-il passé, pourquoi c'est important)
- ▶ Si vous avez le temps, illustrez votre carte.

2 – Données et bases de données

Définition 5

Donnée

Une *donnée* est une valeur décrivant un objet, une personne, un événement digne d'intérêt pour celui qui choisit de la conserver.

Exemple 6

Le numéro de téléphone d'un contact est une donnée. La quantité de sucre dans un gâteau est une donnée. L'adresse d'un lieu touristique est une donnée.

Définition 7

Données structurées

- ▶ Plusieurs *descripteurs* peuvent être utiles pour décrire un même objet.
- ▶ Une *collection* regroupe des objets partageant les mêmes descripteurs.
- ▶ La structure de *table (de données)* permet de présenter une collection : les objets en ligne, les descripteurs en colonne et les données à l'intersection. Les données sont alors dites *structurées*.

Descripteurs →

Prénom	Nom	Classe
Bob	Marlon	201
Carole	Holiday	216
David	Goodenough	204
Emy	Altava	216

Données {

Table de données

Exemple 8

Dans un carnet d'adresses,

- ▶ Les nom, prénom, adresse et numéro de téléphone sont des descripteurs permettant de caractériser un contact,
- ▶ L'ensemble des contacts forme une collection (de données) : ils ont tous un nom, un prénom, une adresse et un numéro de téléphone.

Définition 9

Base de données

Une *base de données* regroupe plusieurs collections de données reliées entre elles.

Exemple 10

La base de données d'une bibliothèque conserve les données sur les livres, les abonnés et les emprunts effectués.

Remarque 11

- ▶ La recherche dans des données structurées a d'abord été effectuée selon une indexation préalable faite par l'homme. Des algorithmes ont ensuite permis d'automatiser l'indexation à partir de textes, d'images ou de sons.
- ▶ La recherche dans une base comportant plusieurs collections peut aussi croiser des collections différentes sur un descripteur commun ou comparable.

Une table de données peut faire l'objet de différentes opérations : rechercher une information précise dans la collection, trier la collection sur une ou plusieurs propriétés, filtrer la collection selon un ou plusieurs tests sur les valeurs des descripteurs, effectuer des calculs, mettre en forme les informations produites pour une visualisation par les utilisateurs.

Traitement des données**Définition 12**

Traiter des données, c'est effectuer des opérations de recherche, de filtre, de tri, de calcul ou de mise en forme sur une base de données.

Métadonnée**Définition 13**

Les *métadonnées* sont des informations complémentaires sur un fichier.

Exemple 14

- ▶ Pour un document texte, les métadonnées peuvent être l'auteur du document, sa date de rédaction, ou sa taille,
- ▶ Pour une image, cela peut-être le modèle de l'appareil photo, la résolution de l'image, les coordonnées du lieu où elle a été prise,
- ▶ Pour une musique, cela peut-être le nom de l'artiste, la durée du morceau, l'album dans lequel il est inclus.

Information 15

Les métadonnées peuvent être utilisées pour rechercher, filtrer, indexer ou trier des fichiers, par exemples par type (document, image, musique, ...), par taille, par auteur, par date de création...

3 – Stockage des données

Activité 16 (Travail de recherche)

Réaliser un document texte (avec Word, Libre Office, Open Office, ...) présentant différents supports de stockage de l'information à travers l'histoire. Vous devez :

- ▶ Réaliser une page de garde avec le sujet du document, vos nom, prénom et classe,
- ▶ Pour chaque support de stockage :
 - ▶ indiquer une date ou une période d'invention, éventuellement la ou les personnes impliquées dans son invention,
 - ▶ décrire brièvement son fonctionnement,
 - ▶ lister quelques cas d'usage les plus fréquents,
 - ▶ inclure une ou plusieurs illustrations,
- ▶ Tracer à la fin de votre document une frise chronologique plaçant les différents supports présentés.

Votre document ne doit pas faire plus de 5 pages *illustrations non comprises*. Vous êtes libre d'ajouter tout élément qui vous semble pertinent. **Vous devez inclure les URL des sources de vos informations.**

Pour guider vos recherches, voici une liste non exhaustive de mots-clés à explorer :

- | | | | |
|-------------------------|------------------|--------------------|---------------------|
| ▶ Écriture | ▶ Disquette | ▶ Disque dur | ▶ Donnée numérique |
| ▶ Gramophone | ▶ CD/DVD | ▶ SSD | ▶ Donnée analogique |
| ▶ Disque phonographique | ▶ Clé USB | ▶ Casette | |
| ▶ Gravure de sillon | ▶ Carte perforée | ▶ Bande magnétique | ▶ Carte SD |

✨ L'utilisation des intelligences artificielles est tolérée dans la mesure où vous n'en copiez pas directement les résultats. Reformulez ses réponses avec vos propres mots. **Attention, l'IA n'est pas une source valide, et ne se substitue pas à des recherches supplémentaires.**

👤 Vous avez le droit de recevoir de l'aide, mais votre document doit vous rester propre et unique. Votre document ne doit pas avoir strictement le même texte ni être présenté de la même manière qu'un autre.

Information 17**Lieux et accessibilité**

Les données sont stockées sur différents supports :

- ▶ soit *internes*, c'est-à-dire à l'intérieur de la machine qui exploite cette donnée, comme un disque dur ou un SSD,
- ▶ soit *externes*, c'est-à-dire dans un support qui ne fait pas partie de la machine, comme une clé USB ou une carte SD.

Les données peuvent être accessibles :

- ▶ *localement*, c'est-à-dire depuis la machine qui les contient,
- ▶ *à distance*, c'est-à-dire en les récupérant directement des données stockées dans une autre machine accessible *via* Internet (on parle alors de *cloud* de données, ou de stockage *en ligne*).

Exercice 18

Donner, si possible, plusieurs exemples de données qui complètent les cases du tableau :

	Interne	Externe
Local		
Distant		

Remarque 19

Les supports pouvant subir des dommages entraînant des altérations ou des destructions des données, il est nécessaire de réaliser des sauvegardes sur plusieurs supports.

Remarque 20

Les grandes bases de données sont souvent implémentées sur des serveurs dédiés (machines puissantes avec une importante capacité de stockage sur disques). Les *centres de données* (*datacenter* en anglais) stockent ces serveurs mettant à disposition les données et des applications les exploitant.

Leur fonctionnement nécessite des ressources (en eau pour le refroidissement des machines, en électricité pour leur fonctionnement, en métaux rares pour leur fabrication) et génère de la pollution (manipulation de substances dangereuses lors de la fabrication, de la destruction ou du recyclage). De ce fait, les usages numériques doivent être pensés de façon à limiter la transformation des écosystèmes (notamment le dérèglement climatique) et à protéger la santé humaine.

Exercice 21 (Travail de recherche)

Chercher un ordre de grandeur de la consommation d'un centre de données, comparée à la consommation annuelle d'un certain nombre d'habitants.

4 – Confidentialité des données

A – Les données sensibles

Définition 22

Donnée personnelle

Une *donnée personnelle* est une donnée sur un individu identifié (on sait à qui appartient la donnée) ou identifiable (la donnée permet de retrouver l'individu).

L'évolution des capacités de stockage, de traitement et de diffusion des données fait qu'on assiste aujourd'hui à un phénomène de surabondance des données et au développement de nouveaux algorithmes capables de les exploiter.

L'exploitation de données massives (*Big Data* en anglais) est en plein essor dans des domaines aussi variés que les sciences, la santé ou encore l'économie. Les conséquences sociétales sont nombreuses tant en termes de démocratie, de surveillance de masse ou encore d'exploitation des données personnelles.

Mais on assiste aussi au développement d'un marché de la donnée où des entreprises collectent et revendent des données sans transparence pour les usagers. D'où l'importance d'un cadre juridique permettant de protéger les usagers, préoccupation à laquelle répond le *règlement général sur la protection des données (RGPD)*.

Information 23

RGPD

Le *RGPD (Règlement Général sur la Protection des Données)* est une réglementation européenne en vigueur depuis le 25 mai 2018. Elle vise à protéger les données personnelles des individus et à harmoniser les pratiques au sein de l'Union européenne.

Ses principes fondateurs sont :

- ▶ Transparence : informer clairement la personne concernée.
- ▶ Finalité : collecter les données pour un objectif précis et légitime.
- ▶ Minimisation : ne recueillir que ce qui est strictement nécessaire.
- ▶ Sécurité : protéger les données contre tout accès ou usage non autorisé.
- ▶ Droits des personnes : Chaque individu dispose notamment du droit d'accès, de rectification, d'effacement (« droit à l'oubli »), de portabilité et d'opposition au traitement de ses données.

Lorsque nous naviguons sur Internet, de nombreux sites utilisent des traceurs (c'est souvent le cas avec certains cookies, prêtez attention à la fenêtre de consentement qui s'affiche à l'ouverture d'une page), qui enregistrent des informations sur notre activité : pages consultées, recherches effectuées, temps passé, produits regardés, localisation approximative... Ces données seuls semblent anodines, mais combinées avec celles provenant d'autres sites, elles permettent de créer un profil détaillé de chaque internaute.

Pour les entreprises, l'intérêt est avant tout commercial : ce profil détaillé permet une publicité ciblée et des recommandations personnalisées. Mais ces informations présentent également un risque : des escrocs peuvent exploiter ces informations pour créer des arnaques sur mesure, comme du *hameçonnage (phishing en anglais)* (faux messages ou sites imitant un service que nous utilisons), ou d'autres fraudes ciblées, par exemple une fausse offre commerciale adaptée à nos centres d'intérêt. Comprendre ce fonctionnement et régler les paramètres de confidentialité de nos appareils permet de réduire ces risques.

B – Les données ouvertes

Définition 24

Données ouvertes

Certaines données sont dites *ouvertes (OpenData en anglais)*, leurs producteurs considérant qu'il s'agit d'un bien commun.

Exemple 25

Il peut s'agir d'articles de presse, de publications sur les réseaux sociaux, de vidéos en ligne, de documents officiels ou encore d'images satellites.

Activité 26 (OSINT)

Regarder la vidéo « [On vous dévoile les coulisses de nos enquêtes en OSINT](#) » du journal Le Monde sur YouTube et répondre aux questions suivantes.

ATTENTION : le sujet et les images de 4 :25 à 5 :44 peuvent être choquantes. Vous n'êtes pas obligés de visionner ce segment et il n'est pas utile pour répondre aux questions.

- 1► Que signifie l'acronyme « OSINT » ?
- 2► Qu'est-ce que l'OSINT ?
- 3► Donner des exemples de sources d'informations exploitées en OSINT.
- 4► Quels éléments d'une image ou d'une vidéo peuvent donner des indications sur :
 - a► le lieu géographique ?
 - b► le jour ?
 - c► l'heure ?
- 5► Quel peut être l'intérêt l'OSINT ?
- 6► À quels dangers peut-on s'exposer lorsqu'on publie en ligne des données ?

Définition 27

L'OSINT (*Open Source Intelligence*) est l'art d'analyser des données ouvertes.

Remarque 28

L'OSINT ne relève pas du piratage : elle s'appuie uniquement sur des données accessibles légalement par n'importe qui.

Remarque 29

L'OSINT trouve plusieurs domaines d'application :

- Les journalistes l'utilisent pour vérifier l'authenticité d'images ou de témoignages, et retracer des faits,
- Les enquêteurs peuvent en tirer des indices pour localiser et dater un événement,
- Le jeu GeoGuessr fonde son principe sur l'OSINT.

L'OSINT constitue un outil précieux pour établir des faits et recouper des sources. Toutefois, il peut aussi être détourné à des fins malveillantes : diffusion de fausses informations, atteintes à la vie privée (*doxxing* en anglais), ou exploitation abusive de données personnelles. Sa pratique exige donc un usage responsable et un esprit critique.

Une publication apparemment anodine sur les réseaux sociaux peut révéler des informations très précises. Des éléments de second plan — un panneau de rue, la position des ombres, un monument identifiable ou encore les métadonnées intégrées automatiquement par un appareil photo — peuvent suffire à déterminer avec exactitude le lieu et le moment de la prise de vue. Les techniques d'OSINT permettent de croiser ces indices pour reconstituer un contexte, localiser une personne ou analyser ses habitudes, parfois à son insu. Cette réalité souligne la nécessité de maîtriser les informations que l'on rend publiques et de réfléchir aux conséquences potentielles d'une publication.

Exemple 30

La vidéo « [a fan told me i couldn't find his fishing spot](#) » sur YouTube illustre comment une simple image, supposément insignifiante, postée sur les réseaux sociaux permet de localiser précisément son auteur en quelques minutes.

5 – Activités de fin de chapitre

Activité 31

Définir les métiers du traitement des données listés ci-dessous :

- | | |
|--|---------------------------------|
| ▶ Analyste de données | ▶ Analyste financier |
| ▶ Data Scientist | ▶ Data Engineer |
| ▶ Ingénieur en traitement des données | ▶ Architecte de données |
| ▶ Analyste de business intelligence | ▶ Analyste en cybersécurité |
| ▶ Ingénieur en apprentissage automatique | ▶ Data Analyst en santé |
| ▶ Analyste de données marketing | ▶ Analyste en sciences sociales |

Correction

- ▶ Analyste de données : Analyse les données pour identifier des tendances, des modèles et des informations utiles pour la prise de décision.
- ▶ Data Scientist : Utilise des techniques avancées d'analyse de données, de machine Learning et de statistiques pour extraire des prédictions à partir des données.
- ▶ Ingénieur en traitement des données : Conçoit et développe des systèmes de gestion des données, des bases de données et des pipelines de traitement des données.
- ▶ Analyste de business intelligence : Collecte, organise et analyse des données pour aider les entreprises à prendre des décisions stratégiques et à surveiller leurs performances.
- ▶ Ingénieur en apprentissage automatique : Développe des modèles d'apprentissage automatique pour automatiser des tâches et créer des systèmes intelligents.
- ▶ Analyste de données marketing : Analyse les données marketing pour évaluer l'efficacité des campagnes, identifier les segments de clientèle et améliorer les stratégies marketing.
- ▶ Analyste financier : Utilise les données financières pour évaluer les performances de l'entreprise, prévoir les tendances économiques et recommander des stratégies d'investissement.
- ▶ Data Engineer : Gère l'infrastructure de données, construit et optimise les pipelines de données et assure la qualité des données pour les analyses ultérieures.
- ▶ Architecte de données : Conçoit l'architecture des systèmes de données, définit les flux de données et garantit la cohérence et l'intégrité des données.
- ▶ Analyste en cybersécurité : Analyse les données de sécurité pour détecter et prévenir les menaces, les attaques et les vulnérabilités dans les systèmes informatiques.
- ▶ Data Analyst en santé : Analyse les données médicales pour identifier des modèles de santé, améliorer les soins aux patients et prendre des décisions éclairées en matière de santé.
- ▶ Analyste en sciences sociales : Utilise les données pour étudier les comportements humains, les tendances sociales et les modèles sociétaux

Activité 32

Voici une collection d'expressions :

Internet des objets (IoT)	Réplication des données	Entrée de données	Recherche de données
Cloud computing	Sélection des données	Algorithme de tri	Normalisation des données
Archivage des données	Scraping de données	Acquisition des données	Extraction de données
Perte de données	Sécurité des données	Indexation des données	Fuite de données
Agrégation des données	Erreur humaine	Stockage en mémoire flash	Capteurs
Transformation des données	Sondages	Gestion des identités et des accès (IAM)	Serveur de données
Enquêtes	Base de données distribuée	API	Structuration des données
Compression des données	Filtrage des données	Sécurité des données	Nettoyage des données
Collecte en temps réel	Violation de la vie privée	Conformité réglementaire	Piratage informatique

En effectuant des recherches, classer ces expressions entre les quatre catégories :

- COLLECTE ► TRI ► STOCKAGE ► RISQUES

(Profitez du fait d'être en groupe pour vous répartir le travail !)

2 Formats et tailles de fichiers

Une partie des activités est inspirée de [d'un parcours Algorea](#). Vous pouvez le consulter en parallèle de ce cours.

1 – Les fichiers informatiques

A – Les formats de fichiers

Activité 1 (Classez les langues)

- 1► Vous avez 20 textes à disposition. Classez les textes selon la langue dans laquelle ils sont écrits. Est-ce simple ? Expliquer pourquoi.
- 2► Vous disposez de 20 autres textes, mais cette fois-ci une étiquette indique la langue dans laquelle le texte est écrit. Classez les textes selon leur langue. Pourquoi est-ce plus simple que précédemment ?
- 3► Dans quelle situation est-il plus simple de traduire les textes ? Expliquer pourquoi et comment.

Information 2

Le *format* d'un fichier est une indication sur la manière de lire ce fichier : est-ce que c'est un document, une image, ... avec quel logiciel faut-il l'ouvrir...

L'*extension* d'un nom de fichier indique à l'ordinateur le format de ce fichier. Il s'agit d'un point suivi de quelques lettres ou chiffres (souvent 3 ou 4) après le nom du fichier.

`<nom du fichier> . <extension>`

Exemple 3

- Extensions correspondant à des images : .jpg, .png, .webp, ...
- Extensions correspondant à des documents : .pdf, .odt, .docx, ...
- Extensions correspondant à des sons : .wav, .mp3, .ogg, ...
- Extensions correspondant à des vidéos : .mp4, .avi, .m4v, ...

Activité 4 (Erreur de traduction)

Essayez d'ouvrir une image avec le Bloc Notes.

- 1► Qu'observez-vous ?
- 2► Essayez de modifier un caractère, n'importe lequel. Sauvegardez et ouvrez l'image. Qu'observez-vous ?
- 3► Essayez d'expliquer la situation avec vos propres mots. Pourquoi après avoir modifié le fichier ainsi, l'image a-t-elle été corrompue ?

Information 5

L'intérêt des formats de données réside dans l'*interopérabilité* : en s'accordant sur une même « langue » (un même format), les différents appareils peuvent partager des fichiers et les comprendre.

Sans cette normalisation, les appareils ne peuvent pas savoir comment lire les fichiers, ou alors les liront avec des erreurs.

Remarque 6

Comme abordé dans l'exemple précédent, il existe plusieurs formats, et donc plusieurs extensions, pour un même type de fichier. Le choix d'un format dépend de l'usage que l'on fait du fichier.

Exemple 7

Pour une image :

- ▶ Le format JPEG (*Joint Photographic Experts Group*, extensions `.jpg` ou `.jpeg`) permet de compresser les images pour les rendre moins gourmandes en espace de stockage, en sacrifiant quelques détails de l'image.
- ▶ Le format PNG (*Portable Network Graphics*, extension `.png`), au contraire, conserve l'intégrité de l'image, au prix d'un volume de stockage plus conséquent.

Exemple 8

Pour un document textuel :

- ▶ Un fichier texte (extension `.txt`) permet d'écrire du texte simple, sans mise en forme (pas de gras, pas de soulignage, pas de couleur, ...). Cela consomme peu d'espace de stockage, mais ne permet que de prendre des notes sommaires.
- ▶ Un document est sauvegardé dans un format relatif au logiciel de traitement de texte qui permet de l'éditer. Par exemple, `.docx` pour Microsoft Word, `.pages` pour Pages, `.odf` (*OpenDocument Format*) soutenu par OpenOffice, ou `.odt` soutenu par LibreOffice, sont les principaux formats utilisés par les logiciels de traitement de texte. Ils permettent de construire des documents complexes, avec du texte avec mise en forme, des images, des formes...
- ▶ Le format PDF (*Portable Document Format*, extension `.pdf`) permet de partager des documents complexes avec l'assurance qu'ils seront bien affichés à la réception, mais au prix de ne plus être modifiables.

Remarque 9

Les formats `.docx` ou `.webp` par exemple, sont des formats dits *propriétaires*, ou *fermés* : leur définition et leurs règles de lecture sont contrôlées par une entité unique.

Par opposition, les formats `.jpg` ou `.pdf` par exemple sont ouverts : leur définitions et leurs règles sont contrôlées par la communauté de leurs utilisateurs.

B – Les unités de taille de fichiers

Information 10

Dans un appareil numérique, l'information est traitée et stockée en *binnaire*, c'est-à-dire à l'aide d'une succession de 0 et de 1 (respectivement un état « éteint », autrement dit le courant ne passe pas, et un état « allumé », autrement dit le courant passe).

Bit, octet

Définition 11

L'unité fondamentale d'information numérique, autrement dit l'alternative « 0 ou 1 », est appelée *bit*. Un *octet* (*byte* en anglais, prononcé « baïte ») est un regroupement de 8 bits.

Information 12

L'octet est traité comme les unités physiques (le gramme, le mètre, ...), et est souvent affublé des différents préfixes du Système international :

Unité	Symbole	Valeur
Octet	o	1o
Kilo-octet	Ko	10^3o
Méga-octet	Mo	10^6o
Giga-octet	Go	10^9o
Téra-octet	To	10^{12}o
Péta-octet	Po	10^{15}o

Remarque 13

On ne parle pas de milli-octet ou de déci-octet car, contrairement aux grammes, aux mètres, ..., 1 octet de ne pas être indéfiniment découpé en mesures plus petites.

Exemple 14

- ▶ Un cours de mathématiques complet de seconde, de plus de 200 pages, pèse un peu moins de 2Mo.
- ▶ Une photo prise avec un smartphone récent pèse un peu moins de 5Mo.
- ▶ Une musique de qualité moyenne pèse environ 1Mo par minute.
- ▶ Un vidéo filmée en 4K pèse de l'ordre de 2Go par minute.
- ▶ On estime que plus de 29 000Go de données transitent par Internet chaque seconde dans le monde.
- ▶ On estime que le volume total de données sur le Web est de l'ordre de 6×10^{22} octets.

Remarque 15 (Pour les experts)

Un bit n'a que 2 valeurs possibles. Un octet, qui contient 8 bits, a $2^8 = 256$ valeurs possibles. Ainsi, les puissances de 2 sont celles qui émergent naturellement en informatique, et non les puissances de 10. Un fichier occupe toujours un nombre d'octets qui est une puissance de 2, et donc les préfixes du système international ne sont pas adaptées. Autrement dit, lorsqu'on dit qu'une image pèse 2Mo, ceci est en fait une approximation, car 12 millions n'est pas une puissance de 2.

Pour pallier ce problème, l'on a inventé de nouveaux préfixes :

Unité binaire	Symbole	Valeur
Kibi-octet	KiB	$2^{10}\text{o} = 1\,024\text{o}$
Mébi-octet	MiB	$2^{20}\text{o} = 1\,048\,576\text{o}$
Gibi-octet	GiB	$2^{30}\text{o} \approx 1,07\text{Go}$

2 – Les formats données structurées

A – Les formats CSV et TSV

Activité 16 (Relevé de notes)

- 1► Dans son carnet de notes, un professeur a écrit :

NOTES DU 1ER TRIMESTRE

À la 1ère évaluation, Bob a eu 40, David a eu 55, Alice a eu 60 et Carole a eu 50.

À la 2e évaluation, Carole a eu 80, Bob a eu 50, Alice a eu 65 et David a eu 55.

- a► Décrivez les inconvénients de cette présentation.
 - b► Proposer une manière plus pertinente de présenter les notes des élèves aux différentes évaluations du trimestre.
Rédiger un court texte expliquant comment lire votre présentation.
 - c► Expliquer les avantages de votre présentation.
- 2► a► Présenter de nouveau les notes des élèves, mais avec une contrainte : vous n'avez pas le droit de tracer des traits (autrement dit vous pouvez uniquement écrire du texte).
b► Donner un algorithme (une règle) permettant de transformer cette présentation en tableau classique.

- 3► Sur la page suivante de son carnet de notes, le professeur a écrit :

NOTES DU 2EME TRIMESTRE

À la 1ère évaluation, Alice WONDERLAND a eu 40, Bob MARLON a eu 55, Carole HOLIDAY a eu 60 et David GOODENOUGH a eu 50. À la 2e évaluation, Alice WONDERLAND a eu 80, Bob MARLON a eu 50, Carole HOLIDAY a eu 65 et David GOODENOUGH a eu 55.

- a► De même que précédemment, présenter les notes des élèves en ne traçant aucun trait.
 - b► L'algorithme formulé précédemment fonctionne-t-il toujours ? Si non, expliquer pourquoi et proposer une solution.
- 4► Sur la page suivante de son carnet de notes, le professeur a écrit :

NOTES DU 3EME TRIMESTRE

À la 1ère évaluation, Alice WONDERLAND a eu 40,5, Bob MARLON a eu 55,3, Carole HOLIDAY a eu 60 et David GOODENOUGH a eu 50,9. À la 2e évaluation, Alice WONDERLAND a eu 80,1, Bob MARLON a eu 50,6, Carole HOLIDAY a eu 65,7 et David GOODENOUGH a eu 55.

- a► De même que précédemment, présenter les notes des élèves en ne traçant aucun trait.
- b► L'algorithme formulé précédemment fonctionne-t-il toujours ? Si non, expliquer pourquoi et proposer une solution.

Information 17

Les formats CSV (*Comma-separated Values*) et TSV (*Tab-separated Values*) servent à représenter des *tableaux* de données.

Sur la première ligne se trouvent les descripteurs, et sur chacune des suivantes les données associées à chaque objet. Les éléments d'une même ligne sont séparés par des *virgules* ou des *points-virgules* dans un fichier CSV, et par des *tabulations*⁽¹⁾ dans un fichier TSV.

(1). Une *tabulation* est un caractère représenté par un espace large. La touche Tabulation est en général au-dessus de la touche Verr. maj.

Exemple 18

Tableau de données

Article	Réserve	Prix
Cookie	50	1
Éclair	30	2
Flan	3	15

Fichier CSV correspondant

```
Article,Réserve,Prix
Cookie,50,1
Éclair,30,2
Flan,3,15
```

Fichier TSV correspondant

Article	Réserve	Prix
Cookie	50	1
Éclair	30	2
Flan	3	15

Remarque 19

Dans un fichier CSV, si une donnée contient une virgule (par exemple un nombre décimal), alors on l'encadre entre guillemets. Si une donnée contient des guillemets, alors on l'encadre également entre guillemets. On peut toujours délimiter une donnée entre guillemets dans le doute même si ce n'est pas nécessaire.

Exemple 20

Tableau de données

Titre	Prix (€)
Le Seigneur des agneaux	32,50
"Hairy" Potter	57,99
Le Mouton de M. Seguin	12,30

Fichier CSV correspondant

```
Titre,Prix (€)
Le Seigneur des agneaux,"0,50"
""Hairy" Potter","1,40"
Le Mouton de M. Seguin,"2,25"
```

Fichier TSV correspondant

Titre	"Prix (€)"
"Le Seigneur des agneaux"	0,50
""Hairy" Potter"	1,40
"Le Mouton de M. Seguin"	2,25

Exercice 21

1► Traduire le tableau en un fichier CSV :

Candidat	Épreuve n°1	Épreuve n°2	Mention
Alice	5,6	17	Sans mention
Bob	12	19	Bien
Carole	20	17	Très bien
David	20	0	Sans mention

2► Traduire le fichier CSV suivant en tableau :

```
Date,Activité,Dépense
"14/03/15","Restaurant","56,89"
"15/03/15","Bowling","23,18"
"16/03/15","Restaurant","45,67"
```

Remarque 22

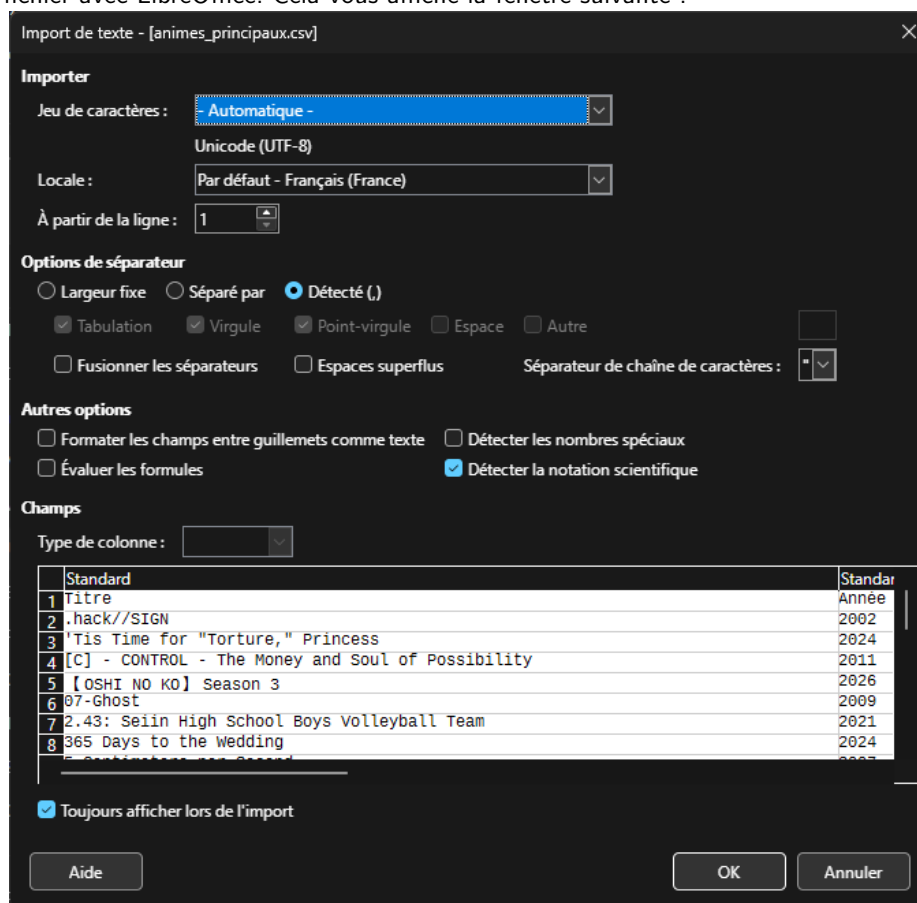
Le format TSV et le format CSV remplissent les mêmes fonctions, mais le format TSV est plus flexible (les guillemets sont moins souvent nécessaires), et peut être plus lisible humainement.

Remarque 23

Le Bloc Notes permet de lire et éditer un fichier CSV, mais cela est peu lisible pour un être humain. Un fichier CSV représentant un tableau de données, un logiciel tableur est plus adapté.

Activité 24 (Animés populaires)**Partie A : Ouverture des données**

- 1► Rendez-vous sur <https://esuong-maths.fr/snt-activites.html> et téléchargez le fichier de l'activité.
- 2► Quelle est l'extension du fichier ?
- 3► Ouvrez le fichier avec le Bloc Notes ou Notepad.
 - a► Lisez la première ligne : quelles informations sur chaque animé semble contenir ce fichier ?
 - b► Essayez de trouver l'animé le plus ancien. Est-ce facile ? Expliquez pourquoi.
 - c► Quel symbole semble séparer les différentes informations sur une même ligne ?
- 4► Ouvrez le fichier avec LibreOffice. Cela vous affiche la fenêtre suivante :



Cliquez sur « OK ». Comment s'appelle le type de logiciel, ou de document affiché par LibreOffice ?

Partie B : Trier les données

- 1► Combien y a-t-il d'animés référencés ?
- 2► Surlignez les colonnes A à H (cliquez, maintenez et glissez le long des lettres des colonnes, tout en haut). Dans la barre de navigation, allez dans Données, puis cliquez sur AutoFiltre. À quoi servent les petites flèches qui sont apparues ?
- 3►
 - a► À l'aide de la petite flèche de l'AutoFiltre, triez les animés selon leur date de parution, du plus ancien au plus récent.
 - b► Surlignez les lignes 35 à 42 (cliquez, maintenez et glissez le long des numéros des lignes, tout à gauche). Dans la barre de navigation, allez dans Données, puis cliquez sur Trier. Une fenêtre apparaît.
 - c► Dans Clé de tri 1, sélectionnez la colonne E, et cliquez sur OK. Que s'est-il passé ? En déduire l'animé le mieux noté de 1993.
 - d► Répétez l'opération : dans Clé de tri 1, sélectionnez cette fois-ci la colonne F. Quel est l'animé le plus populaire de 1993 ?

Partie C : Exploitation des données

En vous servant des techniques apprises dans la partie B, **et selon ce que l'on peut lire directement dans la base de données**, répondre aux questions suivantes :

- 1► Quel est l'animé le plus populaire tiré d'une œuvre de :
 - a► Yoshihiro Togashi ?
 - b► Jun Mochizuki
- 2► Quel est l'animé le plus populaire du studio :
 - a► A-1 Pictures ?
 - b► CloverWorks ?
- 3► Quel est le troisième animé le plus populaire du studio MAPPA ?
- 4► Quel est l'animé :
 - a► le moins populaire ?
 - b► le plus populaire ?
- 5► Quel est l'animé :
 - a► le moins bien noté ?
 - b► le plus populaire parmi les mieux notés ?
- 6► Quel est l'animé le plus vieux ?
- 7► Quel est l'animé le plus ancien de :
 - a► Hajime Yatate ?
 - b► Mamoru Hosoda ?
 - c► Yasuhiro Nightow ?
- 8► Quel est l'animé le plus populaire des années :
 - a► 1980 à 1989 ?
 - b► 1990 à 1999 ?
 - c► 2000 à 2009 ?
 - d► 2020 à 2025 ?
- 9► Parmi les travaux du Studio Ghibli, l'œuvre la mieux notée est-elle la plus populaire ?
Si oui, préciser le titre de l'œuvre, si non, préciser les titres des deux œuvres concernées.
- 10► Parmi les travaux du Studio Pierrot, l'œuvre la mieux notée est-elle la plus populaire ?
Si oui, préciser le titre de l'œuvre, si non, préciser les titres des deux œuvres concernées.
- 11►
 - a► Parmi les films (les animés ayant 1 épisode), lequel est le plus populaire ?
 - b► Quel est l'animé avec le plus d'épisodes ?

- 12►
 - a► En triant la base de données de manière pertinente, déterminer le nombre d'animés faisant partie du genre "Action".
 - b► Pourquoi la méthode de tri utilisée à la question ne peut pas s'appliquer au genre "Comédie" ?
 - c► Proposer une méthode alternative pour déterminer le nombre d'animés de genre "Comédie".
- 13► Un peu d'esprit critique : la base de données permet-elle de déterminer la popularité totale de la saga *Demon Slayer : Kemetsu no Yaiba* ? Expliquer pourquoi/comment.

B – Les formats XML et JSON

Dans la continuité de l'activité précédentes, on considérera dans cette section le jeu de données suivant :

« L'animé *My Hero Academia* » est produit depuis **2016** par le studio *bones* d'après l'œuvre originale de **Kouhei Horikoshi**. Il enregistre **784051** vues pour une note de **76** sur 100. »

Le but de cette section est de découvrir de nouveaux formats pour organiser ces données.

Activité 25 (Limites du format CSV)

- 1► Essayez de présenter ces informations dans un tableau. Quelles difficultés rencontrez-vous ?
- 2► Le format CSV est-il pertinent pour présenter ces données ? Justifier.

Information 26

Format XML

Le format XML (*Extensible Markup Language*, extension `.xml`) est un autre format permettant de structurer des données. Il présente ces dernières sous la formes de paires « clé/valeur », autrement dit d'un nom décrivant la donnée, associé à ladite donnée.

Les clés sont des *balises* qui encadrent leur valeur correspondante : `<clé>valeur</clé>`. L'ensemble du jeu de données est lui-même encadré entre deux balises.

```
<oeuvre>
  <titre>My Hero Academia</titre>

  <genres>
    <genre>Action</genre>
    <genre>Aventure</genre>
    <genre>Comédie</genre>
  </genres>

  <production>
    <date>2016</date>
    <studio>bones</studio>
    <auteur>Kouhei Horikoshi</auteur>
  </production>

  <popularite>
    <notePublic>76</notePublic>
    <vues>784051</vues>
  </popularite>
</oeuvre>
```

Exemple 27

Un tableau peut être représenté au format XML. Il y a plusieurs manières d'organiser les données, l'exemple ci-dessous n'est qu'une façon parmi tant d'autres.

Tableau de données

Article	Réserve	Prix
Cookie	50	1
Éclair	30	2
Flan	3	15

Fichier XML correspondant

```
<patisserie>
  <article>
    <nom>Cookie</nom>
    <réserve>50</réserve>
    <prix>1</prix>
  </article>
  <article>
    <nom>Éclair</nom>
    <réserve>30</réserve>
    <prix>2</prix>
  </article>
  <article>
    <nom>Flan</nom>
    <réserve>3</réserve>
    <prix>15</prix>
  </article>
</patisserie>
```

Exercice 28

1► Traduire le tableau en un fichier XML :

Candidat	Épreuve n°1	Épreuve n°2	Mention
Alice	5,6	17	Sans mention
Bob	12	19	Bien
Carole	20	17	Très bien
David	20	0	Sans mention

2► Traduire le fichier XML suivant en tableau :

```
<notes>
  <matière>
    <nom>Maths</nom>
    <moyenne>85</moyenne>
    <coefficient>3</coefficient>
  </matière>
  <matière>
    <nom>Français</nom>
    <moyenne>70</moyenne>
    <coefficient>3</coefficient>
  </matière>
  <matière>
    <nom>Philo</nom>
    <moyenne>60</moyenne>
    <coefficient>8</coefficient>
  </matière>
</notes>
```

Information 29**Format JSON**

Le format JSON (*JavaScript Object Notation*, extension `.json`) remplit les mêmes fonctions que le XML, mais sa présentation est différente, plus versatile et plus légère.

Le jeu de données est encadré entre accolades, les clés sont des textes (entre guillemets), suivi d'un deux-points et de la valeur associée. Une virgule sépare les différentes paires « clé/valeur ». Si la valeur est une liste de données, celle-ci est encadrée entre crochets. Si la valeur est elle-même un jeu de données, on ouvre une nouvelle paire d'accolades et on recommence.

```
{
  "titre": "My Hero Academia",
  "genres": ["Action", "Aventure", "Comédie"],
  "production": {
    "date": 2016,
    "studio": "bones",
    "auteur": "Kouhei Horikoshi"
  },
  "popularité": {
    "note public": 76,
    "vues": 784051
  }
}
```

Exemple 30

Un tableau peut être représenté au format JSON. Il y a plusieurs manières d'organiser les données, l'exemple ci-dessous n'est qu'une façon parmi tant d'autres.

Tableau de données

Article	Réserve	Prix
Cookie	50	1
Éclair	30	2
Flan	3	15

Fichier JSON correspondant

```
{
  "Cookie": {
    "réserve": 50,
    "prix": 1
  },
  "Éclair": {
    "réserve": 30,
    "prix": 2
  },
  "Flan": {
    "réserve": 3,
    "prix": 15
  }
}
```

Exercice 31

1► Traduire le tableau en un fichier JSON :

Candidat	Épreuve n°1	Épreuve n°2	Mention
Alice	5,6	17	Sans mention
Bob	12	19	Bien
Carole	20	17	Très bien
David	20	0	Sans mention

2► Traduire le fichier JSON suivant en tableau :

```
{
  "Alice": {
    "sport": "course",
    "temps": 37.9
  },
  "Bob": {
    "sport": "natation",
    "temps": 65.7
  },
  "Carole": {
    "sport": "vélo",
    "temps": 16.3
  }
}
```

Exercice 32

Voici un fichier JSON présentant des informations au sujet du jeu vidéo *Super Mario Galaxy*.

```
{
  "titre": "Super Mario Galaxy",
  "développement": {
    "développeur": "Nintendo EAD Tokyo",
    "éditeur": "Nintendo",
    "éditeur": "Yoshiaki Koizumi",
    "musique": {
      "compositeurs": ["Mahito Yokota", "Koji Kondo"],
      "chef d'orchestre": "Koji Haishima"
    },
    "producteurs": ["Shigeru Miyamoto", "Takao Shimizu"]
  },
  "date de parution": 2007,
  "genre": "Plates-formes",
  "consoles": ["Wii", "Nintendo Switch"]
}
```

En vous inspirant de l'exemple, créez un fichier JSON (entre 15 et 20 lignes avec une présentation similaire à l'exemple) sur le jeu *The Legend of Zelda : Breath of the Wild*.

Vous sélectionnerez les informations dans le bandeau résumé de la [page Wikipédia](#) du jeu. Vous n'êtes pas obligés de traduire toutes les informations, ni de vous restreindre aux informations dans le bandeau.

Thème d'étude II

Internet et le Web

Programme

Sciences numériques et technologie

Internet

Introduction

Grâce à sa souplesse et à son universalité, internet est devenu le moyen de communication principal entre les hommes et avec les machines.

Repères historiques

Dès les années cinquante, les ordinateurs ont été mis en réseau pour échanger des informations, mais de façon très liée aux constructeurs d'ordinateurs ou aux opérateurs téléphoniques. Les réseaux généraux indépendants des constructeurs sont nés aux États-Unis avec ArpaNet (1970) et en France avec Cyclades (1971). Cet effort a culminé avec internet, né en 1983.

Le protocole TCP/IP

Internet est défini par le protocole IP (Internet Protocol), ensemble de normes qui permettent d'identifier et de nommer de façon uniforme tous les ordinateurs ou objets qui lui sont connectés. IP est accompagné de protocoles de transmission pour transférer l'information par paquets, le principal étant TCP/IP (Transmission Control Protocol). De nature logicielle, internet s'appuie sur une grande variété de réseaux physiques où IP est implémenté. Il uniformise l'accès à tous les ordinateurs, les téléphones et les objets connectés.

Les données et l'information

Internet manipule deux types d'information : les contenus envoyés et les adresses du destinataire et de l'émetteur. Ces deux types d'information sont regroupés dans des paquets de taille fixe, de façon uniforme et indépendante du type de données transportées : texte, images, sons, vidéos, etc. Les adresses sont numériques et hiérarchiques mais l'utilisateur connaît surtout des adresses symboliques normalisées, comme wikipedia.fr. Le système DNS (Domain Name System) transforme une adresse symbolique en adresse numérique. Il est réalisé par un grand nombre d'ordinateurs répartis sur le réseau et constamment mis à jour.

Les algorithmes et les programmes

Le principal algorithme d'internet est le routage des paquets de leurs émetteurs vers leurs destinataires. Il est effectué par des machines appelées routeurs, qui échangent en permanence avec leurs voisins pour établir une carte locale de ce qu'ils voient du réseau. Chaque paquet transite par une série de routeurs, chacun l'envoyant à un autre routeur selon sa carte locale et la destination prévue. Les routeurs s'ajustent en permanence et de proche en proche quand on les ajoute au réseau ou quand un routeur voisin disparaît. Il n'y a plus besoin de carte globale, ce qui permet le routage à grande échelle.

Lors du routage, un paquet peut ne pas arriver pour deux raisons : une panne matérielle d'une ligne ou d'un routeur, ou sa destruction. Chaque paquet contient l'information d'un nombre maximal de routeurs à traverser : pour ne pas encombrer le réseau, il est détruit si ce nombre est atteint. C'est le protocole TCP qui fiabilise la communication en redemandant les paquets manquants. Il garantit que tout paquet finira par arriver, sauf panne matérielle incontournable. TCP réordonne aussi les paquets arrivés dans le désordre et diminue la congestion du réseau en gérant au mieux les redemandes. Mais ni internet ni TCP ne possèdent de garantie temporelle d'arrivée des paquets, ce qui nuit à la qualité du streaming du son ou des vidéos et de la téléconférence. En effet, dans une vidéo, on peut perdre une image isolée, mais pas le fil du temps. D'autres protocoles s'appuient sur ceux d'internet, par exemple les protocoles du Web (HTTP et HTTPS) et le protocole NTP (Network Time Protocol) qui permet de synchroniser finement les heures des ordinateurs et objets connectés.

Les machines

Réseau mondial, internet fonctionne à l'aide de routeurs, de lignes de transmissions à très hauts débits (fibres optiques) entre routeurs, de réseaux de téléphonie mobile, et de réseaux locaux. Ses protocoles étant logiciels, il peut s'appuyer sur n'importe quel réseau physique qui les implémente : 4G, Ethernet, ADSL, Wi-Fi, Bluetooth, etc. TCP/IP n'est pas implémenté dans l'infrastructure, mais dans chacun des ordinateurs connectés, et un serveur DNS est aussi un ordinateur connecté. Des mécanismes complexes assurent la continuité de la connexion, par exemple pour passer sans interruption de téléphonie 4G au Wi-Fi, ou son ubiquité, par exemple pour passer de façon invisible d'antenne à antenne avec un téléphone portable quand on voyage.

Dans les réseaux pair-à-pair s'appuyant sur internet et souvent utilisé pour le transport de vidéos, chaque ordinateur sert à la fois d'émetteur et de récepteur.

Impacts sur les pratiques humaines

Internet a fait progressivement disparaître beaucoup des moyens de communication précédents : télégramme, télex, le courrier postal pour une bonne partie, et bientôt le téléphone fixe grâce à VoIP (voix sur IP). Son trafic prévu pour 2021 est de 3 300 milliards de milliards d'octets ($3,3 \times 10^{21}$ octets).

Internet a aussi ses problèmes : absence de garantie temporelle sur l'arrivée des paquets et possibilité d'attaques par saturation en envoyant un très grand nombre de messages à un site donné, pour y provoquer un déni de service.

La neutralité du Net, présente dès l'origine du réseau, exprime l'idée que les routeurs doivent transmettre les paquets indépendamment du type de leur contenu : texte, vidéo, etc. Mais elle est constamment remise en cause par certains lobbies industriels.

Contenus	Capacités attendues
Protocole TCP/IP : paquets, routage des paquets	Distinguer le rôle des protocoles IP et TCP. Caractériser les principes du routage et ses limites. Distinguer la fiabilité de transmission et l'absence de garantie temporelle.
Adresses symboliques et serveurs DNS	Sur des exemples réels, retrouver une adresse IP à partir d'une adresse symbolique et inversement.
Réseaux pair-à-pair	Décrire l'intérêt des réseaux pair-à-pair ainsi que les usages illicites qu'on peut en faire.
Indépendance d'internet par rapport au réseau physique	Caractériser quelques types de réseaux physiques : obsolètes ou actuels, rapides ou lents, filaires ou non. Caractériser l'ordre de grandeur du trafic de données sur internet et son évolution.

Exemples d'activités

- ▶ Illustrer le fonctionnement du routage et de TCP par des activités débranchées ou à l'aide de logiciels dédiés, en tenant compte de la destruction de paquets.
- ▶ Déterminer l'adresse IP d'un équipement et l'adresse du DNS sur un réseau.
- ▶ Analyser son réseau local pour observer ce qui y est connecté.
- ▶ Suivre le chemin d'un courriel en utilisant une commande du protocole IP.

Le Web

Introduction

Le Web (toile ou réseau) désigne un système donnant accès à un ensemble de données (page, image, son, vidéo) reliées par des liens hypertextes et accessibles sur le réseau internet.

Repères historiques

- ▶ 1965 : invention et programmation du concept d'hypertexte par Ted Nelson ;

-
- ▶ 1989 : naissance au CERN par Tim Berners Lee ;
 - ▶ 1993 : mise dans le domaine public, disponibilité du premier navigateur Mosaic ;
 - ▶ 1995 : mise à disposition de technologies pour le développement de site Web interactif (langage JavaScript) et dynamique (langage PHP) ;
 - ▶ 2001 : standardisation des pages grâce au DOM (Document Object Model) ;
 - ▶ 2010 : mise à disposition de technologies pour le développement d'applications sur mobiles.

Normalisation de la présentation de l'information

Sur le Web, les textes, photos, vidéos, graphiques, sons, programmes sont exprimés et assemblés dans divers formats normalisés par un consortium mondial (W3C : World Wide Web Consortium), ce qui permet une circulation standardisée de ces informations.

Les pages Web sont écrites dans le langage de balises HTML (Hypertext Markup Language). Leur style graphique est défini dans le langage CSS (Cascading Style Sheets).

Les pages ont une adresse unique, nommée URL (Uniform Ressource Locator). Elles sont accessibles via internet en utilisant le protocole HTTP (Hypertext Transfer Protocol) ou sa version sécurisée HTTPS qui crypte les échanges. L'affichage des pages est réalisé chez l'utilisateur par un programme appelé navigateur.

Un hypertexte est un texte augmenté de renvois automatiques à des textes, des images ou des sons. Initialement, un hypertexte se restreignait à la mémoire d'un seul ordinateur. Dans une page Web, ce renvoi se fait sur n'importe quelle machine du réseau internet, par le truchement de l'adresse de la page Web du texte (URL) auquel il fait référence. La toile d'araignée construite par les liens peut être représentée sous la forme d'un graphe qui matérialise la structure du Web.

Moteurs de recherche

Les moteurs de recherche permettent de trouver des informations dans des pages dont on ne connaît pas l'adresse, voire dont on ignore l'existence. La méthode de recherche appelée référencement naturel se décompose en trois grandes activités, réalisées par les moteurs de recherche : (1) le parcours automatique du Web pour collecter les pages visitées (aspiration des pages Web effectuée par des robots) ; (2) l'analyse du contenu des pages et leur indexation sur les mots qu'elles contiennent (constitution d'un annuaire inversé qui associe à chaque terme les URL des pages où il apparaît) ; (3) la troisième activité, réalisée à chaque fois qu'un internaute fait une requête, construit une liste ordonnée des pages (classement) comportant les mots clés de la requête. Leur ordre dépend notamment de leur popularité (principe des liens), de leur pertinence (aux mots de la requête), et de l'ordre des termes de la requête.

Les concepteurs de site Web peuvent améliorer le référencement de leurs pages en choisissant bien les mots et en les plaçant à des endroits stratégiques dans les pages.

Interaction client-serveur

Le Web s'appuie sur le dialogue entre clients et serveurs. L'interaction est à l'initiative des clients (les applications qui se connectent au Web, dont les navigateurs), qui envoient des requêtes HTTP aux serveurs. Ces derniers renvoient leur résultat : des pages qu'ils ont stockées ou qu'ils créent dynamiquement en fonction de la requête formulée. Les pages reçues par les clients peuvent contenir des codes exécutables (souvent en JavaScript) qui permettent aux clients d'effectuer des traitements en accédant aux ressources de son ordinateur et en interagissant avec les serveurs.

Les applications peuvent être paramétrées pour autoriser ou interdire l'accès à des ressources locales aux programmes téléchargés par les pages.

Sécurité et confidentialité

En formulant des requêtes sur des sites Web dynamiques et en laissant des programmes s'exécuter sur sa machine, l'utilisateur prend des risques : il peut communiquer des informations personnelles à son insu à des serveurs qui en gardent une trace, à distance ou localement par des cookies, ou encore charger des pages contenant des programmes malveillants, par exemple permettant d'espionner en continu les actions de l'utilisateur. Par ailleurs, un navigateur peut garder un historique de toutes les interactions, et le laisser accessible aux sites connectés. L'utilisateur peut utiliser des services qui s'engagent à ne pas garder de traces de ses interactions, par exemple certains moteurs de recherche. Il peut aussi paramétrer son navigateur de façon à ce que celui-ci n'enregistre pas d'historique des interactions. De

fausses pages peuvent encore être utilisées pour l'hameçonnage des utilisateurs. Un nom de lien pouvant cacher une adresse Web malveillante, il faut examiner cette adresse avant de l'activer par un clic.

Impacts sur les pratiques humaines

Dans l'histoire de la communication, le Web est une révolution : il a ouvert à tous la possibilité et le droit de publier ; il permet une coopération d'une nature nouvelle entre individus et entre organisations : commerce en ligne, création et distribution de logiciels libres multi-auteurs, création d'encyclopédies mises à jour en permanence, etc. ; il devient universel pour communiquer avec les objets connectés.

Le Web permet aussi de diffuser toutes sortes d'informations dont ni la qualité, ni la pertinence, ni la véracité ne sont garanties et dont la vérification des sources n'est pas toujours facile. Il conserve des informations, parfois personnelles, accessibles partout sur de longues durées sans qu'il soit facile de les effacer, ce qui pose la question du droit à l'oubli. Il permet une exploitation de ses données, dont les conséquences sociétales sont encore difficiles à estimer : recommandation à des fins commerciales, bulles informationnelles, etc. En particulier, des moteurs de recherche permettent à certains sites d'acquérir de la visibilité sur la première page des résultats de recherche en achetant de la publicité qui apparaîtra parmi les liens promotionnels.

Contenus	Capacités attendues
Repères historiques	Connaître les étapes du développement du Web.
Notions juridiques	Connaître certaines notions juridiques (licence, droit d'auteur, droit d'usage, valeur d'un bien).
Hypertexte	Maîtriser les renvois d'un texte à différents contenus.
Langages HTML et CSS	Distinguer ce qui relève du contenu d'une page et de son style de présentation. Étudier et modifier une page HTML simple.
URL	Décomposer l'URL d'une page. Reconnaître les pages sécurisées.
Requête HTTP	Décomposer le contenu d'une requête HTTP et identifier les paramètres passés.
Modèle client/serveur	Inspecter le code d'une page hébergée par un serveur et distinguer ce qui est exécuté par le client et par le serveur.
Moteurs de recherche : principes et usages	Mener une analyse critique des résultats fournis par un moteur de recherche. Comprendre les enjeux de la publication d'informations.
Paramètres de sécurité d'un navigateur	Maîtriser les réglages les plus importants concernant la gestion des cookies, la sécurité et la confidentialité d'un navigateur. Sécuriser sa navigation en ligne et analyser les pages et fichiers.

Exemples d'activités

- ▶ Construire une page Web simple contenant des liens hypertextes, la mettre en ligne.
- ▶ Modifier une page Web existante, changer la mise en forme d'une page en modifiant son CSS. Insérer un lien dans une page Web.
- ▶ Comparer les paramétrages de différents navigateurs.
- ▶ Utiliser plusieurs moteurs de recherche, comparer les résultats et s'interroger sur la pertinence des classements.
- ▶ Réaliser à la main l'indexation de quelques textes sur quelques mots puis choisir les textes correspondant à une requête.
- ▶ Calculer la popularité d'une page à l'aide d'un graphe simple puis programmer l'algorithme.
- ▶ Paramétrer un navigateur de manière qu'il interdise l'exécution d'un programme sur le client.
- ▶ Comparer les politiques des moteurs de recherche quant à la conservation des informations sur les utilisateurs.

-
- ▶ Effacer l'historique du navigateur, consulter les cookies, paramétrer le navigateur afin qu'il ne garde pas de traces.
 - ▶ Utiliser un outil de visualisation tel que Cookieviz pour mesurer l'impact des cookies et des traqueurs lors d'une navigation.
 - ▶ Régler les paramètres de confidentialité dans son navigateur ou dans un service en ligne.

3 Internet : un moyen pour échanger

Table des matières		
1	Réseaux informatiques	35
A	Des graphes pour représenter des réseaux	35
B	Les infrastructures réseau	37
C	Modèles d'architecture	41
2	Internet : un réseau de réseaux	42
A	Vocabulaire et analogies	42
B	Des adresses pour se repérer	45
C	Des protocoles pour bien s'entendre	47

1 – Réseaux informatiques

A – Des graphes pour représenter des réseaux

Définition 1

Un *graphe* est une manière de représenter de manière schématique les relations entre différentes données. Les données sont représentés sur les *nœuds* du graphe, et les connexions entre les nœuds sont appelées des *arêtes*. Sur chaque arête, on peut ajouter un nombre appelé *pondération*, qui peut représenter une distance, un temps, un coût, ... selon la situation. On parle alors de *graphe pondéré*.

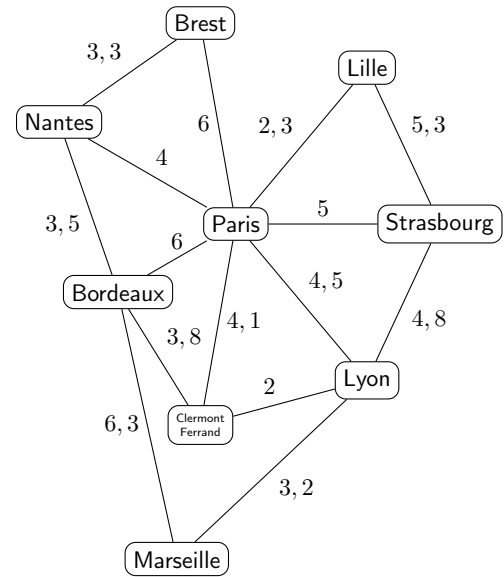
Exemple 2

Le réseau routier peut-être représenté par un graphe, dont les villes sont les nœuds, et les routes sont les arêtes. Les pondération sur les arêtes peuvent, par exemple, représenter la distance en km entre deux villes voisines.

Remarque 3

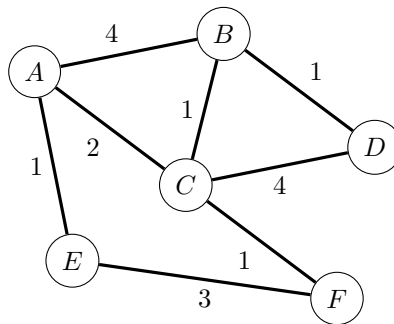
Dans l'exemple précédent, les nœuds ont été placés de manière à ce que la longueur d'une arête représente approximativement la distance de la pondération, comme une vraie carte à l'échelle.

Dans la carte ci-contre, les pondérations sont les temps de trajet en voiture, en heures. Cette fois-ci, on a arrangé le graphe de sorte à équilibrer les longueurs des arêtes, et renforcer la position centrale de Paris : la longueur d'une arête n'est donc plus représentative de sa pondération.



Activité 4

Voici un graphe dont les représentations représentent une distance, en km : par exemple, pour voyager entre A et B , il faut parcourir 4km.



- 1► a► Quelle est la distance du trajet $A \rightarrow B \rightarrow D \rightarrow C$?
b► Si on part de B , quels nœuds sont accessibles à 4km ou moins ?
- 2► Dans cette question, on part du nœud A .

- a► Déterminer le chemin le plus court pour aller en E . Faire de même vers C .
- b► Déterminer le chemin le plus court pour aller en F . Faire de même vers D .
- c► Déterminer le chemin le plus court pour aller en B .
- d► À l'aide des questions précédentes, compléter le tableau ci-contre.

Départ : A	
Destination finale	Première étape
B	
C	
D	
E	
F	

- 3► En groupe, répétez la question précédente en Départ : B , puis de C , etc. jusqu'à F . Rassembler tous les résultats dans le tableau suivant, appelé *table de routage*. (DEST : destination finale de la route, PRE : première étape de la route).

Départ : A		Départ : B		Départ : C		Départ : D		Départ : E		Départ : F	
DEST	PRE	DEST	PRE	DEST	PRE	DEST	PRE	DEST	PRE	DEST	PRE
A	×	A		A		A		A		A	
B		B	×	B		B		B		B	
C		C		C	×	C		C		C	
D		D		D		D	×	D		D	
E		E		E		E		E	×	E	
F		F		F		F		F		F	×

Remarque 5

Dans le graphe de l'activité précédente, vous avez cherché les plus courts chemins par essais et erreurs. Dans un graphe plus grands, avec des dizaines, des milliers voire des millions de nœuds, il serait impossible de le faire à la main. Des algorithmes, comme celui du mathématicien Edsger Dijkstra, permettent à des ordinateurs d'effectuer ce travail de manière efficace.

B – Les infrastructures réseau

Réseau**Définition 6**

Un *réseau* est un ensemble d'entités (des personnes, des ordinateurs, des lieux, ...) liés entre elles. Un réseau peut donc être représenté par un graphe.

Exemple 7

- ▶ Le réseau ferroviaire relie les différentes villes par des voies ferrées. Les nœuds sont les villes, les arêtes sont les chemins de fer.
- ▶ Un réseau social relie différentes personnes. Les nœuds sont ces personnes, et les arêtes sont les liens d'amitié entre les personnes.
- ▶ Le réseau électrique distribue l'électricité entre les différents bâtiments. Ces bâtiments sont les nœuds du graphe, et les câbles électriques en sont les arêtes.

Il existe de nombreux réseaux informatiques, reliant entre eux les différents appareils. Lorsqu'on appelle quelqu'un par téléphone, lorsqu'on se connecte à un casque par Bluetooth, quand on branche un ordinateur par Ethernet... on utilise différents réseaux informatiques. On distingue deux types de connexion entre deux appareils : celles où on branche un câble entre les deux machines, et les connexions sans fil.

Liaison filaire**Définition 8**

Lorsque deux appareils se connectent l'un à l'autre par l'intermédiaire d'un canal physique, on dit que leur liaison est *filaire*.

Exemple 9

- ▶ Les câbles téléphoniques, qui sont des câbles électriques en cuivre, étaient autrefois utilisés pour acheminer la voix lors des appels téléphoniques, et plus tard des informations numériques *via* l'ADSL.
- ▶ Les câbles Ethernet permettent de relier des appareils à l'échelle d'un bâtiment.
- ▶ Les fibres optiques permettent des connexions filaires longue distance.

Définition 10**Liaison hertzienne**

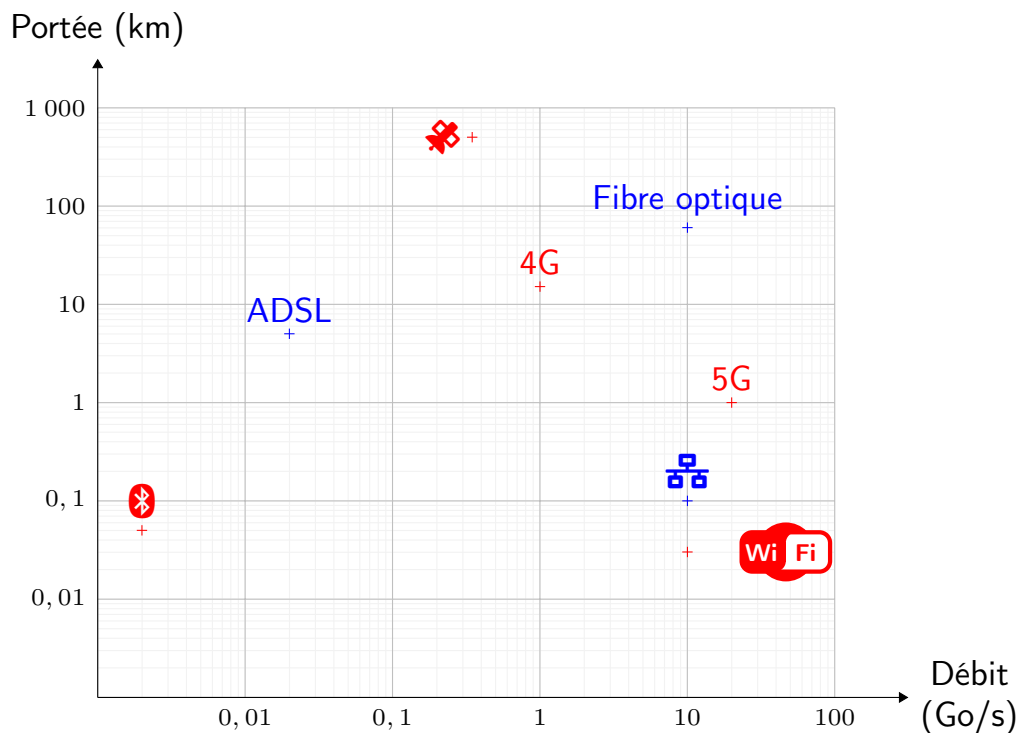
Lorsque deux appareils se connectent l'un à l'autre sans canal physique, on dit que leur liaison est *hertzienne*.

Exemple 11

- ▶ Le Bluetooth permet de relier deux appareils proches.
- ▶ Le WiFi permet de relier des appareils dans un rayon de quelques dizaines de mètres.
- ▶ Les infrastructures de téléphonie mobile (antennes 5G, 4G, ...) permettent de connecter de nombreux appareils sur des dizaines de kilomètres.
- ▶ Les liaisons satellites peuvent atteindre potentiellement n'importe quel endroit sur Terre.

Définition 12**Débit**

Le *débit* d'une connexion est une mesure de la quantité de données que cette connexion peut transmettre par unité de temps. On l'exprime en général en o/s (octet par seconde), ou en version anglaise b/s (*byte* par seconde).

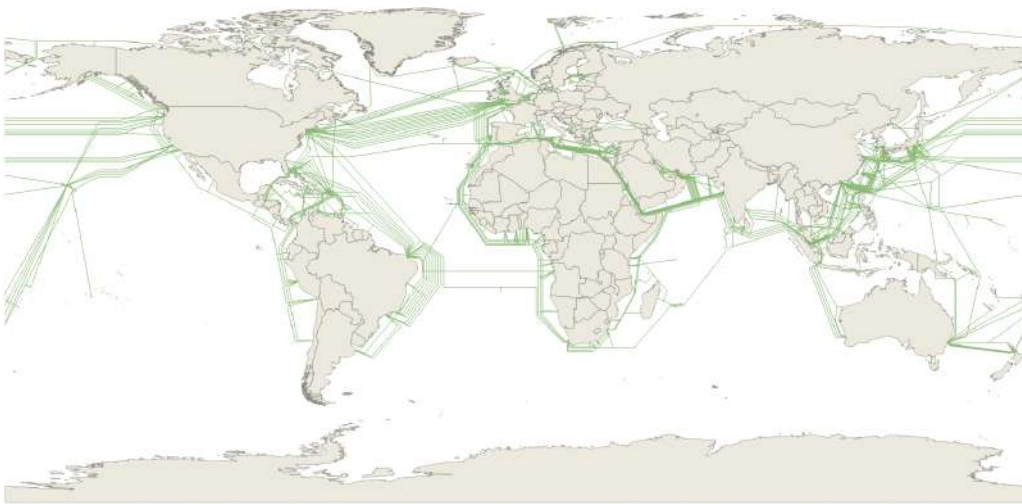
Exemple 13

Exercice 14

- 1► Exprimer en français, puis convertir en octets : 1ko, 1Mo, 1Go, 1To.
- 2► Convertir 2 345,6ko en gigaoctets.
- 3► La transmission d'une vidéo prend 2 minutes avec un débit de 1,5Mo/s. Quelle est la taille de la vidéo ?
- 4► Avant les Box Internet modernes, les foyers se connectaient au réseau Internet *via* des « Modems 56k », qui, comme leur nom l'indique, pouvaient transmettre un débit maximal de 56ko/s.
 - a► Combien de temps fallait-il pour télécharger une photo de 2Mo ?
 - b► Même question pour un jeu de 5Go.

Information 15

La majeure partie du flux de données transmis autour du globe transite par des *câbles sous-marins*, un réseau de fibres optiques protégés dans des câbles, et regroupés en *faisceaux* de câbles, déposés au fond des océans.



Réseaux de câbles sous marin. Chaque ligne représente un faisceau de câbles.

Activité 16 (Étude de documents)

À l'aide des documents et de vos connaissances, répondre aux questions suivantes.

- 1► Comment est transmis physiquement plus de 99% du trafic Internet et téléphone ?
- 2► Pourquoi y a-t-il davantage de câbles sous-marins entre les États-Unis et l'Europe, qu'entre l'Amérique du Sud et l'Afrique ?
- 3► Citez d'autres moyens de communication utilisés pour les communications par Internet.
- 4► Par combien a été multiplié le trafic Internet mondial entre 1992 et 2002 ? Entre 2002 et 2022 ?
- 5► Si la progression de 2002 à 2022 reste la même, quelle sera le trafic Internet mondial en 2042 ?
- 6► Justifiez l'affirmation suivante : « Les câbles sous-marins forment un réseau de communication indispensable au fonctionnement de la société de l'information ».
- 7► Quelles menaces pèsent sur l'infrastructure ?

DOCUMENT 1 : Sébastien Gavois. TAT-14 : le câble transatlantique qui a fait tousoter internet cette nuit. Next Impact, 20 mai 2014.

Cette nuit, plusieurs sites et services comme Amazon, CloudFlare, Dropbox ou encore Twitter ont connu des difficultés. En cause, un problème sur un câble sous-marin qui relie les États-Unis à une partie de l'Europe du Nord, dont la France.

[...]

Afin de desservir les différents pays à travers les cinq continents, internet passe bien souvent par des câbles sous-marins. Ces derniers transportent les données via un réseau de fibres optiques, assurant à la fois rapidité et très haut débit. L'un d'entre eux, TAT-14, qui revendique une bande passante de 3,2Tb/s, a été victime d'un incident provoquant quelques ralentissements de ce côté de l'Atlantique. [...] Parmi les services touchés, on peut notamment citer Twitter (qui a publié une alerte), Amazon, CloudFlare [...], Dropbox, etc. Pour autant, la situation n'était pas catastrophique, car fort heureusement ce n'est pas le seul câble transatlantique de disponible, et il était donc possible de passer par une autre route afin que tout rentre dans l'ordre.

DOCUMENT 2 : Florian Bayard. *Trois câbles sous-marins ont été sabotés l'un après l'autre en Europe : l'enquête continue*. 01Net, 19 janvier 2026

Les sabotages de câbles sous-marins de communication par fibre optique se multiplient dans la mer Baltique. Ces derniers jours ont en effet été marqués par la destruction successive de plusieurs câbles chargés d'acheminer le trafic Internet et les télécommunications à travers le monde. Le premier incident est survenu le 30 décembre 2025, entre l'Estonie et la Suède. **Deux sections du câble** ont été endommagées et le trafic Internet a été réorienté. L'incident est rapidement considéré comme « suspect » par les gouvernements estonien et suédois. Les dégâts sont apparus alors qu'aucune activité sismique n'a été enregistrée.

Quelques heures plus tard, le 31 décembre 2025, un deuxième câble est coupé dans le golfe de Finlande. Tôt le matin, un grand opérateur télécom finlandais détecte une « perturbation » sur un câble de données reliant Helsinki à Tallinn. Le câble, considéré comme **une infrastructure critique** par la Finlande, est sévèrement endommagé. Néanmoins, le trafic est rapidement rerouté via d'autres liaisons, ce qui a évité une panne d'Internet. La « panne n'affecte pas les utilisateurs, mais elle augmente la vulnérabilité du système », explique Anders Wallinder, responsable préparation et sécurité à l'autorité suédoise des postes et télécoms. Enfin, un troisième câble a été sectionné le vendredi 2 janvier 2026 entre la Lituanie et la Lettonie. [...] L'opérateur télécom suédois Arelion indique que le câble, posé depuis 1995, est « entièrement sectionné », mais que le trafic a été rerouté, évitant là encore une panne de réseau.

Les autorités ont pu remonter jusqu'à un navire venu d'un port russe. Suite au sabotage du 31 décembre au large de la Finlande, les garde-côtes [...] ont épinglé le navire Fitburg, **un cargo parti de Saint-Petersbourg** quelques jours plus tôt. Les radars et données maritimes montrent que le Fitburg se trouvait exactement sur la trajectoire du câble au moment de la rupture. Le navire [...] part d'un port russe, et transporte de l'acier russe illégalement. [...] Le navire est suspecté d'avoir sciemment fait **traîner son ancre dans les fonds marins** sur des dizaines de kilomètres pour tout détruire sur son passage.

DOCUMENT 3 : *Trafic Internet mondial par année*, CISCO VNI, 2018.

1992	100 Go/jour	2007	2000 Go/s
1997	100 Go/h	2017	46600 Go/s
2002	100 Go/s	2022	150700 Go/s

C – Modèles d'architecture

Activité 17 (Jeu du routage)

Voir les règles du jeu en annexe.

Après avoir joué au jeu, répondez aux questions suivantes :

- 1► Bilan de la phase 1 :
 - a► Quelles stratégies avez vous adoptées? Étaient-elles efficaces?
 - b► Quels problèmes avez vous rencontrés?
- 2► Bilan de la phase 2 :
 - a► Quelles sont les différences entre le déroulement de la première phase et de la deuxième?
- 3► Bilan de la phase 3 :
 - a► Quels avantages cette organisation a-t-elle par rapport aux deux premières phases?
 - b► Quels problèmes le serveur a-t-il pu rencontrer? Et les clients?
- 4► Bilan de la phase 4 : quel problème est rapidement apparu pour le serveur?

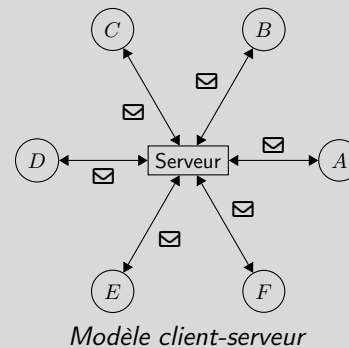
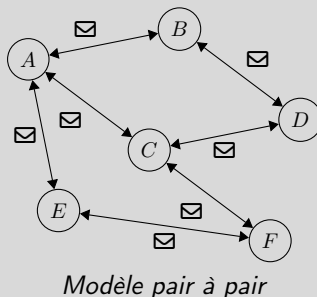
Architectures réseau

Définition 18

Une *architecture* d'un réseau est un ensemble de règles qui décrivent l'organisation entre les différents acteurs du réseau.

Il existe deux grands modèles d'architecture :

- Dans le modèle *pair à pair* (en anglais, *Peer to peer*, ou *P2P*), toutes les machines peuvent envoyer et recevoir des données directement entre elles, sans intermédiaire.
- Dans le modèle *client-serveur*, une machine centrale (le *serveur*) répond aux demandes (aux *requêtes*) des autres machines (les *clients*). Pour qu'un client transmette une donnée à un autre, il doit la confier au serveur qui se chargera de la mettre à disposition de client destinataire.



Exemple 19

- Client-serveur : mails, réseaux sociaux, plateformes de streaming, ...
- Pair à pair : AirDrop™ / Quick Share™, cryptomonnaies (blockchain), mode multijoueur local dans les jeux vidéos, ...

Les différents modèles d'architecture ont leurs avantages et inconvénients, et le choix d'un modèle dépend des contraintes et des besoins.

	Pair à pair	Client-serveur
Contrôle des données	Aucun (toutes les données sont libres de circuler)	Total (le serveur décide quelles données ont le droit ou non de circuler)
Sécurité	Faible (en l'absence de contrôle, les données malveillantes circulent librement)	Élevée (le serveur peut empêcher la diffusion de données malveillantes)
Coût	Faible (pas besoin d'infrastructure supplémentaire)	Élevé (il faut mettre en place une machine spéciale supplémentaire qui consomme de l'énergie en continu)
Performances	Illimitée (on peut inclure autant de machine que l'on souhaite dans le réseau)	Limitée (le serveur est capable d'accepter un nombre limité de connexions simultanées)
Résilience	Haute (même si une machine du réseau tombe en panne, les autres peuvent prendre le relai)	Limitée (si le serveur tombe en panne, plus aucune donnée ne peut circuler)

Définition 20**Déni de service**

Un *déni de service* est l'événement où un serveur reçoit dans un court laps de temps plus de requêtes qu'il n'est capable de gérer.

Une *attaque par déni de service* (en anglais : *DDoS*) est l'acte malveillant consistant à envoyer simultanément un très grand nombre de requêtes dans le but de mettre en défaut un serveur.

Information 21

Les réseaux pair à pair sont souvent utilisés pour le partage illégal de fichiers. Le téléchargement d'un fichier, quelle que soit la méthode, sans l'accord de son auteur, est un crime puni par la loi.

Les réseaux pair à pair peuvent aussi être utilisés pour le bien commun : par exemple, plusieurs projets de recherche participative proposent à quiconque de mettre son ordinateur à disposition d'un réseau pair à pair pour en prêter la puissance de calcul.

2 – Internet : un réseau de réseaux

A – Vocabulaire et analogies

L'ensemble des technologies qui permettent le partage de données numériques à distance est regroupé en plusieurs « couches », chacune reposant sur la précédente.

- ▶ Couche physique : Ensemble des infrastructures matérielles permettant le transport des données, et des règles pour les utiliser
- ▶ Couche liaison : Ensemble de méthodes pour s'assurer que les données arrivent dans la bonne machine à la fin de leur voyage
- ▶ Couche réseau : Ensemble de méthodes pour « rapprocher » les données de leur destination
- ▶ Couche transport : Ensemble de méthodes qui décrient la manière de transmettre les données
- ▶ Couche application : Ensemble de règles pour façonner les données

Pour chacune d'entre elle, on peut faire une analogie entre le transport de données dans un réseau informatique, et le transport de courrier par La Poste.

ANALOGIE Courrier

Écrire le contenu de la lettre
au bon format (est-ce que
c'est un CV, un courrier offi-
ciel, une lettre entre amis...)

Mettre le courrier dans une en-
veloppe, écrire les adresses des-
tinateur et destinataire, four-
nir un accusé de réception

Acheminer la lettre
dans la bonne ville

Distribution par le bu-
reau de poste de la ville

Camions, trains, vélos, ...

Couche application

Couche transport

Couche réseau

Couche liaison

Couche physique

RÉALITÉ Réseau informatique

Mise en forme des données qui sont
envoyées (est-ce que c'est un mail,
une page web, un SMS, ...)

Découper les données en petits pa-
quets ordonnés, écrire les adresses
destinateur et destinataire, s'assurer
de leur bonne livraison

Acheminer les données vers le bon
réseau local

Réseaux locaux (maisons, écoles, ...)

Réseau cuivré, fibre optique, ondes
radio, ...

Définition 22

Protocole, norme

Une *norme* est une règle sur laquelle tout le monde se met d'accord.

Un *protocole* (informatique) est une méthode, un plan d'action sur lequel tout le monde se met d'accord.

Les normes et les protocoles servent à faciliter l'interopérabilité : si tout le monde agit de la même manière, il est plus simple de se coordonner.

Exemple 23

- ▶ La norme Ascii permet d'associer un nombre à chaque caractère du clavier. Par exemple, un A majuscule est représenté par le numéro 65, un b minuscule par 98, ...
- ▶ Certains formats de fichiers, par exemple JPEG, PNG ou PDF, sont définis par des normes.
- ▶ Les règles de politesse (dire bonjour, ne pas couper la parole, ...) peuvent être vues comme un protocole pour parler.

Remarque 24

En Python, la commande `ord` donne le code Ascii du caractère passé en argument, et la commande `chr` prend un nombre en argument, et renvoie le caractère correspondant.

Par exemple, `ord("A")` donne 65, et `chr(98)` donne 'b'.

Définition 25

Internet

Internet est un « réseau de réseaux » : plus précisément, Internet est un ensemble de protocoles permettant l'échange de données entre plusieurs réseaux locaux, lesquels sont reliés par un réseau plus grand.

Le protocole phare, au cœur d'Internet, est le *Internet Protocol*, ou IP.

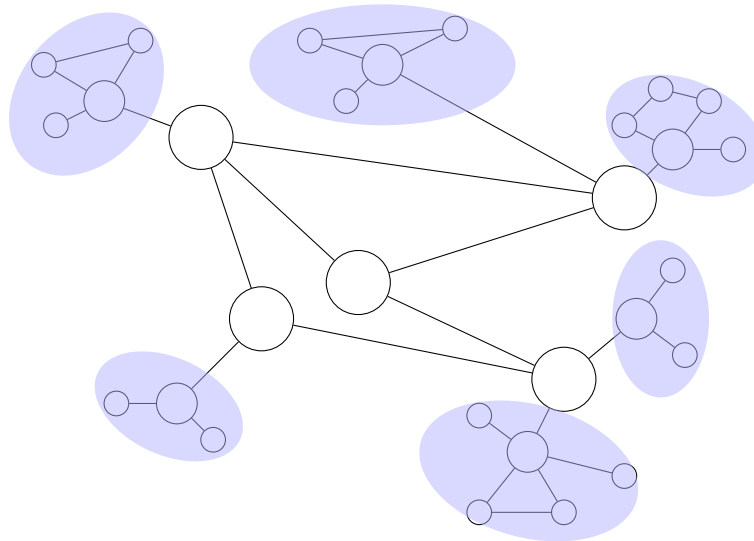


Illustration d'un réseau de réseaux : les réseaux locaux, surlignés en bleu, sont reliés entre eux par l'intermédiaire d'un réseau plus large, Internet.

Remarque 26

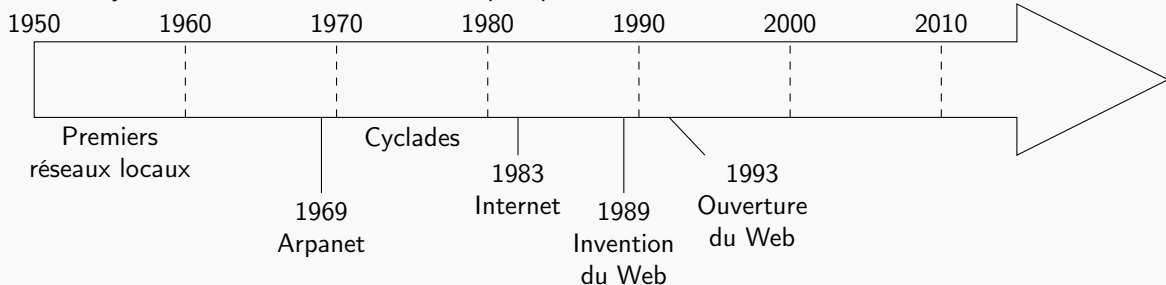
En anglais, un réseau se dit « *net* », et « *Internet* » est donc l'*inter*connexion de plusieurs *réseaux*.

Information 27

Repères historiques

Arpanet visait à relier entre eux les réseaux universitaires, malgré l'hétérogénéité de ces derniers. Il a été financé par l'ARPA ((Defense) Advanced Research Projects Agency).

Le réseau *Cyclades*, inventé en France, est le plus proche d'être l'ancêtre d'Internet.



Définition 28

Routeur, Switch

Les réseaux locaux sont constitués de différentes machines. Les appareils reliant les différents réseaux locaux sont appelés des *routeurs*. Les réseaux locaux sont reliés au réseau des routeurs par l'intermédiaire d'un *switch*.

Remarque 29

Sur le schéma précédent, les nœuds les plus gros représentent les routeurs, les nœuds de taille intermédiaire représentent les switches, et les plus petits sont les machines des réseaux locaux.

On peut faire une analogie avec les transports publics : le réseau des routeurs est comme le réseau autoroutier, les switches sont alors les portes de péage qui permettent d'entrer et sortir des autoroutes, et les réseaux locaux sont les villes desservies par les autoroutes.

Neutralité du Net

Information 30

La *neutralité du Net* est le principe selon lequel les Fournisseurs d'Accès Internet (FAI, en France : Orange, Bouygues, Free, ...) doivent garantir l'égalité entre tous les utilisateurs, leur permettre de transmettre des données librement, sans prioriser certains utilisateurs, ou en censurer d'autres. Depuis 2016, la neutralité du Net est légalement garantie dans l'Union Européenne.

B – Des adresses pour se repérer

Activité 31 (Problèmes de dénombrement)

- 1▶ Alice se rend au restaurant.
 - ▶ Pour l'entrée, elle a le choix entre des carottes râpées, du taboulé, une salade de lentilles ou des œufs mimosa.
 - ▶ Pour le plat principal, elle a le choix entre un gratin de légumes, du poulet rôti ou un saumon meunière.
 - a▶ Lister toutes les formules entrée-plat qu'Alice peut choisir. Combien y en a-t-il au total ?
 - b▶ Combien y a-t-il d'entrées différentes ? Combien de plats ? Comment combiner ces deux nombres pour calculer le nombre total de formules, trouvé à la question précédente ?
- 2▶ Bob a oublié le code de son cadenas à quatre chiffres.
 - ▶ Combien y a-t-il de possibilités pour le premier chiffre ? Et pour les autres ?
 - ▶ En déduire le nombre total de combinaisons possibles pour le code du cadenas. Exprimer le résultat sous la forme d'une puissance.
 - ▶ Bob se souvient que le premier chiffre est 1, et que le dernier chiffre est pair. Combien de codes Bob doit-il tester au maximum pour ouvrir son cadenas ?
- 3▶ On rappelle qu'un octet est constitué de 8 bits, et que chaque bit vaut soit 0 soit 1.
 - a▶ Combien d'octets différents peut-on écrire ? Exprimer le résultat sous la forme d'une puissance avant de le calculer.
 - b▶ Une adresse IPv4 utilise 4 octets. Combien existe-t-il d'adresses IPv4 ?
 - c▶ Une adresse IPv4 utilise 128 bits. Combien existe-t-il d'adresses IPv6 ?

Définition 32**Adresse IP**

Chaque machine d'un réseau est repérée à l'aide de son *adresse IP*, une suite de nombres séparés par des points. C'est l'équivalent informatique de l'adresse postale que l'on écrit sur une enveloppe.

Information 33

Le protocole IP, et par extension les adresses IP, ont évolué en plusieurs versions. La plus utilisée est l'IPv4, qui code les adresses IP sur 32 bits, soit 4 octets, représentés par 4 nombres entre 0 et 255, séparés par des points.

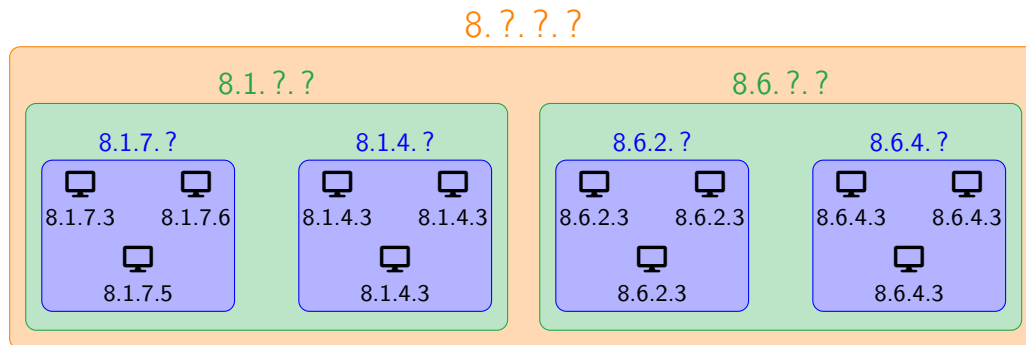
En raison de l'explosion du nombre de machines sur le réseau Internet, les adresses IPv4 viennent à manquer, c'est pourquoi on passe progressivement à l'IPv6, qui code les adresses sur 128 bits, soit 16 octets. Contrairement à l'IPv4, les nombres de l'adresse sont écrits en base hexadécimale.

Exemple 34

L'adresse IPv4 du serveur de YouTube est 142.251.142.206, et son IPv6 est 2a00:1450:400e:807::200e.

Sur une enveloppe postale, on écrit l'adresse de manière hiérarchique : le numéro du bâtiment, puis la rue où il se

trouve, puis la ville où se trouve la rue, et éventuellement le pays ou la région où se trouve la ville. De la même manière, les nombres successifs d'une adresse IP représentent des sous-réseaux, du plus large au plus fin, jusqu'au numéro de la machine.



Ils serait bien compliqué de devoir connaître l'adresse IP de chaque serveur quand on peut accéder à un site qu'il héberge ! Ce serait comme devoir retenir les coordonnées GPS de la personne à laquelle on envoie une lettre... Pour remédier à ce problème, on utilise une *URL (Uniform Resource Locator)* (par exemple : youtube.com), et un *serveur DNS (Domain Name System)* s'occupe de faire la traduction entre URL et adresse IP (pour faire une analogie, un serveur DNS est l'équivalent des annuaires téléphoniques, qui permettent de retrouver le nom d'une personne à partir de son numéro de téléphone).

Plus de détails sur les URL dans le chapitre sur le Web !

Activité 35 (TP)

- 1► Ouvrir l'invite de commande Windows (pour cela, vous pouvez presser les touches Windows + R et taper cmd).

- 2► a► Taper ipconfig et compléter le tableau suivant :

Adresse IPv4 de votre ordinateur	
Adresse IPv6 de votre ordinateur	

- b► Demandez son adresse IPv4 à votre voisin : _____

- c► Recopier ici l'adresse appelée « Passerelle par défaut » : _____

Il s'agit de l'adresse du switch du réseau local auquel vous êtes connecté, autrement dit le premier routeur par lequel transitent les données entre votre ordinateur et le réseau Internet.

- 3► La commande ping permet de tester la connexion avec une machine du réseau. Par exemple, ping 1.2.3.4 teste la connexion entre votre machine et celle à l'adresse 1.2.3.4.

- a► Testez la communication avec l'adresse de la Passerelle par défaut. Que se passe-t-il ?
- b► Testez la connexion avec l'adresse 10.11.12.13. Existe-t-il une machine possédant cette adresse ?
- c► Testez la communication avec votre voisin (utilisez l'adresse relevée dans la première question). Que se passe-t-il ?

- 4► La commande ipconfig ne donne que des informations sur le réseau local. Le système d'adresses IP est différent selon que l'on considère le réseau local ou le réseau Internet (réseau global).

Le switch du réseau local étant le passage obligatoire des données entre les machines du réseau local et Internet, il est le seul à disposer d'une adresse globale, appelée **adresse IP publique** (il s'occupe par la suite de redistribuer les données au sein du réseau local).

- a► À l'aide du site <https://www.my-ip-finder.fr>, trouver l'adresse IP publique du switch du lycée.

- b ▶ Selon ce qui est affiché par le site, quelle information en particulier une adresse IP publique permet-elle de trouver ?
- c ▶ Vous avez reçu un message provenant de l'adresse 18.181.0.31. D'où vient la personne qui a envoyé ce message ? (*Aidez vous de l'onglet « Trouver une adresse IP » du site précédent.*)
- 5 ▶ a ▶ Dans l'invite de commande, taper `ping google.fr`. Que se passe-t-il ?
- b ▶ À l'aide de l'onglet « DNS Lookup » du site de la question précédente, trouver l'adresse IP correspondant au nom de domaine `google.fr`.

C – Des protocoles pour bien s'entendre

Information 36

Il existe différents protocoles pour différents usages :

- ▶ SMTP pour l'envoi de mails
- ▶ IMAP ou POP pour la lecture de mails
- ▶ HTTP pour échanger de l'hypertexte (voir le chapitre sur le Web)
- ▶ TCP (*Transmission Control Protocol*) pour échanger des données de manière fiable
- ▶ UDP (*User Datagram Protocol*) pour échanger des données rapidement

Il en existe bien d'autres !

Le protocole le plus utilisé pour l'échange de fichiers est le protocole TCP (souvent adjoint au protocole IP dans la dénomination « TCP/IP »). Le protocole TCP se déroule en plusieurs grandes étapes :

- 1 ▶ Il découpe les données en petits paquets, et numérote ces paquets.
- 2 ▶ Il écrit sur chaque paquet plusieurs informations, notamment les adresses IP de l'émetteur et du destinataire.
- 3 ▶ Il envoie les paquets les uns après les autres (le protocole IP prend alors le relais pour les amener à bon port).
- 4 ▶ Une fois que le destinataire reçoit un paquet, il envoie un accusé de réception à l'émetteur. Si l'émetteur ne reçoit pas d'accusé de réception après un certain temps (*timeout*), alors ce dernier renvoie le paquet.
- 5 ▶ Une fois que le destinataire a reçu tous les paquets, il les remet dans l'ordre grâce à leur numéro.

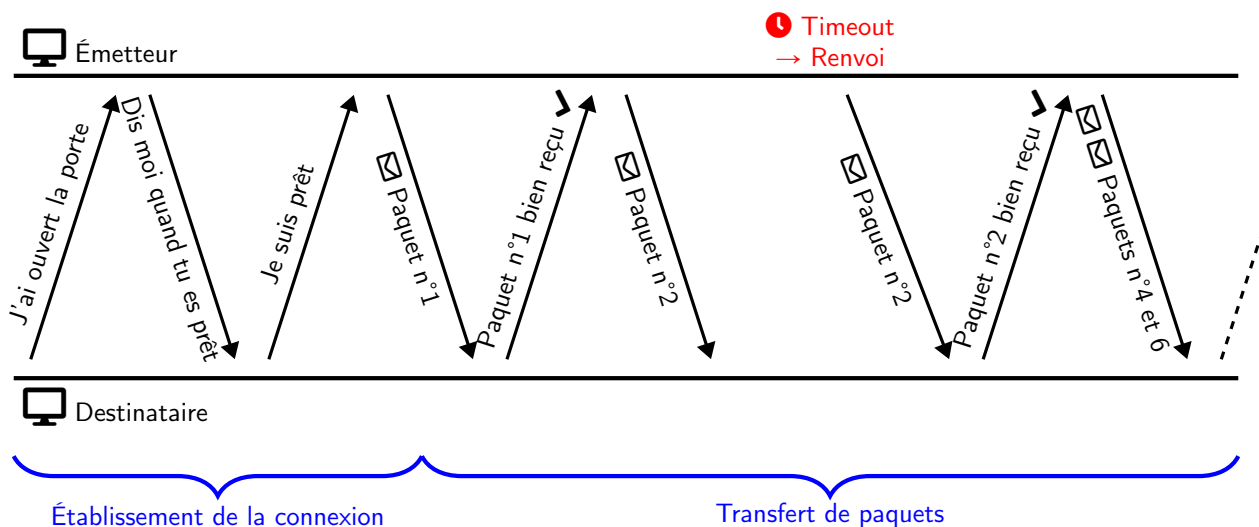


Illustration du protocole TCP

Quel intérêt ? Imaginez devoir transmettre un très gros fichier en un morceau. Durant le transfert, il y aura

peut-être des erreurs, des altérations, des délais, ... et si les problèmes s'accumulent, alors le fichier risque d'être trop dégradé et vous n'aurez plus qu'à recommencer le transfert de zéro. C'est tout l'intérêt du découpage en petit paquet : s'il y a un problème sur le transfert d'un paquet, alors on peut se contenter de renvoyer ce paquet-là, au lieu du document entier.

L'intérêt du protocole TCP est la fiabilité : grâce à lui, peu importe si certains paquets sont perdus ou détruits en chemin, peu importe s'ils sont mélangés en route, à la fin, le destinataire a l'assurance d'avoir le fichier complet et dans le bon ordre. Cependant, cette fiabilité se fait au détriment de la rapidité : même s'il est possible d'envoyer plusieurs paquets en même temps, il faudra tout de même attendre leur accusé de réception.

Un autre protocole, le protocole UDP, est souvent utilisé quand on cherche à transmettre un fichier rapidement, quitte à autoriser la perte de quelques données en chemin. Typiquement, un appel vocal, un stream vidéo, ... peuvent se permettre de perdre quelques données ça et là, et ont besoin de transmettre dans une fenêtre très restreinte un grand volume de données.

Exercice 37

Carole affirme que si on regarde une vidéo avec les paramètres « 720p@30Hz », on consomme environ 5 fois de données qu'avec les paramètres « 4K@60Hz ».

À l'aide de vos connaissances et des documents ci-dessous, commenter l'affirmation de Carole

DOCUMENT 1 : Définition d'image

Sur les plateformes de streaming vidéo, on peut régler la qualité de la vidéo, avec comme options 360p, 720p, 1080p, 4K, ... Ces notations désignent la *définition* de l'image, autrement dit elles renvoient au nombre de pixels contenus dans l'image.

<i>Formats haute définition</i>		
Nom	Raccourci	Définition (pixels)
HD	720p	720 × 1280
Full HD	1080p	1080 × 1920
4K	4K ou 4096p	4096 × 2160

DOCUMENT 2 : Taux de rafraîchissement

Un écran est capable d'afficher des images les unes à la suite des autres, mais changer d'image lui prend du temps. Il existe une vitesse maximale à laquelle un écran peut changer d'image : on l'appelle le *taux de rafraîchissement*, et on l'exprime en Hertz (Hz). Par exemple, un écran qui peut afficher 144 images différentes chaque seconde a un taux de rafraîchissement de 144Hz.

Une vidéo est elle aussi une succession d'images statiques, on utilise donc également un taux de rafraîchissement pour quantifier le nombre d'images qu'elle contient chaque seconde. Pour dire qu'une vidéo affiche 24 images en définition 360p chaque seconde, on utilise la notation 360p@24Hz.

Annexe : jeu du routage

Le professeur a formé des groupes de trois élèves. Asseyez vous en îlots dans la salle, assez proches pour pouvoir atteindre l'îlot voisin à bout de bras (mais restez bien séparés quand même!).

⚠ Important – Consigne de sécurité : chaque élève prend avec lui un stylo, et met le reste de ses affaires sous le tableau, *hors du passage*.

Phase 1

Vous disposez d'enveloppes, de cartes « Action » (6 cartes « Tout va bien ! », 3 cartes « Délai », 2 cartes « Panne »), de jetons « Formes » et de bouts de papier.

Le professeur vous donne une carte « Destinataire » (une par personne) : gardez là devant vous. Votre objectif va être d'envoyer un message à cette personne. Pour cela :

- ▶ Piochez 4 jetons « Formes » pour constituer un message.
- ▶ Utilisez les enveloppes et les papiers pour transmettre ce message à votre destinataire.
(⚠ Interdiction d'écrire sur le matériel, pour pouvoir le réutiliser. Utilisez les bouts de papier.)

Pour cela, vous devrez passer ce message à vos voisins en respectant les règles suivantes :

- ▶ Interdiction de se lever ou de faire des signes. Vous n'avez le droit de communiquer qu'avec vos voisins directs.
- ▶ **Avant** de confier le message à la personne suivante, piochez une carte « Action » et effectuez l'action demandée.
- ▶ Une fois l'action réalisée, vous pouvez remettre la carte « Action » en-dessous du paquet.

Vous avez 5 minutes pour transmettre votre message. À l'issue de ce temps, écoutez les consignes et allez consulter votre destinataire pour savoir s'il a bien reçu un message de votre part (et lequel!).

Bonne chance !

✂ ----- ✂

Phase 2

Commencez par rassembler le matériel de la Phase 1, le professeur va distribuer le nouveau matériel.

Dans cette nouvelle phase de jeu, vous allez rejouer la première phase, mais avec des nouvelles règles :

- ▶ Vous n'avez plus le droit de mettre n'importe quel papier dans les enveloppes : pour transmettre un message, utilisez les jetons « Formes » et les cartons « Adresse » (en vert).
- ▶ Pour déterminer le message à transmettre, piochez 4 jetons « Formes ». Notez sur votre « Feuille de suivi » le code que vous allez transmettre.
- ▶ Vous n'avez le droit d'envoyer qu'un seul jeton dans chaque enveloppe.
- ▶ Lorsque le destinataire reçoit un jeton, il doit transmettre un carton « Accusé de réception » à l'émetteur.
- ▶ Action « Timeout » (2 cartes par personne) : si vous avez envoyé un jeton mais que vous n'avez pas encore reçu son Accusé de réception, renvoyez ce jeton.

Prenez également 3 cartes « Tout va bien ! » supplémentaires. Le reste des règles pour transmettre une enveloppe avec les cartes « Action » ne change pas.

Vous avez 10 minutes. Bonne chance !

Phase 3

Commencez par ranger le matériel de la phase 2. Rendez les cartes « Action » au professeur, vous ne vous en servirez plus.

Nous allons réorganiser la classe : rassemblez les groupes voisins afin de former des pôles d'entre 7 et 10 élèves.

Installez une table à côté du pôle, et espacez les autres tables de manière à pouvoir circuler.

Si la place le permet, installez les tables en arc de cercle, voire en cercle, de façon à pouvoir circuler au centre sans se blesser.



Dans chaque pôle, désignez deux élèves (pour l'exemple, nommons les Alice et Bob).

Alice et Bob se tiennent debout à côté de la table spéciale (on l'appellera le « serveur »). Les autres élèves s'assoient aux autres tables (on les appellera les « clients ») (si les tables sont disposées en cercle ou en arc de cercle, les clients doivent faire face au serveur.)

Les nouvelles règles sont les suivantes :

- ▶ Chaque client doit envoyer un message (3 jetons) à un autre client de son choix.
- ▶ Les clients n'ont pas le droit de s'échanger des enveloppes directement : lorsqu'ils souhaitent envoyer une enveloppe, ils lèvent la main, et Alice se charge d'aller la récupérer, pour la poser sur la table du serveur.
- ▶ Bob quant à lui se charge de redistribuer les enveloppes lorsqu'elles arrivent au serveur.
- ▶ Alice et Bob n'ont le droit de porter qu'une seule enveloppe à la fois.
- ▶ Il n'y a plus besoin des cartes « Action », mais utilisez la « Feuille de suivi », les cartons « Adresse » et les « Accusés de réception » comme dans la phase 2.

Vous avez 5 minutes. Bon courage !





Phase 4

Rangez les tables dans leur disposition habituelle. Désignez 2 élèves (voire 4 si la classe a au moins 22 élèves) qui joueront le même rôle qu'Alice et Bob dans la phase précédente.

Rejouez la phase précédente avec toute la classe en même temps.

Soyez attentifs au signal du professeur pour terminer le jeu. Bonne chance !

Feuille de suivi

Code à transmettre (redessiner dans les cases et poser les jetons dessus)

Lorsque vous l'avez reçu, placez le jeton Accusé de réception sur la case correspondante.

Feuille de suivi

Code à transmettre (redessiner dans les cases et poser les jetons dessus)

Lorsque vous l'avez reçu, placez le jeton Accusé de réception sur la case correspondante.

**Accusé de
réception**

**Accusé de
réception**

**Accusé de
réception**

**Accusé de
réception**

**Accusé de
réception**

**Accusé de
réception**

**Accusé de
réception**

**Accusé de
réception**

**Accusé de
réception**

**Accusé de
réception**

**Accusé de
réception**

**Accusé de
réception**

Émetteur :

Destinataire :

Paquet n°

Émetteur :

Destinataire :

Paquet n°

Émetteur :

Destinataire :

Paquet n°

Émetteur :

Destinataire :

Paquet n°

Émetteur :

Destinataire :

Paquet n°

Émetteur :

Destinataire :

Paquet n°

Émetteur :

Destinataire :

Paquet n°

Émetteur :

Destinataire :

Paquet n°

DÉLAI



Attendez d'envoyer la prochaine enveloppe avant de transférer celle-ci.
Si vous n'avez pas d'autre enveloppe, gardez l'actuelle jusqu'à en recevoir une nouvelle.

DÉLAI



Attendez d'envoyer la prochaine enveloppe avant de transférer celle-ci.
Si vous n'avez pas d'autre enveloppe, gardez l'actuelle jusqu'à en recevoir une nouvelle.

DÉLAI



Attendez d'envoyer la prochaine enveloppe avant de transférer celle-ci.
Si vous n'avez pas d'autre enveloppe, gardez l'actuelle jusqu'à en recevoir une nouvelle.

PANNE



Détruisez* le message !
(* remettez l'enveloppe et le jeton dans la pile de matériel, et déchirez les papiers colorés.)

TOUT VA BIEN !



Transférez le message sans problème.

PANNE



Détruisez* le message !
(* remettez l'enveloppe et le jeton dans la pile de matériel, et déchirez les papiers colorés.)

TIMEOUT



Si vous avez précédemment envoyé un jeton mais que vous n'avez pas encore reçu d'accusé de réception, renvoyez ce jeton.

Si vous n'avez aucun jeton dans ce cas, ignorez cette action.

Si vous avez plusieurs jetons dans ce cas choisissez en un.

TIMEOUT



Si vous avez précédemment envoyé un jeton mais que vous n'avez pas encore reçu d'accusé de réception, renvoyez ce jeton.

Si vous n'avez aucun jeton dans ce cas, ignorez cette action.

Si vous avez plusieurs jetons dans ce cas choisissez en un.

ACTION

ACTION

ACTION

ACTION

ACTION

ACTION

ACTION

ACTION



TOUT VA BIEN !

Transférez le message sans problème.



TOUT VA BIEN !

Transférez le message sans problème.



TOUT VA BIEN !

Transférez le message sans problème.



TOUT VA BIEN !

Transférez le message sans problème.



TOUT VA BIEN !

Transférez le message sans problème.



TOUT VA BIEN !

Transférez le message sans problème.



TOUT VA BIEN !

Transférez le message sans problème.



TOUT VA BIEN !

Transférez le message sans problème.

ACTION

ACTION

ACTION

ACTION

ACTION

ACTION

ACTION

ACTION

4 Le Web : une manière d'échanger

Table des matières

Parler du ASCII pour les protocoles

Analogie pour protocole HTTP : (attention modification par rapport à la vidéo d'Adrien) Livreur de pizza Protocole : client appelle la pizzeria Lorsqu'elle est dispo, elle décroche <=> serveur dispo et prêt à écouter Client commande <=> requête serveur S'il demande autre chose qu'un pizza, ou une pizza qu'ils n'ont pas, le serveur répond qu'ils ne l'ont pas (404) Sinon, il lui dit qu'il prépare la commande et qu'elle sera envoyée Ensuite le livreur s'occupe de livrer la pizza <=> transfert de paquets Si trop de pizza à livrer, besoin de plusieurs livreurs ; certains vont se perdre, avoir un souci en route, et ne pas arriver à destination, ou alors pas dans le même ordre <=> perte de paquets À la fin, la pizzeria appelle le client pour confirmer la réception de toutes les pizza ; si ok c'est bon, sinon le client dit ce qui lui manque et le serveur renvoie

Thème d'étude III

Imagerie numérique

Programme

La photographie numérique

Introduction

Les technologies de la photographie argentique ont eu une évolution très lente, liée aux progrès en optique, mécanique et chimie. Ce n'est plus du tout le cas de l'évolution actuelle, davantage due aux algorithmes qu'à la physique : algorithmes de développement et d'amélioration de l'image brute, algorithmes d'aide à la prise de vue. Cet exemple est caractéristique des façons de procéder de la révolution informatique par rapport aux approches traditionnelles.

La photographie numérique présente un coût marginal très faible et une diffusion par internet facile et immédiate : chaque jour, des milliards de photos sont prises et partagées.

Repères historiques

- ▶ 1826 : naissance de la photographie argentique ;
- ▶ 1900 : photographie en couleurs. Après la seconde guerre mondiale, généralisation du format 24 x 36 et de la visée reflex ;
- ▶ 1969 : arrivée des premiers capteurs CCD (Charge Coupled Device) ;
- ▶ 1975 : apparition des premiers appareils numériques ;
- ▶ 2007 : arrivée du smartphone.

Les données et l'information

En entrée, le capteur est formé de photosites en matrice de petits carrés de quatre photosites, deux verts, un bleu et un rouge, correspondant à la répartition des cônes de la rétine. La résolution du capteur se mesure en millions de photosites.

En sortie, l'image est formée de pixels colorés homogènes, représentés par trois nombres RVB (rouge, vert, bleu). La résolution de l'image se compte en mégapixels ; elle n'est pas forcément égale à celle du capteur. La profondeur de couleur est en général de 8 bits par pixel et par couleur pour l'image finale.

Des métadonnées sont stockées dans les fichiers images sous format EXIF (Exchangeable Image File Format) : modèle de l'appareil, objectif, vitesse, diaphragme, distance de mise au point, auteur, copyright, localisation, etc.

Les couleurs peuvent être représentées dans différents systèmes : RVB, TSL (teinte, saturation, lumière), avec des formules empiriques de passage d'un modèle à l'autre. On distingue différents formats des fichiers images, compressés ou non, avec ou sans perte : RAW, BMP, TIFF, JPEG.

Les algorithmes et les programmes

Des algorithmes permettent de traiter toutes les lumières, d'effectuer une retouche facile, avec une qualité maintenant bien supérieure à l'argentique. Avec l'arrivée du téléphone mobile, des algorithmes de fusion d'images permettent de concilier une excellente qualité avec un capteur et un objectif minuscules. De nombreux algorithmes sophistiqués sont utilisés dans les appareils de photographie numérique :

- ▶ Lors de la prise de vue : calcul de l'exposition, mise au point, stabilisation par le capteur et/ou l'objectif, le tout en automatique ou manuel assisté, focus-peaking (scintillement des contours nets), prise en rafales rapides d'images multiples avant et après appui sur le déclencheur.
- ▶ Lors du développement de l'image issue du capteur en une image pixellisée : gestion de la lumière et du contraste, balance des blancs, netteté, débouchage des ombres, correction automatique des distorsions ou des aberrations optiques.
- ▶ Après le développement : compression du fichier (TIFF sans perte, JPEG avec perte).

- ▶ En utilisant la fusion d'images : réduction du bruit et amélioration de la netteté, panoramas, HDR (High Dynamic Range), super-résolution par micro-décalages du capteur, focus stacking pour étendre la netteté avec plusieurs mises au point successives, réduction du bruit et amélioration de la netteté.
- ▶ Certains appareils peuvent augmenter leurs fonctionnalités par téléchargement de nouveaux logiciels.

Les machines

Comme les algorithmes de prise de vue et de développement demandent beaucoup de calcul, les appareils embarquent plusieurs processeurs, généraux ou spécialisés. Les algorithmes prennent le relais des capteurs physiques en calculant les pixels de l'image finale : ils compensent par exemple les distorsions des lentilles. Des algorithmes permettent également de commander la mise au point et l'exposition automatique, ainsi que de compenser le bougé de l'utilisateur (stabilisation).

Impacts sur les pratiques humaines

La gratuité et l'immédiateté de la réplique des images introduisent de nouveaux usages de la photographie : à la photographie archive (histoire de famille) s'ajoutent la photographie à partager et la photographie utilitaire, prothèse de la mémoire (photo d'un ticket de caisse, d'une présentation lors d'une réunion de travail, d'une place de parking, etc.). Les images s'intègrent à tous les dispositifs de communication et de partage, téléphones, Web et réseaux sociaux.

De nouveaux problèmes apparaissent, liés à la diffusion de photos qui ne disparaîtront jamais (notion de droit à l'oubli), au truquage difficile à détecter des images, au pistage des individus ou à l'obsolescence des supports. Est ainsi posée la question de l'archivage de photographies historiques, scientifiques ou culturelles.

Contenus	Capacités attendues
Photosites, pixels, résolution (du capteur, de l'image), profondeur de couleur	Distinguer les photosites du capteur et les pixels de l'image en comparant les résolutions du capteur et de l'image selon les réglages de l'appareil.
Métadonnées EXIF	Retrouver les métadonnées d'une photographie.
Traitement d'image	Traiter par programme une image pour la transformer en agissant sur les trois composantes de ses pixels.
Rôle des algorithmes dans les appareils photo numériques	Expliciter des algorithmes associés à la prise de vue. Identifier les étapes de la construction de l'image finale.

Exemples d'activités

- ▶ Programmer un algorithme de passage d'une image couleur à une image en niveaux de gris : par moyenne des pixels RVB ou par changement de modèle de représentation (du RVB au TSL, mise de la saturation à zéro, retour au RVB).
- ▶ Programmer un algorithme de passage au négatif d'une image.
- ▶ Programmer un algorithme d'extraction de contours par comparaison entre pixels voisins et utilisation d'un seuil.
- ▶ Utiliser un logiciel de retouche afin de modifier les courbes de luminosité, de contraste, de couleur d'une photographie.

5 Les images numériques

Table des matières

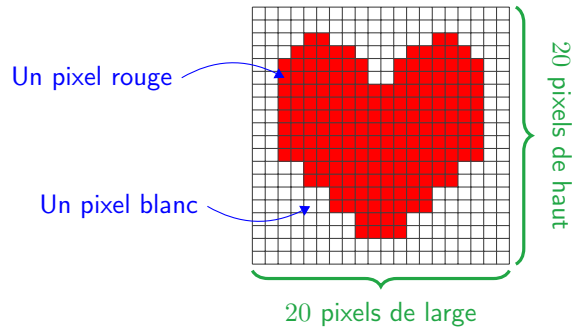
1	Les images comme grilles de pixels	65
2	Représentations numériques de couleurs	66
A	Encoder la perception humaine : le code RVB	66
B	Encoder la sensation humaine : le code TSL	69
C	Autres encodages	70
3	Le fonctionnement des écrans	71

1 – Les images comme grilles de pixels

Définition 1

Une image numérique est une grille (on parle aussi parfois de *matrice*) composée de petits carrés colorés. Chaque cellule de la grille est appelée un *pixel* (contraction phonétique de l'anglais *P*icture *E*lement).

Pixel



Définition 2

La *définition* d'une image est le nombre de pixels qu'elle contient. Lorsque le nombre de pixels devient grand, on utilise les préfixes du système international, avec px comme abréviation pour « pixels ». Souvent, on donne un ordre de grandeur plutôt que la valeur exacte.

La *résolution* d'une image est, une fois l'image imprimée sur une feuille ou affichée sur un écran, le nombre de pixels par unité de longueur. La résolution s'exprime en PPP (*pixels par pouces*) (ou en anglais : DPI pour *dots per inch*).

Remarque : 1 pouce = 1 in = 2,54 cm.

Définition, résolution

Remarque 3

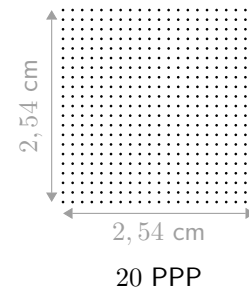
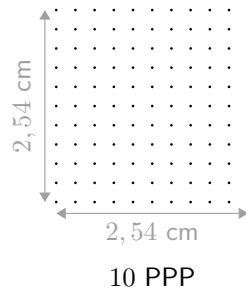
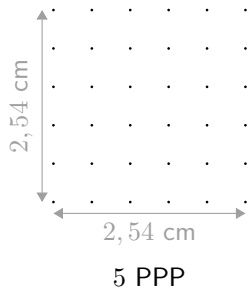
On confond souvent, à tort, la définition avec la résolution.

Exemple 4

- L'image donnée en illustration de la Définition 1 contient un total de $20 \times 20 = 400$ pixels, elle a donc une définition de 400 pixels.
- Une image en Full HD a pour dimensions 1 920 pixels de large et 1 080 pixels de haut, soit une définition de $1\,920 \times 1\,080 = 2\,073\,600$. Cette valeur est proche de 2,1 millions de pixels, on dira donc que sa définition est de 2,1 Mpx (ce qui, en réalité, est une approximation).
- Si on imprime l'image de la Définition 1 avec une résolution de 150 PPP, alors elle mesurera environ

3,4 mm sur la feuille. En effet, une résolution de 150 PPP signifie que 150 pixels mesurent 2,54 cm. Or l'image mesure 20 pixels de large, donc par un argument de proportionnalité, elle mesurera $\frac{20 \times 2,54}{150} \approx 0,338$ cm.

- Plus la résolution est élevée, plus l'on rentre un grand nombre de pixels par unité de longueur, et plus les points sont densément groupés. Dans les illustrations en taille réelle ci-dessous, chaque point représente un pixel, et leur espacement représente la résolution.



Exercice 5

- Une image mesure 1 280 px par 720 px. Quelle est sa définition ?
 - Une image de définition 5 197 500 px a pour largeur 1 350 px. Quelle est sa longueur ?
 - Une image carrée a une définition de 4 Mpx. Quelles sont ses dimensions ?
- Une image de dimensions 1 280 px par 720 px. Est affichée en plein écran avec une résolution de 150 PPP. Quelle est la taille de l'écran ?
 - Une image de dimensions 4 096 px par 2 160 px. Elle est affichée sur un écran de 60 cm de large. Quelle est la résolution de l'écran ?
 - Une image est imprimée sur une feuille, de sorte qu'elle mesure 19 cm de large et 27,7 cm de long, avec une résolution de 300 PPP. Quelle est la définition de l'image ?

2 – Représentations numériques de couleurs

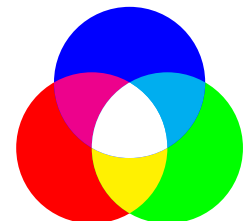
A – Encoder la perception humaine : le code RVB

La vue humaine est sensible à trois couleurs primaires : le rouge, le vert et le bleu (voir le chapitre suivant pour plus de détails). Afin de répliquer ce principe, l'on a inventé une manière de (re)produire les couleurs en indiquant l'intensité du Rouge, du Vert et du Bleu : c'est le code RVB (ou *RGB* en anglais).

On parle de *synthèse additive de la lumière* : en mélangeant les couleurs primaires, on obtient des couleurs secondaires, et en variant l'intensité de chaque source, on peut obtenir des variations. ⁽¹⁾

On retiendra les mélanges principaux, à la base du *cercle chromatique* :

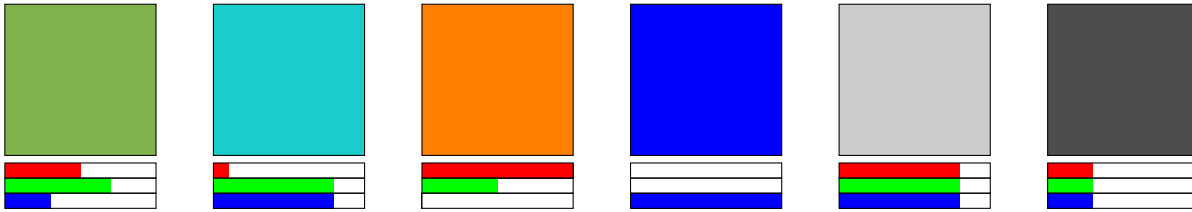
- Bleu + Vert → Cyan
- Rouge + Vert → Jaune
- Bleu + Rouge → Magenta
- Rouge + Vert + Bleu → Blanc



(1). le choix du rouge, vert et bleu comme couleurs primaires est arbitraire : on pourrait choisir d'autres couleurs, et on pourrait en prendre plus que trois. Cependant, il est impossible d'atteindre l'intégralité des couleurs visibles par synthèse additive de couleurs primaires. Voir l'article Wikipédia sur le [Gamut](#).

Exemple 6

Dans chaque exemple ci-dessous, les trois barres en bas représentent l'intensité de chaque couleur primaire : une barre remplie signifie que l'intensité est maximale, et une barre vide signifie que la couleur est absente.

**Information 7****Code RVB**

Le code RVB d'une couleur est la donnée de trois nombres : l'intensité du rouge, l'intensité du vert, et l'intensité du bleu.

Il y a plusieurs manières de donner ces nombres :

- ▶ Donner une valeur relative, comme un pourcentage ou son équivalent entre 0 et 1,
- ▶ Donner une valeur absolue, selon une échelle prédéterminée.

Information 8**Profondeur de couleur**

Dans sa version standard, le code RVB utilise 1 octet pour chaque couleur : autrement dit, chacun des trois nombres peut prendre 256 valeurs différentes, les entiers de 0 à 255.

Le nombre de bits alloués à la représentation d'une couleur est appelée *profondeur de couleur*. Avec une profondeur de 24 bits, on peut représenter plus de 16 millions de couleurs différentes.

Information 9**Base hexadécimale**

Pour représenter des entiers de 0 à 255, la *notation hexadécimale* est pratique. Son principe est le suivant : plutôt que s'arrêter aux chiffres de 0 à 9, on poursuit le compte avec les lettres *A*, *B*, jusqu'à *F*, afin d'avoir 16 « chiffres » les lettres étant maintenant considérées comme des chiffres) différents.

Afin d'éviter les confusions, il est d'usage de précéder les nombres hexadécimaux d'un croisillon. Par exemple, on écrit *#2B7E*.

De même que, une fois arrivé à 9, on compte 1 dizaine et 0 unités et on écrit 10 ; en base hexadécimale, une fois arrivé à *#F*, on compte 1 « seizaine » (puisqu'on a épuisé 16 unités) et 0 unités, et on écrit *#10* (qui vaut donc 17).

Exemple 10

Voici les premiers nombres comptés en base hexadécimale :

#0	#1	#2	#3	#4	#5	#6	#7	#8	#9	#A	#B	#C	#D	#E	#F
#10	#11	#12	#13	#14	#15	#16	#17	#18	#19	#1A	#1B	#1C	#1D	#1E	#1F
#20	#21	#22	#23	#24	#25	#26	#27	#28	#29	#2A	#2B	#2C	#2D	#2E	#2F
⋮															
#A0	#A1	#A2	#A3	#A4	#A5	#A6	#A7	#A8	#A9	#AA	#AB	#AC	#AD	#AE	#AF
#B0	#B1	#B2	#B3	#B4	#B5	#B6	#B7	#B8	#B9	#BA	#BB	#BC	#BD	#BE	#BF
⋮															
#F0	#F1	#F2	#F3	#F4	#F5	#F6	#F7	#F8	#F9	#FA	#FB	#FC	#FD	#FE	#FF

Information 11**Code hexadécimal ou HTML**

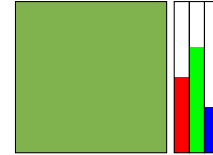
Un nombre sur 1 octet peut être représenté en base hexadécimale à l'aide de deux « chiffres ». Un code RVB est donc représenté par 6 « chiffres » : les deux premiers représentent le niveau de rouge, les deux suivants le vert, et les deux derniers le bleu.

#2AEC05

Exemple 12

Le code RVB de la couleur ci-contre peut être donné des manières suivantes :

- ▶ (0,5 ; 0,7 ; 0,3)
- ▶ (128 ; 179 ; 77)
- ▶ (50% ; 70% ; 30%)
- ▶ #80B34F

**Exercice 13**

Associer à chaque couleur de l'Exemple 6 le ou les bons codes RVB parmi les suivants (on pourra s'aider d'un outil de type « Color picker » en ligne – conseillé : <https://esuong-maths.fr/snt-rgb-picker.html>) :

- | | | | |
|----------------------|---------------------|------------|------------|
| a▶ (0 ; 0 ; 1) | c▶ (26 ; 204 ; 204) | e▶ #0000FF | g▶ #4D4D4D |
| b▶ (10% ; 80% ; 80%) | d▶ #1ACCCC | f▶ #CCCCCC | h▶ #3C823C |

Exercice 14

- 1▶ Calculer le nombre exact de couleurs que l'on peut distinguer avec une profondeur de 24 bits.
- 2▶ Calculer la division euclidienne de 155 par 16. En déduire que 155 s'écrit #9B en hexadécimal.
- 3▶ Justifier que #C4 est l'écriture hexadécimale de 196.

Activité 15 (Pixel art)

- 1▶ Dans un projet Python (par exemple, un Notebook sur [Basthon](#)), commencez par copier-coller le premier code en [annexe](#) de ce chapitre.
- 2▶ Choisir l'un des pixel art en [annexe](#) que vous souhaitez reproduire.
- 3▶ Remplacer A, ..., F dans le code par les valeurs associées au pixel art que vous avez choisi (certains pixel arts ont moins de couleurs).
- 4▶ Identifier à quelles couleurs A, ..., F correspondent.
- 5▶ Remplacer chaque case de matrice par A, ..., F pour colorer le pixel correspondant de la bonne couleur.
- 6▶ Une fois le pixel art terminé, essayez de changer les valeurs de A, ..., F pour voir votre œuvre changer de couleur !

Activité 16 (Pixel art bis)

Vous trouverez en [annexe](#) deux codes permettant de dessiner une image en pixel art. Votre travail : identifier le motif, et trouver les codes couleur afin de reproduire l'une des images. Attention, pour la première image, on cherche les codes RVB des couleurs, et pour l'image 2, on cherche les codes hexadécimaux !

B – Encoder la sensation humaine : le code TSL

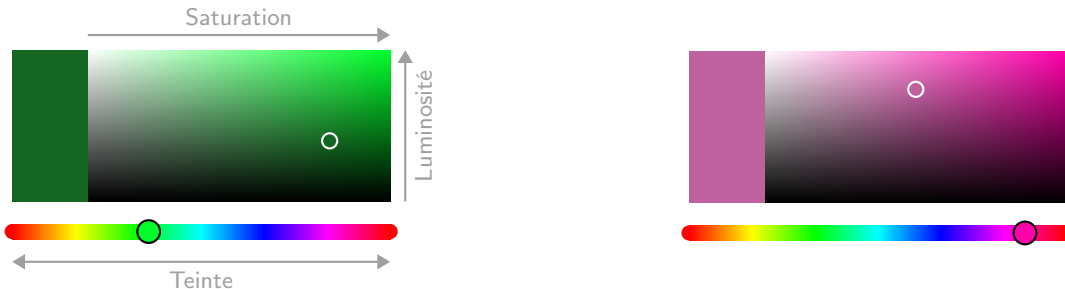
Là où le code RVB transcrit la réalité biologique derrière la perception des couleurs, le code TSL (en anglais : *HSB*) vise à transcrire la sensation humaine des couleurs. Si on demande à quelqu'un de décrire un couleur, elle ne donnera pas les niveaux de rouge, de vert et de bleu : elle décrira plutôt la (famille de) couleur(s) qu'elle voit, à quelle point la couleur est vive ou terne, et à quel point l'image qu'elle perçoit est brillante ou sombre. Ainsi, le code TSL décrit les couleurs avec 3 valeurs :

- ▶ la *teinte* (T) : c'est la couleur que l'on perçoit (plutôt dans les rouges, dans les mauves, les cyan, ...)
- ▶ la *saturation* (S) : c'est l'intensité de la couleur (si la couleur est vive, alors la saturation est élevée; si la couleur est terne, délavée, alors la saturation est faible)
- ▶ la *luminosité* (L) : c'est l'intensité de la lumière en elle-même, autrement dit à quel point la couleur est claire ou sombre.

Remarque 17

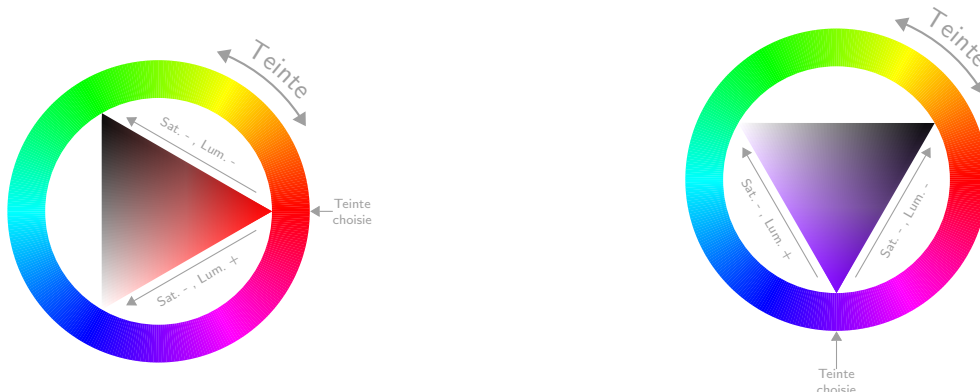
L'on entend souvent que le noir, le blanc et le(s) gris ne sont pas des couleurs : le code TSL rend compte de cet adage. En effet, on obtient du noir, du blanc ou du gris lorsque la saturation est à 0 (ce qui signifie une couleur d'intensité nulle, d'où la soi disant *absence* de couleur). La différence entre le blanc, le noir et le(s) gris réside dans la luminosité : le blanc représente la luminosité la plus forte, le noir l'absence de luminosité, et l'échelle de gris se situe entre les deux.

En tapant « Color picker » dans le moteur de recherche Google, on tombe sur un application ressemblant à celle ci-dessous. Une échelle permet de choisir la teinte, et un rectangle affiche toutes les variations de saturation (horizontalement) et de luminosité (verticalement) possibles : plus on se place à droite, plus la couleur est vive, et plus on se place à gauche, plus elle est terne; plus on se place vers le haut, plus la couleur est claire, et plus on va vers le bas, plus la couleur est foncée.



Remarquez comme les deux extrémités de la barre des teintes sont de la même couleur : pour cette raison, il arrive que l'on représente cette échelle le long d'un cercle; on parle alors de *cercle chromatique*.

Remarquez également qu'à luminosité minimale (le bas du rectangle), la couleur est noire, indépendamment de la saturation. Pour cette raison, on « regroupe » parfois la ligne du bas en un point, ce qui donne un triangle.

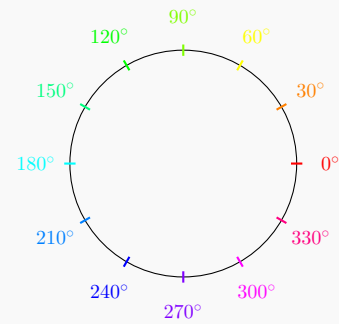


Code TSL

Information 18

Le *code TSL* d'une couleur est la donnée de trois nombres :

- La teinte est donnée comme pourcentage ou son équivalent entre 0 et 1, ou comme un angle entre 0° et 360° (selon le cercle chromatique),
- La saturation et la luminosité sont données comme pourcentage, ou leur équivalent entre 0 et 1.



C – Autres encodages

Les codes RVB et TSL ne sont pas les seules manières d'encoder des couleurs. Voyons plusieurs exemples avec l'image suivante comme support :



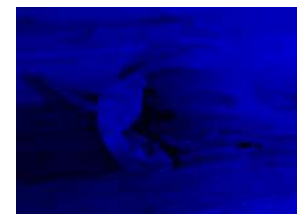
Comme nous l'avons vu, le code RVB reconstruit les images en superposant des niveaux de rouge, de vert et de bleu :



Canal rouge

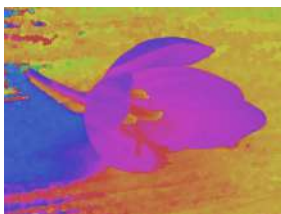


Canal vert

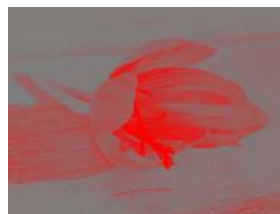


Canal bleu

Le code TSL quant à lui exploite d'autres données :



Teinte



Saturation



Luminosité

On apprend au collège le principe de synthèse négative : lorsqu'on mélange des peintures ou l'encre d'une imprimante, on peut obtenir de nouvelles couleurs. Les couleurs primaires sont le cyan, le magenta et le jaune, le noir permet d'affiner la clarté d'une couleur. On utilise en général l'acronyme anglais *CMYK* (Cyan, Magenta, Yellow, black). Explorez YouTube, qui regorge de démonstrations de comment la superposition de filtres colorés permet de reconstruire une image !



Canal cyan



Canal magenta



Canal jaune



Canal noir

Autre méthode très répandue, le code Y'CbCr. Son principe repose sur le fait que l'œil humain distingue bien mieux les niveaux de luminosité que les couleurs en elles-mêmes (voir le chapitre suivant pour plus de détails). Ainsi, le canal Y' (luma) contient l'image en niveaux de gris, et les canaux Cb et Cr (chrominance) contiennent toute l'information colorée. Ce code est utilisé par le format JPEG qui l'utilise astucieusement pour compresser les images en sacrifiant les informations les moins perceptibles. C'était également la méthode standard choisie par l'industrie audiovisuelle, notamment durant la transition vers les écrans en couleur : en effet, à une période où certains foyers ne disposaient que d'un téléviseur en noir et blanc, le format Y'CbCr permettait de transmettre les images à tous les appareils. Les téléviseurs noir et blanc ne conservaient que le canal Y', et les téléviseurs couleur exploitaient les trois canaux.



Canal Y'



Canal Cb



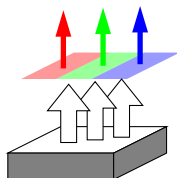
Canal Cr

Certains format d'image rajoutent une information : la transparence, souvent dénotée par un a ou un α . Ce nouveau canal, prenant la forme d'un quatrième nombre dans le code, indique l'opacité d'un pixel : une opacité de 1 signifie que le pixel est complètement opaque, une opacité de 0 signifie que le pixel est invisible, une opacité de 0,5 signifie que le pixel est à moitié transparent, etc. Par exemple, le format PNG est réputé pour supporter cette information de transparence, contrairement au JPEG.

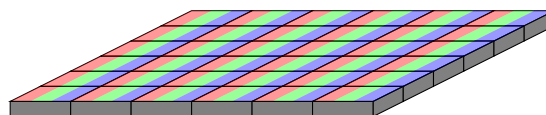
3 – Le fonctionnement des écrans

Depuis le premier écran couleur jusqu'à aujourd'hui, les technologies contenues dans les écrans reposent sur deux composantes : une source de lumière, et une source de couleurs (dans les technologies les plus récentes, ces deux composantes sont fusionnées ; plus de détails dans la suite). Les sources colorées sont toujours du rouge, du vert et du bleu : quel que soit le format de l'image envoyée à l'écran, elle est tôt ou tard convertie au format RVB, qui dicte l'intensité de la lumière de chacune des trois couleurs primaires, et ce pour chaque pixel.

La première famille est celle des technologies dites « transmissives » : une source de lumière blanche passe à travers des filtres rouges, verts et bleus pour former les couleurs primaires. Chaque pixel est décomposé en trois morceaux, chacun pour une couleur primaire, et dispose donc de trois mini-filtres individuels. Pour former une couleur donnée, un pixel règle la quantité de lumière filtrée par chaque couleur, de la filtration minimale (qui donne la couleur la plus vive) à la filtration maximale (la couleur ne passe pas du tout).



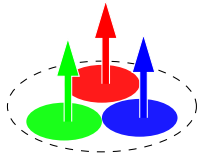
Un pixel



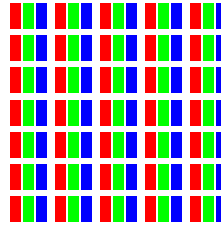
Une grille de pixels

L'autre grande famille est celle des technologies dites « émissives », où chaque pixel est constitué de trois LEDs, rouge, verte et bleue. Cela permet de construire des pixels plus petits, et donc obtenir une meilleure résolution.

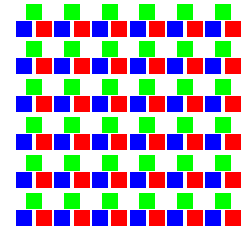
Émettre directement les couleurs sans passer par un filtre permet également d'avoir des couleurs encore plus vives, car les filtres ont tendance à ternir les couleurs. Dernier avantage, la couleur noire est meilleure en tous points. En effet, le noir étant l'absence de couleurs, il suffit d'éteindre les trois LEDs pour faire un pixel noir, là où dans les technologies transmissives, il faut bloquer complètement la source de lumière (à moins que celle-ci puisse être gérée indépendamment pour chaque pixel, ce qui est rarement le cas, sauf dans les technologies les plus récentes). Éteindre plutôt que bloquer permet également de réaliser des économies d'énergie.



Chaque pixel est constitué de trois LEDs colorées



Une grille de pixels



Une autre disposition possible des LEDs dans chaque pixel

Annexes

Annexe pour l'Activité 15 – copiez les deux blocs de code ci-dessous, et remplacez les variables de A à F par les codes RVB donnés sous le dessin que vous avez choisi.

```
from ipythonblocks import BlockGrid

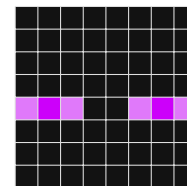
def mystere(x): return tuple(int(x[1:][i:i+2], 16) for i in (0, 2, 4))

def dessiner(matrice):
    nb_lignes = len(matrice)
    nb_colonnes = len(matrice[0])
    grille = BlockGrid(nb_colonnes, nb_lignes, block_size=30)
    for ligne in range(nb_lignes):
        for colonne in range(nb_colonnes):
            couleur = matrice[ligne][colonne]
            if isinstance(couleur, str):
                couleur = mystere(couleur)
            grille[ligne, colonne] = couleur
    return grille
```

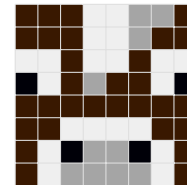
```
A = (... , ... , ...) # Couleur 1 à remplacer selon le dessin
B = (... , ... , ...) # Couleur 2 à remplacer selon le dessin
C = (... , ... , ...) # Couleur 3 à remplacer selon le dessin
D = (... , ... , ...) # Couleur 4 à remplacer selon le dessin
E = (... , ... , ...) # Couleur 5 à remplacer selon le dessin
F = (... , ... , ...) # Couleur 6 à remplacer selon le dessin
```

```
matrice = [
    [... , ... , ... , ... , ... , ... , ... , ...],
    [... , ... , ... , ... , ... , ... , ... , ...],
    [... , ... , ... , ... , ... , ... , ... , ...],
    [... , ... , ... , ... , ... , ... , ... , ...],
    [... , ... , ... , ... , ... , ... , ... , ...],
    [... , ... , ... , ... , ... , ... , ... , ...],
    [... , ... , ... , ... , ... , ... , ... , ...],
    [... , ... , ... , ... , ... , ... , ... , ...],
    [... , ... , ... , ... , ... , ... , ... , ...],
]
```

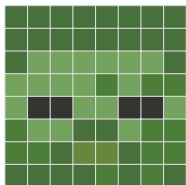
```
dessiner(matrice)
```



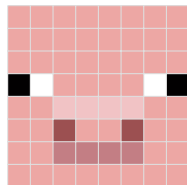
A = (18, 18, 18)
B = (224, 121, 250)
C = (204, 0, 250)



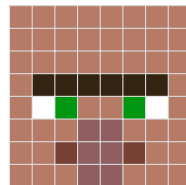
A = (0, 0, 8)
B = (57, 24, 0)
C = (165, 165, 165)
D = (239, 239, 239)



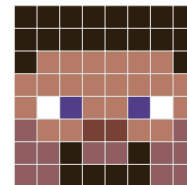
A = (72, 112, 58)
B = (76, 126, 58)
C = (102, 136, 60)
D = (116, 163, 95)
E = (52, 53, 48)



A = (237, 167, 164)
B = (255, 255, 255)
C = (2, 0, 1)
D = (156, 80, 81)
E = (241, 195, 197)
F = (195, 126, 132)

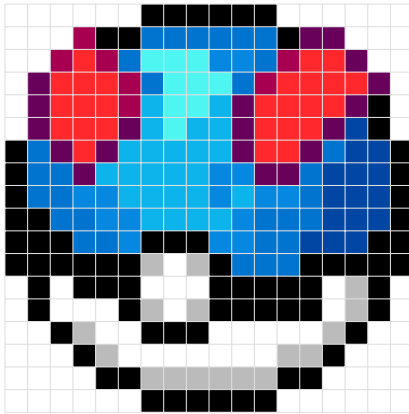


A = (181, 123, 103)
B = (255, 255, 255)
C = (51, 36, 17)
D = (0, 150, 17)
E = (144, 94, 97)
F = (119, 66, 53)



A = (181, 123, 103)
B = (255, 255, 255)
C = (44, 30, 14)
D = (82, 61, 137)
E = (144, 94, 97)
F = (119, 66, 53)

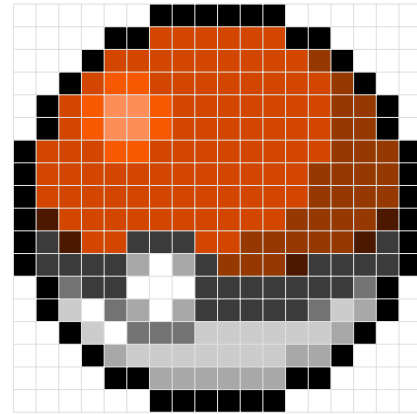
Annexe pour l'activité 16 –  commencez par copier le début du code de l'activité précédente (le premier bloc), puis rajoutez le code plus bas selon le dessin que vous choisissez de faire.



```
A = (79, 245, 241) # Couleur 1
B = (... , ... , ...) # Couleur 2
C = (... , ... , ...) # Couleur 3
D = (... , ... , ...) # Couleur 4
E = (... , ... , ...) # Couleur 5
F = (... , ... , ...) # Couleur 6
G = (... , ... , ...) # Couleur 7
H = (... , ... , ...) # Couleur 8
I = (... , ... , ...) # Couleur 9
J = (... , ... , ...) # Couleur 10
K = (... , ... , ...) # Couleur 11
```

```
matrice = [
    [I,I,I,I,I,I,K,K,K,K,K,I,I,I,I,I,I],
    [I,I,I,G,K,K,D,D,D,D,D,K,H,H,I,I,I],
    [I,I,G,F,G,D,A,A,A,C,C,D,G,F,F,H,I,I],
    [I,H,F,F,F,G,D,A,A,A,D,G,F,F,F,H,I,I],
    [I,H,F,F,F,G,B,A,A,B,H,F,F,F,H,K,I,I],
    [I,H,F,F,F,H,B,A,B,B,H,F,F,F,H,E,K,I],
    [K,D,H,F,H,B,B,B,B,B,C,H,H,D,E,E,K],
    [K,D,D,H,B,B,B,B,B,C,C,H,H,D,E,E,K],
    [K,D,D,C,C,B,B,B,B,C,C,D,E,E,E,K],
    [K,K,D,D,C,C,B,B,B,B,C,D,D,D,E,E,E,K],
    [K,K,K,D,D,C,K,K,K,C,C,D,D,E,E,E,K],
    [K,K,K,K,K,K,J,I,J,K,D,D,D,K,K,K,K],
    [I,K,I,K,K,K,I,I,I,K,K,K,K,I,J,K,I],
    [I,K,I,I,I,K,J,I,J,K,K,K,K,I,J,K,I],
    [I,I,K,J,I,I,K,K,K,I,I,I,I,J,K,I,I],
    [I,I,I,K,J,I,I,I,I,I,I,I,J,J,K,I,I],
    [I,I,I,I,K,K,J,J,J,J,J,K,K,I,I,I,I],
    [I,I,I,I,I,I,K,K,K,K,K,I,I,I,I,I,I]
]

dessiner(matrice)
```



```
A = "#FD8E57" # Couleur 1
B = "#....." # Couleur 2
C = "#....." # Couleur 3
D = "#....." # Couleur 4
E = "#....." # Couleur 5
F = "#....." # Couleur 6
G = "#....." # Couleur 7
H = "#....." # Couleur 8
I = "#....." # Couleur 9
J = "#....." # Couleur 10
K = "#....." # Couleur 11
```

```
matrice = [
    [F,F,F,F,F,K,K,K,K,K,F,F,F,F,F],
    [F,F,F,F,K,K,C,C,C,C,C,K,K,F,F,F],
    [F,F,F,K,C,C,C,C,C,C,C,C,D,K,F,F],
    [F,F,K,C,B,B,C,C,C,C,C,C,C,D,K,F],
    [F,K,C,B,A,A,B,C,C,C,C,C,C,D,D,K,F],
    [F,K,C,B,A,A,B,C,C,C,C,C,C,D,D,K,F],
    [K,C,C,C,B,B,C,C,C,C,C,C,C,D,D,D,K],
    [K,C,C,C,C,C,C,C,C,C,C,C,C,D,D,D,K],
    [K,C,C,C,C,C,C,C,C,C,C,C,C,D,D,D,K],
    [K,E,C,C,C,C,C,C,C,C,C,C,D,D,D,E,K],
    [K,J,E,C,C,J,J,J,C,C,D,D,D,D,E,J,K],
    [K,J,J,J,J,H,F,H,J,D,D,D,E,J,J,J,K],
    [F,K,I,J,J,F,F,F,J,J,J,J,J,I,K,F],
    [F,K,G,F,I,H,F,H,J,J,J,J,I,G,H,K,F],
    [F,F,K,G,F,I,I,I,G,G,G,G,G,H,K,F,F],
    [F,F,F,K,H,G,G,G,G,G,G,H,H,K,F,F,F],
    [F,F,F,F,K,K,H,H,H,H,H,K,K,F,F,F,F],
    [F,F,F,F,F,F,K,K,K,K,K,K,F,F,F,F,F]
]

dessiner(matrice)
```

6 Autour de la photographie

Table des matières

1	Photographie chimique	75
2	Photographie électronique	76
A	Repères historiques	76
B	Capter la réalité : de l'œil à l'appareil photo	77
C	Régler un appareil photo	80
D	De la capture au fichier	87
3	Traitement d'images	88
4	Les enjeux sociétaux liés aux images	88

Ce cours est inspiré des ressources du site [Enseigner les SNT](#).

1 – Photographie chimique

En 1816, Joseph Nicéphore Niépce débute ses recherches sur la photographie, à l'aide de procédés dits « héliographiques ». Il s'est servi de bitume de Judée, un pigment utilisé en peinture pour produire des teintes marron foncé, et qui a la propriété de réagir à la lumière. On considère que la première photographie héliographique a été réalisée en 1822. malheureusement, la technique développée par Niépce est encore expérimentale : le pigment se dégrade et la photo disparaît avec le temps. La plus ancienne photographie permanente, c'est-à-dire encore conservée jusqu'à aujourd'hui, date de 1827, et s'intitule *Point de vue du Gras*. On estime que le temps de pose nécessaire à la capture de cette image était de plusieurs jours. Louis Daguerre invente avec Niépce en 1839 le daguerréotype, qui permet de fixer une image sur une plaque recouverte d'argent (d'où l'appellation « photographie argentique »). Le procédé ne nécessite qu'une vingtaine de minutes de pose, et les images produites sont ensuite fixées grâce à un bain de mercure, leur permettant de durer dans le temps.

En parallèle, William Henry Fox Talbot développe en 1840 un autre procédé de traitement de l'argent, dit « calotype ». Son exploitation est cependant drastiquement opposée à celle du daguerréotype : en effet, après une exposition de quelques secondes à quelques minutes, l'image obtenue sur la plaque argentée est un *négatif* de l'image réelle, c'est-à-dire que les zones les plus lumineuses dans la réalité sont les plus sombres sur le négatif, et inversement. L'on pose ensuite le papier sur lequel on souhaite imprimer l'image sur le négatif, et après un traitement chimique (on dit que l'on *développe* la photo), l'image positive est fixée sur le papier. L'intérêt majeur de ce procédé est que l'on peut ré-exploiter le négatif, et ainsi imprimer autant de positif que l'on souhaite, là où le daguerréotype ne pouvait produire qu'un positif à la fois. Frederick Scott Archer raffine le procédé de développement en 1851, avec une nouvelle technique, la *collodion humide*. Malgré certains inconvénients, notamment le fait qu'il faut développer l'image rapidement après sa prise, cette méthode permet d'obtenir des images plus nettes et plus contrastées. Cette technique a été largement utilisée pendant plusieurs décennies. Jusqu'en 1861, les clichés étaient exclusivement en noir et blanc, les grains d'argent ne permettant que de capter les différences dans l'intensité de la lumière. À cette date, James Clerk Maxwell invente une méthode, dite « trichromique », permettant enfin la réalisation de photos en couleur à l'aide de trois couleurs primaires. Les travaux de Maxwell ont conduit à l'invention en 1903 par les frères Auguste et Louis Lumière de la méthode *autochrome* permettant de photographier en couleur une scène sur une unique plaque de verre.

En 1871, Richard Leach Maddox améliore le procédé de développement et contribue à le rendre plus accessible. Cependant, la photographie reste un procédé complexe : il faut disposer d'un appareil photo, qui est souvent un objet imposant, disposer de plaques (alors en verre pour la plupart) pour capturer un négatif, puis faire développer ces négatifs... Toute cette chaîne logistique se simplifie en 1888 lorsque George Eastman, fondateur de l'entreprise Kodak,

lance le premier appareil photo portable et simple d'utilisation. Les plaques sont remplacées par une pellicule enroulée dans un tube, que les photographes expédient à Kodak une fois la pellicule épuisée, lesquels renvoient alors les images développées. Kodak poursuivra le développement d'appareils toujours plus accessibles, comme sa série Brownie vendu à partir de 1900 qui a contribué à la popularisation de la photographie. Les premières pellicules mesuraient 70 mm de large; à partir de 1927, avec le Leica I d'Oskar Barnack, la pellicule de 35 mm s'impose comme un format standard pour la photographie. Son format plus petit offre plusieurs avantages : les appareils photos peuvent être rendus plus compacts, et les pellicules peuvent prendre plus de photos et coûtent moins cher à produire. De plus, Kodak a développé un conditionnement de ces rouleaux permettant de les charger dans les appareils même en plein jour : car en effet, la pellicule ne doit être exposée à aucune lumière en amont de la photographie, au risque de la détériorée (puisque c'est justement la lumière lors de la prise de vue qui imprime l'image sur la pellicule). La majeure contrepartie de ce rétrécissement de la pellicule et que lors du développement, l'image doit être agrandie, ce qui détériore sa résolution. Il faudra attendre jusqu'en 1935 pour que les pellicules photographiques en couleur soient commercialisées par Kodak avec leur appareil, le Kodachrome (bien que les frères Lumière aient déjà adapté leur technique autochrome sur pellicule quelques années auparavant, ce qui resta très marginal commercialement parlant).

Jusqu'alors, les pellicules devaient encore être expédiée puis renvoyées après développement, rendant le processus long. Il faudra attendre 1948 pour que Polaroid lance le premier appareil photo instantané, duquel la photo s'imprime instantanément après avoir été prise.

Activité 1 (Frise chronologique 1)

(Cette activité se poursuit dans l'Activité 3.)

À l'aide du texte précédent, réaliser une frise chronologique sur laquelle apparaîtront (au minimum) les éléments suivants :

- | | | |
|----------------------------|-----------------------------------|--|
| ▶ Louis Daguerre | ▶ Le daguerréotype | ▶ L'autochrome |
| ▶ William Henry Fox Talbot | ▶ Le calotype | |
| ▶ Louis Ducos du Hauron | ▶ Trichromie | ▶ Naissance de la photographie argentique |
| ▶ George Eastman | ▶ Premier appareil photo portable | |
| ▶ Nicéphore Niépce | | ▶ Développement de la photographie couleur |
| ▶ Louis Lumière | ▶ L'héliographie | |

Pour chaque inventeur et chaque invention qui lui est associée, découpez et collez les images en [annexe](#) pour décorer votre frise (vous pouvez également ajouter d'autres décorations, dessins ou photos!).

Pour chaque invention, ajoutez une courte phrase expliquant en quoi elle consiste.

Exercice 2

Sur les plaques argentiques, l'argent qui sert à capter la lumière se regroupe en petits grains, qui sont semblables à des pixels. On peut donc parler de résolution d'une image prise en photographie argentique.

- 1▶ Calculer la résolution si les grains mesurent 20 μm .
- 2▶ Calculer la résolution si les grains mesurent 4 μm .

2 – Photographie électronique

A – Repères historiques

En 1957, le Dr. Russel A. Kirsch et son équipent créent le premier scanner électronique d'images, qui est considéré comme le premier pas vers la numérisation des images. Deux décennies plus tard, en 1970, le premier capteur CCD (le constituant d'un appareil photo responsable de la numérisation des images) est inventé, et les premières caméras numériques voient par la suite. Ces dernières étant encore expérimentales et très coûteuses, celles-ci sont réservées à des applications professionnelles et militaires. En 1974, le premier capteur CCD couleur est inventé par Kodak, et

utilisé dès 1975 par Steven Sasson dans ce que l'on considère être le premier appareil photo électronique moderne. Le grand public doit attendre une décennie supplémentaire pour voir arriver sur le marché la Mavica, contraction de *Magnetic Video Camera*, en 1981, qui stocke les images sur des disquettes. Les premiers capteurs CCD avaient une définition de 10 000 pixels, autrement dit 0,01 Mpx. C'est à nouveau Kodak qui révolutionnera l'industrie en 1986 en dévoilant son capteur CCD d'une définition de 1,4 Mpx, améliorant considérablement la qualité de l'imagerie électronique.

Jusqu'alors, les appareils photo traitent le signal issu du capteur CCD de manière analogique, souvent à l'aide de bandes magnétiques, comme les cassettes ou les disquettes. En 1988, Fuji lance l'AP-10, premier appareil photo électronique grand public à stocker les images sur une carte mémoire, avec une capacité de 32 images, traitait le signal de manière analogique. En 1994, Kodak commercialise le DC40, premier appareil photo à traiter le signal de manière numérique, que l'on appellera alors *appareil photo numérique*. L'appareil photo n'a depuis eu de cesse d'évoluer, et de gagner en fonctionnalités. En 1995, Casio introduit le premier appareil photo numérique compact grand public, le Casio QV-10, qui dispose d'un écran pour visualiser directement depuis l'appareil les clichés qui viennent d'être pris. En 2000, Nikon lance le D1, premier appareil photo numérique reflex (c'est-à-dire disposant d'un système particulier d'ocilleton, permettant de prévisualiser fidèlement la photo avant de la prendre) professionnel, marquant le début de l'ère de la photographie professionnelle. En 2003, Canon dévoile l'EOS 300D, rendant accessible la technologie reflex au grand public.

La photographie connaît sa plus grande révolution en 2007, lorsque Apple révèle le premier smartphone doté d'un appareil photo numérique intégré, permettant à tout un chacun de prendre des photos de qualité. Le lancement d'Instagram en 2010 marque le début de la culture du partage de photos, renforçant encore plus l'attrait des consommateurs pour la photographie mobile.

Activité 3 (Frise chronologique 2)

(Cette activité est la suite de l'Activité 1.)

À l'aide du texte ci-dessus et des illustrations en [annexe](#), compléter la frise chronologique de la précédente activité en ajoutant (au minimum) les éléments suivants :

- ▶ Steven Sasson
- ▶ Premier capteur CCD
- ▶ Premier appareil photo numérique
- ▶ Premier logiciel de retouche photo (Photoshop)
- ▶ Premier smartphone avec appareil photo intégré

B – Capturer la réalité : de l'œil à l'appareil photo

L'anatomie d'un appareil photo numérique est très similaire à celle d'un œil. Étudions les deux pour mettre en lumière leurs points communs.

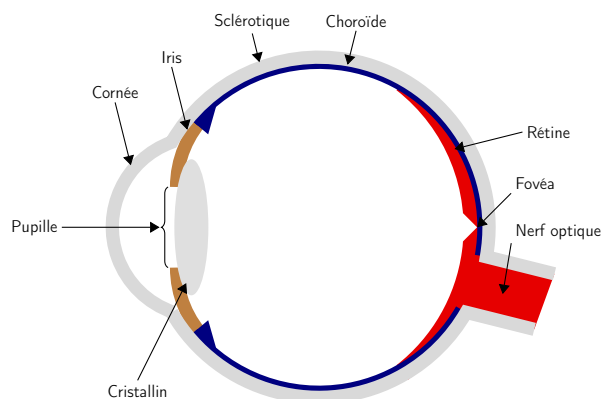


FIGURE 1 : Schéma de l'anatomie d'un œil humain

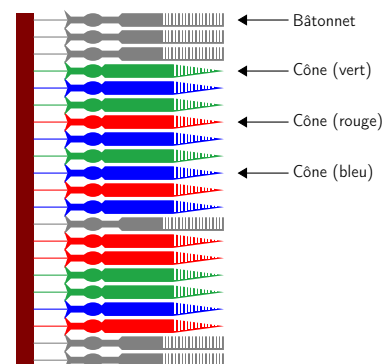


FIGURE 2 : Schéma de la rétine

L'œil humain utilise la cornée et le cristallin pour focaliser la lumière sur la rétine. Le cristallin peut se déformer pour

courber plus ou moins fortement les rayons lumineux, en fonction de la distance de l'objet regardé (c'est ce que l'on appelle l'*accommodation*). L'iris recouvre partiellement le cristallin, et peut se dilater ou se contracter pour régler la taille de l'ouverture (la pupille), afin de laisser passer plus ou moins de lumière. Une fois arrivée sur la rétine, la lumière interagit avec les *cellules photoréceptrices* de la rétine : les *bâtonnets* et les *cônes*. Les bâtonnets sont sensibles à l'intensité de la lumière, et les cônes sont sensibles à la couleur. Il existe trois types de cônes : un type sensible à la lumière rouge, un type sensible à la lumière verte, et un type sensible à la lumière bleue. Chaque cellule photoréceptrice envoie un signal électrique *via* le nerf auquel elle est relié, et ces nerfs se rejoignent pour former le nerf optique, qui envoie toutes les informations au cerveau. En un sens, l'œil ne « voit » pas : il ne fait rien de plus que traduire la lumière en signaux électrique, et c'est ensuite le cerveau qui interprète ces signaux comme des images.

Les cônes et les bâtonnets sont répartis plus ou moins aléatoirement sur la rétine : il existe cependant un point particulier, le fovéa, dans lequel il n'y a que des cônes. C'est la zone de la rétine avec la plus grande acuité visuelle, c'est-à-dire que la zone de l'image qui tombe dans le fovéa sera vue avec la plus grande netteté. Elle correspond à ce qui est directement dans l'axe de la vision. Plus on s'éloigne du fovéa, plus les cônes se font rares. Il y a également un autre point remarquable : l'endroit où tous les nerfs se rejoignent pour former le nerf optique. Aucune cellule photoréceptrice ne se trouve à cet endroit là, et donc la zone de l'image qui tombe à cet endroit n'est pas vue par l'œil ; d'où son nom, la *tache aveugle*. En temps normal, le cerveau comble ce trou inconsciemment, mais il existe [une expérience très simple](#) pour « voir » cette tache aveugle !

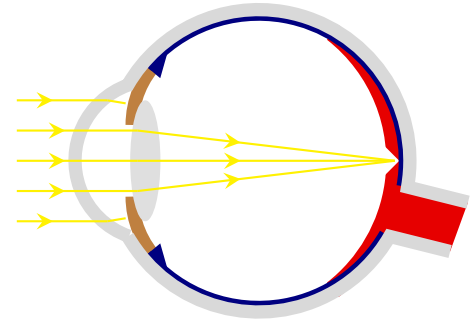


FIGURE 3 : Trajet de la lumière dans l'œil

Un œil humain est composé d'environ 5 millions de cônes et 95 millions de bâtonnets. Chaque cellule photoréceptrice fait environ $2\text{ }\mu\text{m}$ de diamètre, soit une résolution de 12 700 pixels par pouce. Hors exceptions (daltonisme ou tétrachromie), l'être humain peut distinguer environ 300 000 teintes, et 10 millions de couleurs (teintes + saturation + luminosité).

Les appareils photo, argentiques comme électroniques, fonctionnent de manière très similaire. Un ensemble de lentilles jouent le rôle du cristallin et de la cornée, et focalisent l'image sur le capteur CCD, qui joue le rôle de la rétine. L'équivalent des cellules photoréceptrices sont les cellules de la grille composant le capteur CCD, que l'on appelle *photosites*. Les photosites convertissent l'information lumineuse en signal électrique, qui est ensuite envoyé vers la partie de l'appareil qui s'occupe de reconstituer l'image, équivalent du cerveau.

La majeure partie de la différence entre un œil et un appareil photo réside dans la discrimination des informations de couleur : là où l'œil humain dispose de quatre types de cellules, les appareils photo s'en contentent de trois : un type pour chaque couleur primaire. Cependant, un photosite seul ne peut pas distinguer les couleurs : il ne peut rien faire de plus que transcrire à quel point la lumière qu'il reçoit est intense ou faible. Pour séparer les couleurs du signal lumineux, une *matrice de Bayer* est placée au-dessus des photosites : il s'agit d'une grille de filtres rouges, verts et bleus. Ainsi, à chaque photosite est assignée une couleur, et ils mesurent l'intensité de cette couleur uniquement.

La vision humaine étant plus sensible au vert qu'au rouge et au bleu, la matrice de Bayer est arrangée de sorte à former des carrés de quatre photosites, deux verts, un rouge et un bleu. Ces carrés de quatre photosites constituent les différents pixels de l'image finale.

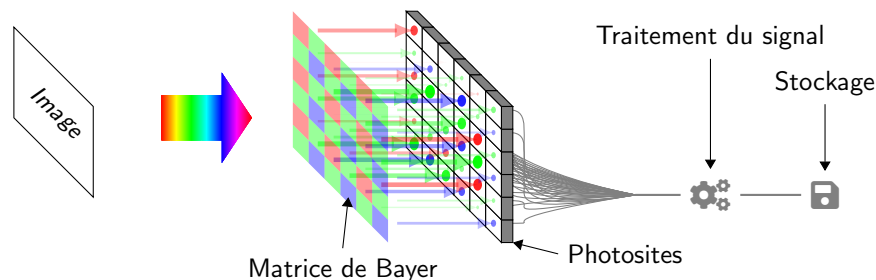


FIGURE 4 : Trajet de la lumière dans un appareil photo – photosites et matrice de Bayer

Exercice 4

À l'aide des informations précédentes, répondre au QCM suivant :

- | | |
|---|--|
| <p>1► Quand la photographie argentique est-elle née ?</p> <p>a► 1847 b► 1950 c► 1826 d► 1990</p> <p>2► Qui a créé le premier daguerréotype ?</p> <p>a► William Henry Fox Talbot
b► Louis Daguerre
c► Nicéphore Niépce
d► Georges Eastman</p> <p>3► Quelle invention est associée à Louis Ducos du Hauron ?</p> <p>a► Le calotype
b► Le daguerréotype
c► Le principe trichromique
d► L'appareil photo portable Kodak</p> <p>4► Quel inventeur est lié à l'invention du premier appareil photo numérique ?</p> <p>a► Louis Daguerre
b► William Henry Fox Talbot
c► George Eastman
d► Steven Sasson</p> <p>5► En quelle année l'iPhone, premier smartphone tactile généralisant la photo numérique, a-t-il été lancé ?</p> <p>a► 1990 b► 2007 c► 1950 d► 1986</p> <p>6► De combien de types de cellules photoréceptrices l'œil humain normal dispose-t-il ?</p> <p>a► 1 b► 2 c► 3 d► 4</p> <p>7► Quels sont les deux types de cellules photoréceptrices ?</p> <p>a► Cônes et bâtonnets
b► Tiges et cornets
c► Photosites et capteurs
d► Photodiodes et convertisseurs</p> | <p>8► À quoi sont sensibles les cônes dans la rétine ?</p> <p>a► À l'intensité de la lumière
b► Au fovéa
c► Aux couleurs
d► À rien</p> <p>9► Quelles sont les trois couleurs primaires utilisées dans la synthèse additive des couleurs ?</p> <p>a► Rouge, vert, bleu
b► Cyan, Magenta, Jaune
c► Bleu, Blanc, Rouge
d► Rouge, Jaune, Bleu</p> <p>10► En synthèse additive, qu'obtient-on en mélangeant du rouge et du vert en proportions égales ?</p> <p>a► Magenta c► Cyan
b► Jaune d► Blanc</p> <p>11► Comment les photosites enregistrent-ils l'intensité de la lumière en photographie numérique ?</p> <p>a► En produisant de la lumière
b► En convertissant de la lumière en son
c► En générant un signal électrique proportionnel
d► En stockant la lumière en mémoire</p> <p>12► Quel est la principale fonction des filtres devant chaque photosite en photographie numérique ?</p> <p>a► Empêcher la lumière d'atteindre les photosites
b► Accentuer la vision nocturne
c► Enregistrer uniquement l'intensité lumineuse
d► Capturer les couleurs en utilisant la matrice de Bayer</p> <p>13► Quelle est la composition d'une matrice de Bayer ?</p> <p>a► Des carrés de 3 photosites (un de chaque couleur primaire)
b► Des triangles de 2 photosites (un bleu et un vert)
c► Des carrés de 4 photosites (deux verts, un bleu, un rouge)
d► Des cercles de 5 photosites (une combinaison aléatoire des couleurs)</p> |
|---|--|

C – Régler un appareil photo

Les élèves qui n'ont pas peur de suivre une vidéo en anglais sont invité(e)s à voir les vidéos [Do *you* understand ISO ?](#) de Minute Physics et [Intuitively Understand Exposure in 12 Minutes \(With Real Examples\)](#) de Never fStop⁽¹⁾ sur YouTube.

Là où l'anatomie d'un appareil photo est semblable en tout point à celle de l'œil, leurs fonctionnements sont drastiquement différents. Les appareils photo ont tous trois réglages en commun : la *vitesse d'obturation* (ou le *temps d'exposition*), l'*ouverture du diaphragme* et la *sensibilité ISO* (par commodité, on les raccourcit souvent en « obturation », « ouverture » et « ISO »). Les smartphones règlent automatiquement ces paramètres la majeure partie du temps, mais avec un « vrai » appareil photo, il faut ajuster ces trois paramètres manuellement⁽²⁾.

i – Une analogie pour comprendre le fonctionnement

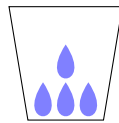
Pour comprendre comment ces paramètres se manifestent au sein de l'appareil, étudions une analogie.

L'on peut voir un appareil photo comme un seau laissé sous la pluie : ici, la pluie représente la lumière, et le seau représente un photosite. Dans cette analogie,

- ▶ L'ouverture : c'est le diamètre du seau. Plus l'ouverture est grande, plus le seau collecte de gouttes de pluie.
- ▶ L'obturation (ou le temps d'exposition) : c'est le temps que l'on laisse le seau sous la pluie. Plus on laisse le seau longtemps, plus il sera rempli.
- ▶ L'ISO : c'est (l'inverse de) la hauteur du seau. Plus le seau est haut, plus il faudra de temps pour le remplir. (Attention à l'analogie : plus l'ISO est haute, plus le seau est petit, et inversement. Autrement dit, plus l'ISO est haute, plus le seau sera rempli rapidement.)



Grande ouverture, faible ISO



Ouverture et ISO moyennes



Petite ouverture, grande ISO

Remarque 5

Toute analogie a ses limites, certains aspects de l'illustration ne sont pas une retranscription fidèle de la réalité. Ici, l'analogie proposée est loin de la réalité : en effet, l'analogie suggère ici que les trois réglages modifient les photosites en eux-même. Or ceci est impossible, les photosites sont immuables. Ces trois paramètres influencent en fait ce qui se passe avant et après le capteur CCD dans l'appareil photo, plus précisément le système de lentilles et le traitement du signal. Voir la suite pour plus de détails !

Notons également que, dans cette analogie, l'intensité de la pluie correspond à l'intensité de la lumière. Une fois la photo prise, autrement dit une fois qu'on est satisfait du remplissage du seau, on regarde à quel point chacun est rempli : s'il est vide, on envoie un signal électrique nul, s'il est plein on envoie un signal électrique d'intensité maximale, et s'il est rempli à moitié, on envoie un signal électrique d'intensité la moitié du maximum.

Une photo est dite *surexposée* sur les photosites collectent trop de lumière. Dans l'analogie, cela revient à laisser les seaux se remplir à raz-bord. Les détails se perdent, et il ne reste qu'une tâche lumineuse. Inversement, une photo est *sous-exposée* lorsque les photosites ne collectent pas assez de lumière. Dans l'analogie, ce revient à une configuration où les seaux ne remplissent que très peu. Les détails sont difficilement distingués, et l'image est sombre.

(1). L'analogie du seau d'eau est inspirée de cette vidéo.

(2). Les appareils photo modernes disposent d'assistants de réglage semi-automatiques, permettant aux photographes de régler un paramètre en particulier en gardant un équilibre avec les deux autres.



Photo surexposée



Bonne exposition d'une photo



Photo sous-exposée

Le but du jeu est donc de régler les trois paramètres pour que les seaux soient ni trop ni trop peu remplis. On comprend assez intuitivement avec cette analogie que :

- Si la pluie est battante, il faut faire en sorte que les seaux ne collectent pas trop de gouttes d'eau. Pour cela, on peut diminuer leur diamètre pour que les gouttes aient moins de chance de tomber dedans, on peut les laisser sous la pluie moins longtemps, ou on peut choisir un seau plus haut pour qu'il ait plus de place pour se remplir. Cependant, il faut faire attention à ne pas agir à outrance sur ces trois paramètres, sinon le seau finira presque vide malgré la pluie intense.
- *A contrario*, si la pluie est très faible, on peut choisir un seau de grand diamètre pour que les gouttes aient plus de chances de tomber dedans, on peut laisser le seau plus longtemps sous la pluie, et on peut réduire sa hauteur pour qu'il soit rempli plus rapidement. Là encore, il ne faut pas abuser de ces trois paramètres conjointement, au risque que le seau se retrouve trop rempli malgré la faible pluie.

ii – L'effet concret des paramètres

À l'issue de la lecture de cette section, les élèves sont invités à expérimenter les réglages d'un appareil photo sur [ce parcours Algoréa](#).

L'obturation

L'*obturateur* est un mécanisme qui joue le rôle d'une porte pour la lumière : lorsqu'il est fermé, la lumière ne peut pas atteindre le capteur CCD. Ainsi, plus l'obturateur reste ouvert longtemps (autrement dit plus la vitesse d'obturation est faible, autrement dit plus le temps d'exposition est long), plus le capteur recevra de lumière, et inversement.



Obturateur ouvert (on distingue le capteur CCD)



Obturateur fermé

Cependant, le temps d'exposition ne sert pas qu'à contrôler la quantité de lumière perçue par le capteur, il a également un effet très spécifique : **le temps d'exposition permet de figer les mouvements lorsqu'il est court, et introduit un flou de mouvement lorsqu'il est long.**



Vitesse d'obturation rapide : peu de flou de mouvement



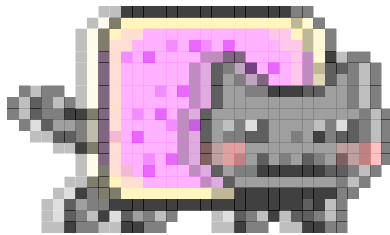
Vitesse d'obturation lente : mouvement très flou

Remarque 6 (Le pourquoi du comment)

Pour mieux comprendre l'origine de ce phénomène, on peut imaginer (et faire !) une petite expérience : demandez à un ami de faire glisser une feuille sur une table. Le but du jeu pour vous est, pendant que la feuille bouge, d'apposer un point sur la feuille à l'aide d'un feutre. L'on peut alors comprendre qu'il vous faudra être vif : si vous laissez la pointe du feutre trop longtemps en position basse, le mouvement de la feuille vous fera tracer un trait. Et plus la feuille bouge rapidement, plus ils vous faudra retirer le feutre rapidement.

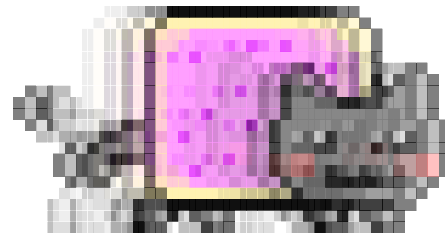
En photographie, le principe est le même : si l'objet de votre photo bouge, il va se déplacer dans la grille de pixels, créant ce flou de mouvement. Ainsi, avec un temps d'exposition faible, la cible a moins le temps de se déplacer, et donc parcourra une distance plus faible (puisque que, comme tout le monde le sait, $d = v \times t$). Inversement, plus le temps d'exposition est long, plus la cible parcourt une grande distance, et envoie sa lumière sur une plus grande plage de pixels voisins.

Déplacement
→



Temps d'exposition court : l'objet se déplace peu et l'image est plus nette

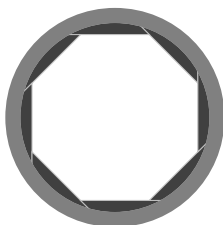
Déplacement
→



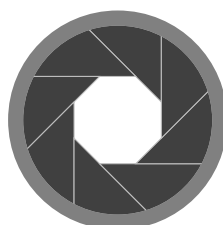
Temps d'exposition long : l'image se déplace beaucoup, et un flou se crée dans la direction du mouvement

L'ouverture

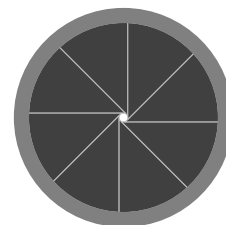
L'ouverture se règle grâce au *diaphragme* placé devant l'ensemble de lentilles. C'est le premier endroit, dans la chaîne des événements, où la lumière est filtrée. Le diaphragme est analogue à la pupille de l'œil, qui se contracte ou s'agrandit pour laisser passer plus ou moins de lumière, selon s'il fait sombre ou lumineux.



Ouverture la plus grande possible

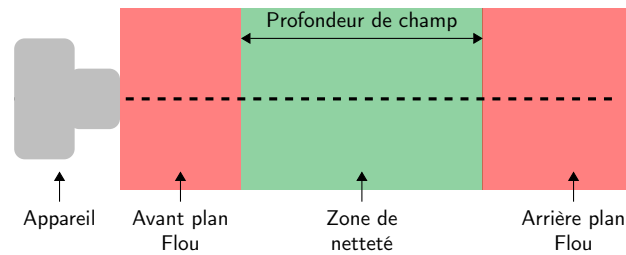


Ouverture intermédiaire



Ouverture la plus petite possible

Cependant, l'ouverture ne sert pas qu'à contrôler la quantité de lumière perçue par le capteur, elle a également un effet très spécifique : **l'ouverture contrôle la profondeur de champ**. La *profondeur de champ* est l'ensemble des distances auxquelles les objets sont nets à l'image.



Par exemple, si les objets doivent être précisément à 1 m de l'objectif pour ne pas être flous, alors la profondeur de champ est faible. Inversement, si tous les objets entre 50 cm et 5 m sont nets, alors il y a une grande profondeur de champ. La distance de la zone de netteté est réglée lorsque l'on fait la *mise au point* (l'équivalent de l'*accommodation* effectuée inconsciemment par nos yeux).

L'ouverture est exprimée comme « f sur un nombre », où f est la distance focale de base de l'objectif. Par exemple, avec une focale $f = 50$ mm et une ouverture $f/5$, le diaphragme se resserrera de sorte que le diamètre ouvert soit de $50/5 = 10$ mm.



Faible profondeur de champ ($f/5$)
[1/320 - ISO 800]



Grande profondeur de champ ($f/32$)
[1/6 - ISO 800]



Faible profondeur de champ ($f/5$)
[1800 - ISO 1000]

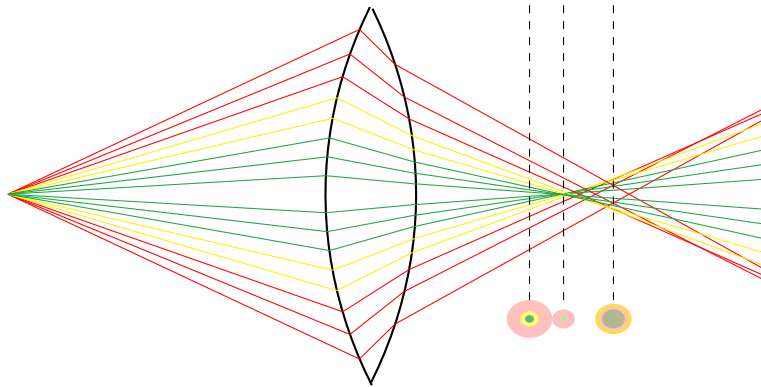


Grande profondeur de champ ($f/32$)
[1/20 - ISO 1000]

Remarque 7 (Le pourquoi du comment)

Les lentilles ne focalisent pas parfaitement la lumière en un point : comme le montre l'illustration ci-dessous réalisée en simulant les lois de Snell-Descartes (qui dictent le trajet des rayons lumineux), tous les rayons issus d'un même point de départ (la source) ne sont pas déviés jusqu'au même endroit. Les rayons verts dessinés en vert, qui sont peu inclinés par rapport à l'axe de la lentille (appelé *axe optique*), sont bien déviés par la lentille, autrement dit les rayons se croisent au même point de l'autre côté (en réalité, si l'on zoomait, on verrait qu'ils sont très légèrement décalés) – on parle de *stigmatisme approché* –. Les rayons jaunes restent encore acceptables, mais les rouges sont complètement dispersés : on dit que la lentille est *astigmatique* pour ces rayons (d'où le nom de maladie de la vue, l'astigmatisme, qui fait que les personnes qui en souffrent voient des traits

ou des taches au lieu d'objets bien nets).



Simulation des trajets des rayons lumineux dans une lentille (source à gauche)⁽¹⁾

Fort heureusement, les photosites d'un capteur ne sont pas des points, mais des petits carrés ; ainsi, il existe une certaine marge de manœuvre : tant que la tache ne dépasse pas la taille d'un photosite, cela ne cause aucun flou. *A contrario*, comme on l'a vu plus haut avec le temps d'exposition, si la tache déborde sur les photosites voisins, alors l'image sera floue à cet endroit car les différentes informations se mélangent.

L'ISO

L'ISO est le troisième levier permettant de contrôler la luminosité d'une image.



ISO trop faible : photo sous-exposée (ISO 250)
[1/3200 - $f/5.6$]



ISO bien réglé (ISO 800)
[1/3200 - $f/5.6$]



ISO trop fort : photo surexposée (ISO 4000)
[1/3200 - $f/5.6$]

L'ISO est spécifique à la photographie électronique, car contrairement au temps d'exposition, il n'agit pas sur la lumière mais sur le signal électrique. **L'ISO correspond à une multiplication des valeurs mesurées par le capteur.** En effet, le capteur traduit l'intensité lumineuse en tension électrique, et son travail ne va pas plus loin. Par exemple, 2 mV pour tel photosite, 10 mV pour un autre... Mais cette information est presque inexploitable seule : en effet, 10 mV correspondent-ils à une intensité faible ? moyenne ? forte ? Le capteur n'est celui qui en décide : le seul juge, c'est le système de traitement du signal qui détermine la tension qu'il considère comme étant l'intensité maximale.

Or il se peut que le photographe et le système de traitement ne soient pas d'accord : peut-être que pour le photographe, l'intensité maximale correspond à 100 mV, alors que le système de traitement va jusqu'à 500 mV. L'ISO entre alors en scène : en multipliant par 5 l'intensité du signal, il faut correspondre les points de vue du photographe et du système !

(1). Les trois disques à droite représentent l'image si on plaçait le capteur au niveau des pointillés : trop tôt, et l'image est floue, comme si on avait zoomé dedans avec une mauvaise qualité ; la deuxième position est l'endroit où les rayons verts se croisent, qui est l'endroit permettant la meilleure netteté. Malheureusement, les rayons rouges ajoutent du flou, d'où la nécessité de les bloquer à l'adire du diaphragme. Enfin, en plaçant le capteur trop loin, les rayons s'échangent et se mélangent, l'image devient floue et complètement indiscernable.

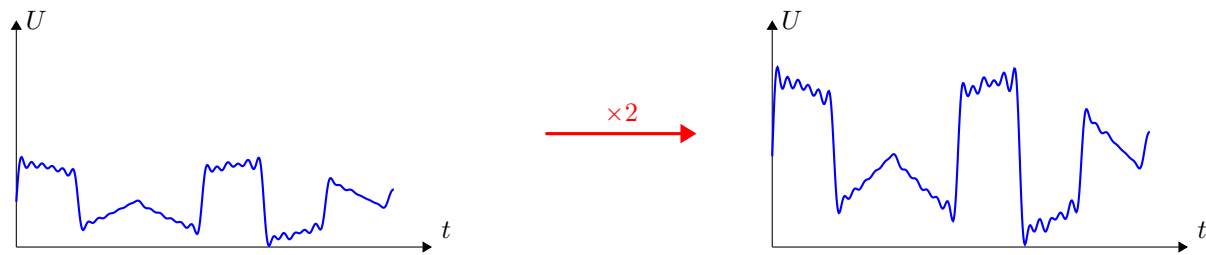
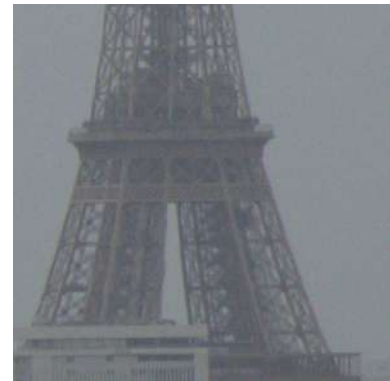


FIGURE : Amplification du signal

Remarquez cependant sur l'illustration ci-dessus comment le signal « tremble » : on parle de *bruit* (électronique). Cela est dû à de nombreux facteurs (température, aléatoire, interférences électriques...); aléatoirement, le signal envoyé par les photosites est légèrement faussé, et envoie un signal un peu trop fort ou un peu trop faible. La conséquence sur l'image est de petites variations de couleurs : un pixel peut avoir la couleur (247, 133, 75) au lieu de la couleur (245, 135, 75) qu'il était censé avoir. Heureusement, cette différence est imperceptible à l'œil nu. Le bruit commence à poser problème lorsqu'il introduit de fortes variations dans le signal, à tel point que la différence de couleur devient perceptible : on dit alors que l'image a du *grain*.



ISO convenable ne montrant pas le bruit



ISO trop fort faisant apparaître le bruit (grain)

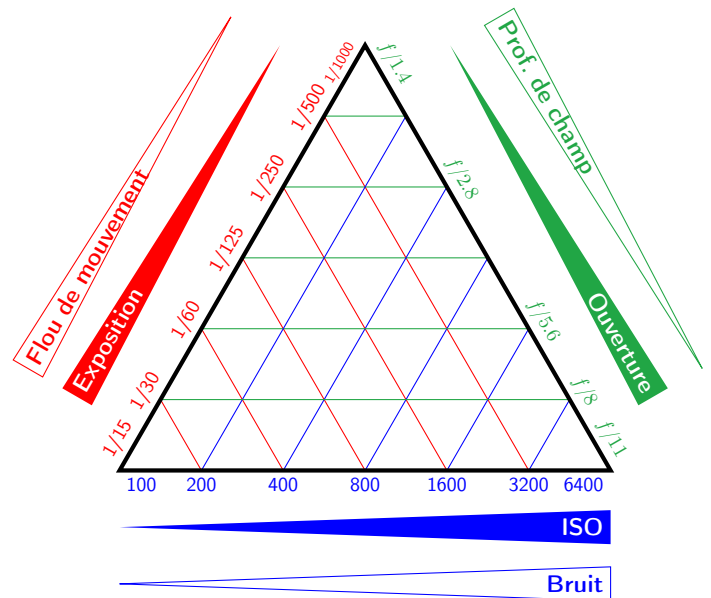
Lorsqu'on amplifie le signal, par exemple en le multipliant par deux, le bruit est lui aussi multiplié par deux, comme illustré sur les courbes ci-dessus.

La règle d'or du réglage

Pour régler un appareil photo, on peut retenir la règle suivante : **on effectue les réglages dans l'ordre du trajet de la lumière**. Comme on l'a vu plus haut, l'ouverture est ce que la lumière rencontre en premier ; puis elle traverse les lentilles de l'objectif avant d'être bloquée ou non par l'obturateur. Enfin, elle traverse le capteur, le l'ISO amplifie le signal qui en sort. Ainsi, on règle d'abord l'ouverture, puis le temps d'exposition, et enfin l'ISO. On pourrait même ajouter un quatrième réglage : l'éclairage de la scène, qui faut adapter avant même de régler l'appareil photo, puisque c'est de là que vient la lumière ; enfin, si cela est possible, bien sûr !

Pour décider de l'ouverture à choisir, on a vu que la profondeur de champ était un critère à prendre en compte. Ensuite, pour choisir le temps d'exposition, on a vu que la vitesse du mouvement de l'objet de la photo était une donnée critique. Paradoxalement à ce que le paragraphe sur l'ISO laisse entendre, il faut cependant régler l'ISO au plus haut niveau acceptable étant donnés les deux précédents réglages. En effet, en se plaçant juste en-dessous de la limite de la surexposition ou du grain, on obtient l'image la plus lumineuse possible, et si plus tard cela ne nous convient pas, on peut toujours l'assombrir dans un logiciel de retouches photo. Cet assombrissement aura pour conséquence bénéfique de réduire le bruit. *A contrario*, si l'on devrait éclaircir l'image dans un logiciel de retouche, cela amplifierait le bruit plus que si on avait réglé l'ISO au préalable, et ce car le logiciel traite les données sous un format fini ; voir la suite du cours pour plus d'explications.

On peut préciser les règles de réglage avec la figure ci-contre. Bon à savoir : les appareils modernes disposent de réglages semi-automatiques, permettant de ne régler que l'ouverture (A ou Av), ou que le temps d'exposition (S ou Tv). Il existe également un mode automatique, mais l'intérêt d'un appareil photo est tout de même aussi de faire certains choix artistiques !



D – De la capture au fichier

Information 8

Format RAW

Les format d'image .raw correspond aux données brutes et non transformées du capteur photographique. Il contient l'intégralité des données de l'image, et est donc plus lourd que les autres formats qui raffinent, voire suppriment une partie des données pour rendre le fichier plus léger. En contrepartie de son poids, la richesse de ses informations permet de traiter plus librement l'image.

Définition 9

Données EXIF

Les (méta)données EXIF (*Exchangeable Image File Format*) sont des métadonnées attachées à une photo, et renseignant les caractéristiques de celle-ci : quel appareil a été utilisé, avec quels réglages, à quelle date, à quel endroit, le nom du photographe, les droits d'utilisations de l'image, ...

Information 10

Les (méta)données EXIF sont au moins partiellement visibles dans les propriétés du fichier image, dans le volet « Détails » sur Windows. Des outils en ligne permettent également une exploration plus poussée de ces données.

Sur téléphone, ces données sont partiellement visibles dans les propriétés de l'image, ou en glissant l'image vers le haut dans certains visionneurs.

Information 11

Les formats d'image .jpg, raw ou .tiff peuvent contenir des données EXIF. Les formats .jpg2 ou .png par exemple ne sont pas compatibles.

Activité 12 (Lecture de données EXIF)

Les photos pour cette activités sont disponibles à l'adresse : <https://esuong-maths.fr/snt-activites.html>.

Pour visualiser les données EXIF, vous pouvez effectuer un clic droit sur le fichier, chercher « Propriétés » puis consulter le volet « Détails ». Si vous disposez d'une connexion internet, le site www.exifdata.com permet une visualisation plus avancée.

(Remarque : ne chargez jamais un fichier sur un site en lequel vous n'avez pas pleinement confiance !)

- 1► Télécharger la photo artifices.jpg.
 - À quelle date cette photo a-t-elle été prise ?
 - À quelle heure cette photo a-t-elle été prise ?
 - Quelle est sa résolution ?
 - Avec quel modèle de téléphone a-t-elle été prise ?
 - Quel a été le temps d'exposition ?
- 2► Télécharger les trois photos du lot n°2.

À quelle distance le photographe se trouve-t-il du sujet sur chaque photo ? Commenter la vraisemblance de ce résultat.
- 3► Télécharger photo.jpg.

Si vous savez interpréter des coordonnées GPS, lisez celle-ci dans les détails de l'image comme précédemment, sinon utilisez l'outil disponible à l'adresse : <https://esuong-maths.fr/exif-viewer.html>.
À l'aide de ces informations, trouvez le lieu où cette photo a été prise.

3 – Traitement d'images

Définition 13

Traitement d'image

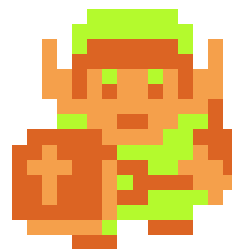
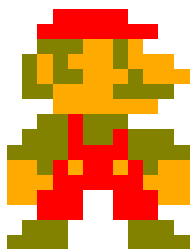
Traiter une image, c'est la modifier en modifiant les pixels qu'elle contient. Il peut s'agir de retouches sur sa luminosité, sur la balance de ses couleurs, appliquer des filtres... à l'aide d'un logiciel de retouche photos, ou réorganiser son information pour la compresser (avec ou sans perte).

Exemple 14

- Pour prendre le négatif d'une image, on modifie le code RVB de chacun de ses pixels selon la formule suivante : sur le code est (r, v, b) , alors on le remplace par $(255 - r, 255 - v, 255 - b)$.
- Pour extraire le canal rouge d'une image, on remplace chacun ses codes (r, v, b) de ses pixels par $(r, 0, 0)$. De même, pour obtenir le canal vert, on écrit $(0, v, 0)$, et $(0, 0, b)$ pour le bleu.
- Pour transformer l'image en noir et blanc (autrement dit en niveaux de gris), on remplace chaque code RVB par la moyenne de ses trois valeurs, autrement dit par $\left(\frac{r+v+b}{3}, \frac{r+v+b}{3}, \frac{r+v+b}{3}\right)$.

Activité 15 (Pixel art)

Les codes Python dessinant les pixel arts ci-dessous ont été corrompus ! L'un a été remplacé par son négatif, et pour l'autre les canaux rouge, vert et bleu ont été permutés ! Copiez le code en [annexe](#) dans un projet Python (par exemple un Notebook sur [Basthon](#)), et corrigez les variables A, B, C et D, correspondant aux codes RVB des différentes couleurs, pour que les images soient telles que ci-dessous.



Activité 16

Les élèves curieux/ses d'expérimenter la manipulation d'images par pixels individuels pourront suivre [cette activité](#).

4 – Les enjeux sociétaux liés aux images

Activité 17

À l'aide des documents suivants, et de vos connaissances, répondre à la question suivante : quels sont les bénéfices et les risques engendrés par les métadonnées EXIF ? La réponse prendra la forme d'un texte d'au moins 100 mots, argumenté, et illustré de situations-exemples.

DOCUMENT 1 : À quoi servent les métadonnées EXIF ?

Certains logiciels se servent de métadonnées EXIF pour classer les photos, et faciliter la recherche d'un fichier,

en composant des filtres avec de nombreux critères. D'autres peuvent indiquer la ville de la prise de vue sur une carte. Un photographe pourra utiliser les paramètres de configuration de la prise de vue afin d'affiner les réglages de son appareil en fonction de ses performances réelles. Plus de 80% des photographes professionnels renseignent les métadonnées pour protéger leurs photos. Les métadonnées EXIF, lorsqu'elles sont renseignées en quantité suffisante, peuvent agir comme une « empreinte digitale » de la photo. Cependant, 85% des images publiées sur internet n'ont pas de métadonnées. Les droits liés à l'usage de la photographie, données essentielles pour les créateurs de ces ressources numériques, ne peuvent alors plus être consultés. La première source de cette disparition est le retraitement systématique des images qui sont intégrées aux sites web dans un format qui permet d'optimiser la bande passante des internautes. Pour contourner ce problème, les professionnels peuvent utiliser le « tatouage numérique » des photos. Une autre solution consiste à enregistrer les images en sécurisant les métadonnées auprès d'un organisme payant, chargé de rechercher les copies sur le web à l'aide d'un algorithme. Il existe des sites dédiés à la recherche inversée d'images. Ceci permet de trouver les métadonnées originales d'une image.

DOCUMENT 2 : Données et vie privée

Les données EXIF sont aussi des données personnelles qui sont accessibles à tous. Rendre disponibles les photographies géolocalisées de vos vacances permet à tous les internautes de potentiellement associer votre nom à un itinéraire, des dates, une adresse personnelle, ... Il paraît donc important de supprimer tout ou partie des métadonnées des photos avant de les diffuser sur le net !

Activité 18

À l'aide de la liste de documents suivante, répondre aux questions plus bas.

- ▶ Wikipédia : [Droit à l'image](#)
- ▶ Portail du gouvernement : [Droit à l'image des personnes et des biens](#) | [Le droit à l'image](#)
- ▶ CNIL : [Demander le retrait de votre image en ligne](#) | [Le droit à l'image s'applique-t-il sur internet ?](#)

- 1▶ Expliquer simplement en quoi consiste le droit à l'image.
- 2▶ Vrai ou faux : en vertu du droit à l'image, il est interdit de photographier une personne sans son consentement.
- 3▶ Pour qu'une personne soit identifiable sur une photo, faut-il nécessairement que son visage apparaisse ?
- 4▶ À quelles sanctions s'expose-t-on en cas de non respect du droit à l'image ?
- 5▶ Voici différentes situations. Pour chacune, dire si oui ou non le droit à l'image a été respecté, ou si cela dépend d'autres facteurs.
 - a▶ Alice se rend au marché de Noël de Strasbourg et photographie l'allée bondée de monde. Elle publie ensuite cette photo sur son blog.
 - b▶ Lors d'un reportage télévisé, un journaliste filme la personne interrogée, et on distingue clairement Bob qui passe en vélo à l'arrière plan.
 - c▶ Carole photographie les animaux dans un parc, et prend la photo d'un papillon posé sur la tête d'un passant. Elle publie ensuite cette photo sur ses réseaux sociaux.
 - d▶ Un stalker espionne David et le photographie à son insu dans la rue. Il imprime cette photo et l'affiche dans sa chambre.
 - e▶ Eve est prise en photo par son entreprise pendant qu'elle travaille, et l'entreprise met cette image sur la page promotionnelle de son site.
 - f▶ Freddie prend une photo d'une personnalité politique durant un meeting, et vend des T-shirts avec cette image imprimée dessus.
 - g▶ Gregory se filme dans les couloirs à l'intercours alors qu'un professeur passe derrière lui, et publie cette story sur ses réseaux sociaux.

- h► Henry est pris en photo par son professeur lors du tournoi de basket, et cette photo est affichée sur le site du lycée.
- i► Illona prend en photo sa voisine qui dort sur sa table, et envoie cette photo sur son petit groupe d'amies.
- j► John participe au film réalisé par sa classe, qui sera projeté devant tout le lycée à la fin de l'année.
- k► Témoin d'une altercation entre deux élèves, Katelyn filme la scène pendant que d'autres élèves essaient de séparer les deux élèves.
- l► Léandre perturbe un cours en frappant à plusieurs reprises à la porte d'une salle. Le professeur lui demande de donner sa carte de lycée pour l'identifier, Léandre refuse. Le professeur prend alors Léandre en photo pour l'identifier plus tard auprès de la direction.

Annexes

(Illustrations pour les activités 1 et 3.)

DOCUMENT 1 : Les figures de la photographie



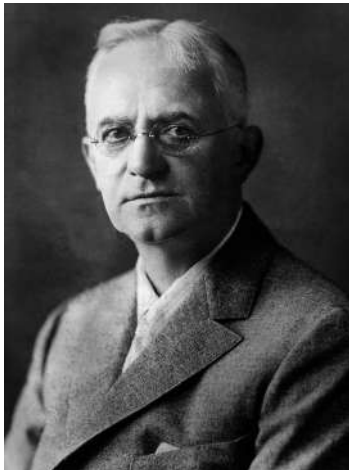
Louis Daguerre



William Henry Fox Talbot



Louis Ducos du Hauron



George Eastman



Steven Sasson



Nicéphore Niépce



Louis Lumière

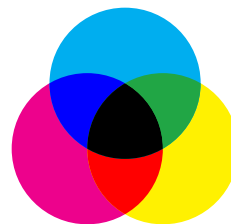
DOCUMENT 2 : Les différentes inventions photographiques



Le daguerréotype



Un calotype



Principe trichromique



Premier appareil portable



Héliographie

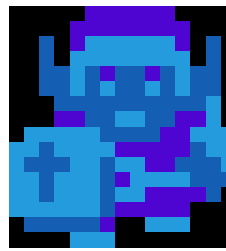


Autochromie

Annexe pour l'Activité 15 – copiez le premier bloc de code ci-dessous, puis le second en fonction de l'image que vous souhaitez traiter.

```
from ipythonblocks import BlockGrid

def dessiner(matrice):
    nb_lignes = len(matrice)
    nb_colonnes = len(matrice[0])
    grille = BlockGrid(nb_colonnes, nb_lignes, block_size=30)
    for ligne in range(nb_lignes):
        for colonne in range(nb_colonnes):
            couleur = matrice[ligne][colonne]
            grille[ligne, colonne] = couleur
    return grille
```



```
A = (255, 255, 255)
B = (0, 255, 0)
C = (0, 255, 170)
D = (0, 130, 130)
```

```
matrice = [
    [A, A, A, B, B, B, B, B, A, A, A, A],
    [A, A, B, B, B, B, B, B, B, A, A, A],
    [A, A, D, D, D, C, C, D, C, A, A, A],
    [A, D, C, D, C, C, C, D, C, C, C, A],
    [A, D, C, D, D, C, C, C, D, C, C, C],
    [A, D, D, C, C, C, C, D, D, D, D, A],
    [A, A, A, C, C, C, C, C, C, C, A, A],
    [A, A, D, D, B, D, D, D, A, A, A, A],
    [A, D, D, D, B, D, D, B, D, D, D, A],
    [D, D, D, D, B, B, B, B, D, D, D, D],
    [C, C, D, B, C, B, B, C, B, D, C, C],
    [C, C, C, B, B, B, B, B, B, C, C, C],
    [C, C, B, B, B, A, A, B, B, B, C, C],
    [A, A, B, B, B, A, A, B, B, B, A, A],
    [A, D, D, D, A, A, A, A, D, D, D, A],
    [D, D, D, D, A, A, A, A, D, D, D, D]
]
```

```
dessiner(matrice)
```

```
A = (0, 0, 0)
B = (35, 155, 220)
C = (20, 95, 180)
D = (78, 5, 210)
```

```
matrice = [
    [A, A, A, A, A, D, D, D, D, D, A, A, A, A],
    [A, A, A, A, D, D, D, D, D, D, D, A, A, A],
    [A, A, C, A, D, B, B, B, B, B, D, A, C, A],
    [A, A, C, A, B, B, B, B, B, B, B, A, C, A],
    [A, A, C, C, B, C, D, C, C, D, C, B, C, C],
    [A, A, C, C, B, C, B, C, C, B, C, B, C, C],
    [A, A, A, C, C, C, C, C, C, C, C, C, B, A],
    [A, A, A, D, D, C, C, B, B, C, C, D, D, B, A],
    [A, B, B, B, B, B, C, C, C, C, D, D, B, B, B],
    [B, B, C, B, B, B, B, D, D, D, D, D, C, B, B],
    [B, C, C, C, B, B, C, B, B, D, D, C, C, C, B],
    [B, B, C, B, B, B, C, D, B, B, B, B, C, C, C],
    [B, B, C, B, B, B, C, B, B, D, D, D, D, C, A],
    [B, B, B, B, B, B, C, D, D, D, D, D, A, A, A],
    [A, C, C, C, C, C, D, A, A, B, B, B, A, A, A],
    [A, A, A, A, B, B, B, A, A, A, A, A, A, A, A]
]
```

```
dessiner(matrice)
```

Thème d'étude IV

Localisation et cartographie

Programme

Localisation, cartographie et mobilité

Introduction

La cartographie est essentielle pour beaucoup d'activités : agriculture, urbanisme, transports, loisirs, etc. Elle a été révolutionnée par l'arrivée des cartes numériques accessibles depuis les ordinateurs, tablettes et téléphones, bien plus souples à l'usage que les cartes papier. Les cartes numériques rassemblent toutes les échelles et permettent de montrer différents aspects de la région visualisée sur une seule carte. Les algorithmes de recherche permettent de retrouver sur la carte les endroits en donnant simplement leur nom, et de calculer des itinéraires entre points selon des modes de transports variés.

Repères historiques

Les cartes ont été systématiquement numérisées à la fin du XXe siècle. Le principal instrument de localisation, GPS (Global Positioning System), a été conçu par l'armée américaine dans les années soixante. Le premier satellite GPS fut lancé en 1978. Il y en a actuellement une trentaine, de sorte qu'à tout moment quatre à six satellites au moins sont visibles depuis tout point de la Terre. Couplé aux cartes numériques, le système GPS permet de se situer. Il n'est pas toujours efficace en ville, et peut être complété par d'autres moyens de localisation comme la détection de bornes Wi-Fi proches. D'autres systèmes plus précis, dont Galileo, sont en cours de déploiement.

Les données et l'information

Les informations des cartes numériques proviennent de nombreuses sources : services géographiques des États, photos prises par des satellites, avions ou voitures, données fournies par les utilisateurs, etc. Ces informations sont de natures diverses : topographiques, géologiques, photographiques, liées aux transports, à l'activité industrielle ou touristique, etc. Des projets collaboratifs comme OpenStreetMap permettent à chaque utilisateur d'ajouter des informations à une carte en libre accès, qui deviennent alors visibles par tous les utilisateurs. Un satellite GPS contient une horloge atomique mesurant le temps à une très grande précision et envoyant régulièrement des messages contenant cette heure. Chaque message se propageant à la vitesse de la lumière, le récepteur peut calculer sa distance au satellite. On peut en déduire sa position en suivant plusieurs satellites, ce que fait automatiquement le récepteur GPS.

Les algorithmes et les programmes

Les algorithmes cartographiques concernent principalement l'affichage sélectif d'informations variées et le calcul d'itinéraires. L'affichage est paramétré par les informations à montrer, que l'on peut choisir par simples clics. Une difficulté est liée au mélange d'informations de types différents lors des changements d'échelle : les graphismes peuvent être très différents et beaucoup d'informations doivent être supprimées pour les grandes échelles, mais une route doit être représentée avec à peu près la même largeur, quelle que soit l'échelle.

Les récepteurs GPS fournissent la localisation sous une forme normalisée facilement décodable, par exemple selon le protocole NMEA 0183 (National Marine Electronics Association), ou directement dans les métadonnées EXIF d'une photo. La localisation et les cartes se couplent dans le suivi permanent de la position sur la carte ou sur un itinéraire précalculé.

Les machines

Les machines utilisées pour la cartographie sont surtout les ordinateurs, tablettes et téléphones classiques équipés d'une application ad hoc. Les récepteurs GPS spécialisés restent importants pour la navigation maritime ou aérienne, mais ceux pour la randonnée pédestre sont en voie de disparition, supplantés par les téléphones.

L'heure fournie par le GPS sert aussi de base pour la synchronisation précise des horloges internes des ordinateurs connectés à internet, ce qui est très important pour tous les échanges d'informations.

Impacts sur les pratiques humaines

Les cartes numériques, accessibles depuis un téléphone, remplacent progressivement les cartes sur papier. Leurs interfaces permettent d'accéder commodément à de nombreux types d'information. Couplé aux algorithmes de calculs d'itinéraires, le GPS est utilisé systématiquement pour les transports, l'agriculture, la randonnée, la navigation à voile, etc. Le maintien à jour des cartes numériques est un problème difficile qui demande beaucoup de ressources au plan mondial. Les erreurs dans les cartes, inévitables à cause de l'énorme quantité d'informations à collecter, peuvent avoir des conséquences dramatiques. Par ailleurs, de nombreuses applications ont accès à la localisation dans un téléphone, ce qui leur permet d'envoyer des publicités non désirées, de suivre vos itinéraires, ou de localiser une personne. Enfin, le GPS n'est pas toujours sûr, car facile à brouiller à l'aide d'appareils simples.

Contenus	Capacités attendues
GPS, Galileo	Décrire le principe de fonctionnement de la géolocalisation.
Cartes numériques	Identifier les différentes couches d'information de GeoPortail pour extraire différents types de données. Contribuer à OpenStreetMap de façon collaborative.
Protocole NMEA 0183	Décoder une trame NMEA pour trouver des coordonnées géographiques.
Calculs d'itinéraires	Utiliser un logiciel pour calculer un itinéraire. Représenter un calcul d'itinéraire comme un problème sur un graphe.
Confidentialité	Régler les paramètres de confidentialité d'un téléphone pour partager ou non sa position.

Exemples d'activités

- ▶ Expérimenter la sélection d'informations à afficher et l'impact sur le changement d'échelle de cartes (par exemple sur GeoPortail), ainsi que les ajouts d'informations par les utilisateurs dans OpenStreetMap.
- ▶ Mettre en évidence les problèmes liés à un changement d'échelle dans la représentation par exemple des routes ou de leur nom sur une carte numérique pour illustrer l'aspect discret du zoom.
- ▶ Calculer un itinéraire routier entre deux points à partir d'une carte numérique.
- ▶ Connecter un récepteur GPS sur un ordinateur afin de récupérer la trame NMEA, en extraire la localisation.
- ▶ Extraire la géolocalisation des métadonnées d'une photo.
- ▶ Situer sur une carte numérique la position récupérée.
- ▶ Consulter et gérer son historique de géolocalisation.

7 Comment (se) repérer ?

Table des matières

1	Coordonnées sphériques et GPS	99
2	Les premiers outils de localisation	103
3	La localisation par satellite	105
A	La trilatération	105
B	L'histoire du GPS	106
C	Limitations techniques du GPS	107
4	Les centrales inertielles	108
5	Représentation des données géographiques	109
A	Les cartes géographiques physiques : histoire et limites	109
i	Des cartes pour la navigation	109
ii	Les problèmes de projection et d'échelle	110
iii	Les mathématiques au service de la mesure topographique	112
iv	Problèmes de conservation, reproduction et péremption	113
v	Prendre de la hauteur	113
B	Les cartes numériques	114
i	Des points, des lignes et des polygones – GeoJSON	114
ii	Le principe des couches	116
iii	La disparition du problème de l'échelle	117
C	Qui fait les cartes ?	118
i	Les cartes physiques	118
ii	Les cartes numériques	118
iii	Et si c'était vous ?	120
6	Annexes	121

1 – Coordonnées sphériques et GPS

Définition 1

Parallèles et méridiens

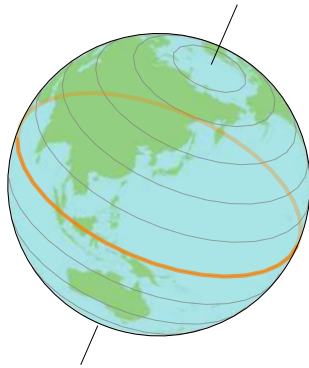
On assimile la Terre à une sphère. Celle-ci tourne autour de l'axe *Nord-Sud*.

- ▶ Les *pôles* de la Terre sont les points de sa surface qui sont traversés par l'axe *Nord-Sud*. Conventionnellement, on appelle *pôle Nord* l'un des deux pôles et *pôle Sud* l'autre pôle.
- ▶ On découpe la Terre le long de l'axe Nord-Sud. Cela en divise la surface en des cercles, appelés *parallèles*.
- ▶ On découpe également la Terre selon divers plans contenant l'axe Nord-Sud. Cela en divise la surface en des cercles, se croisant aux pôles Nord et Sud. Chaque demi-cercle, reliant les deux pôles, est appelé *méridien*.

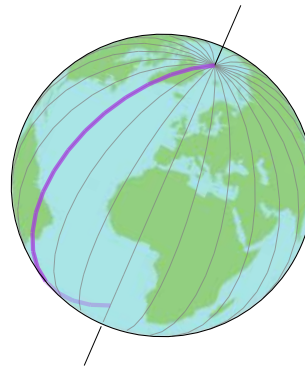
Définition 2

Équateur, méridien de Greenwich

- ▶ Le plus grand parallèle est appelé *équateur*.
- ▶ Le *méridien de Greenwich* est l'unique méridien qui passe par l'Observatoire royal dans le quartier de Greenwich à Londres.



— : équateur



— : méridien de Greenwich

Est, Ouest

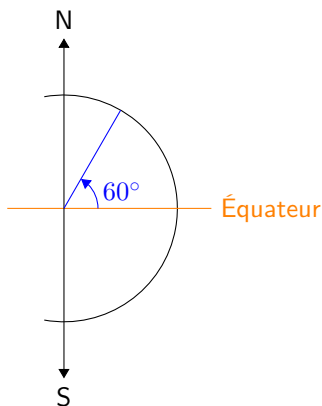
Définition 3

- ▶ On appelle *Est* la direction vers laquelle on observe le Soleil se lever depuis la surface de la Terre.
- ▶ On appelle *Ouest* la direction vers laquelle on observe le Soleil se coucher depuis la surface de la Terre.

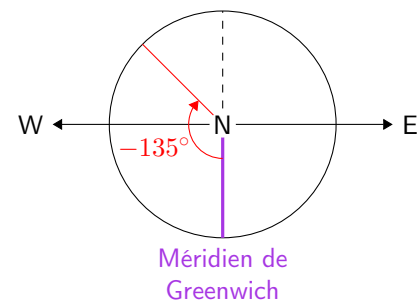
Latitude, longitude

Définition 4

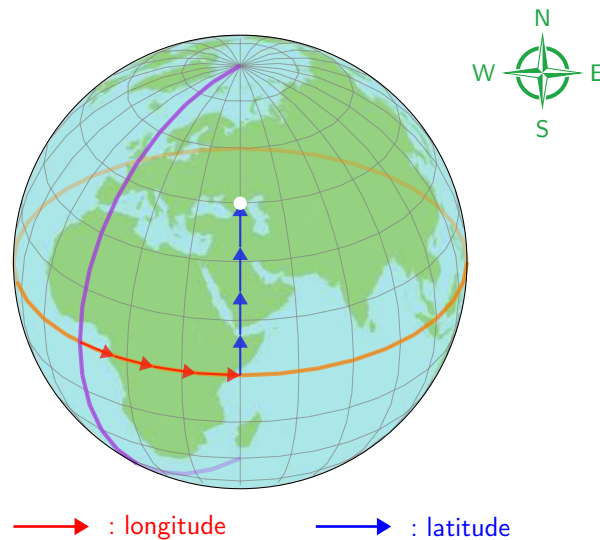
- ▶ On appelle *latitude* d'un parallèle une mesure de l'angle formé au centre de la Terre entre l'équateur et ledit parallèle. Cet angle est compté positivement si le parallèle est dans l'hémisphère Nord, et négativement si le parallèle est dans l'hémisphère Sud. Une latitude s'exprime donc comme une mesure d'angle entre -90° et 90° .
- ▶ On appelle *longitude* d'un méridien une mesure de l'angle formé au centre de la Terre entre le méridien de Greenwich et ledit méridien. Cet angle est compté positivement si le méridien est à l'Est du méridien de Greenwich, et négativement sinon. Une longitude s'exprime donc comme une mesure d'angle entre -180° et 180° .
- ▶ Les *coordonnées sphériques* d'un point à la surface de la Terre sont la donnée de la latitude et de la longitude du parallèle et du méridien desquels ledit point est à l'intersection.



Latitude de 60°



Longitude de -135°



Exercice 5

- 1► Sur une horloge analogique standard, quel angle, en degrés, l'aiguille des minutes parcourt-elle en 1 minute ?
- 2► En supposant que l'aiguille des minutes progresse continûment à vitesse constante, quel angle, en degrés, parcourt-elle en 1 seconde ?

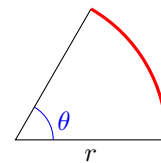
Définition 6

Arcminutes, arcsecondes

- L'*arcminute* (ou *minute d'arc*, ou *minute d'angle*) est une mesure d'angle : 1 arcminute, noté $1'$, correspond à $\frac{1}{60}$ de degré.
- L'*arcseconde* (ou *seconde d'arc*, ou *seconde d'angle*) est une mesure d'angle : 1 arcseconde, noté $1''$, correspond à $\frac{1}{60}$ d'arcminute.

Exercice 7

On admet que la longueur d'un arc de cercle est proportionnelle à l'angle du secteur qui le sous-tend.



- 1► On considère un cercle de rayon r .
 - a► Si on considère le cercle complet comme un secteur, à quel angle correspond-il ?
 - b► Rappeler l'expression de la longueur d'un cercle.
 - c► En s'aidant éventuellement d'un tableau de proportionnalité proprement posé, en déduire en fonction de r et de θ la longueur d'un arc de cercle sous-tendu par un secteur d'angle θ (comme sur le schéma ci-contre).
- 2► Le rayon de la Terre est d'environ 6 371 km.
 - a► Les fuseaux horaires s'appuient sur une division en 24 méridiens régulièrement espacés. Quelle distance faut-il marcher à l'équateur pour traverser un fuseau horaire ?
 - b► Quelle latitude atteint-on si l'on marche 100 km en direction du Nord, partant de l'équateur ?
- 3► On admet que le rayon d'un parallèle est égal au rayon de la Terre multiplié par le cosinus de sa latitude.

- a► Si je marche 1 m le long de l'équateur, de combien d'arcsecondes progresse ma longitude ?
- b► Si je marche 1 m le long du parallèle de latitude 60° , de combien d'arcsecondes progresse ma longitude ?

Définition 8**Coordonnées GPS**

Les *coordonnées GPS* d'un point à la surface de la Terre sont la donnée de sa latitude sans son signe, affublée d'un « N » ou d'un « S » selon que ce point est de latitude positive ou négative ; et la donnée de sa longitude sans son signe, affublée d'un « E » ou d'un « W » selon que la longitude est positive ou négative. Exemple : un point de latitude 60° et de longitude -135° a pour coordonnées GPS $60^\circ N 135^\circ W$.

Les coordonnées GPS sont en général exprimées de trois manières :

- « Degrés Minutes Secondes » (DMS) : On exprime la latitude et la longitude selon un nombre entier de degrés, d'arcminutes (entre $0'$ et $59'$) et un nombre éventuellement décimal d'arcsecondes (bien qu'en pratique, 1 arcseconde donne une précision suffisante).
- « Degrés Minutes » (DM) : On exprime la latitude et la longitude selon un nombre entier de degrés et un nombre décimal d'arcminutes (en pratique, on donne les arcminutes au dixième près, ce qui donne une précision suffisante).
- « Degrés décimaux » (DD) : On exprime la latitude et la longitude sous la forme d'un nombre décimal (en pratique, au dix-millième, ce qui donne une précision suffisante).

Exemple en DMS : les coordonnées $60^\circ 55' 34'' N 135^\circ 05' 48'' W$ se lisent « 60 degrés, 55 minutes 34 secondes Nord, 135 degrés 5 minutes 48 secondes Ouest ».

Exemple 9

La ville de Bordeaux est aux coordonnées $44^\circ 50' 16'' N 0^\circ 34' 46'' W$ dans le format DMS, ce qui correspond à $44^\circ 50,3' N 0^\circ 34,8' W$ dans le format DM (arrondi au dixième), soit encore $44,8378^\circ N 0,5794^\circ W$ dans le format DD.

Exercice 10

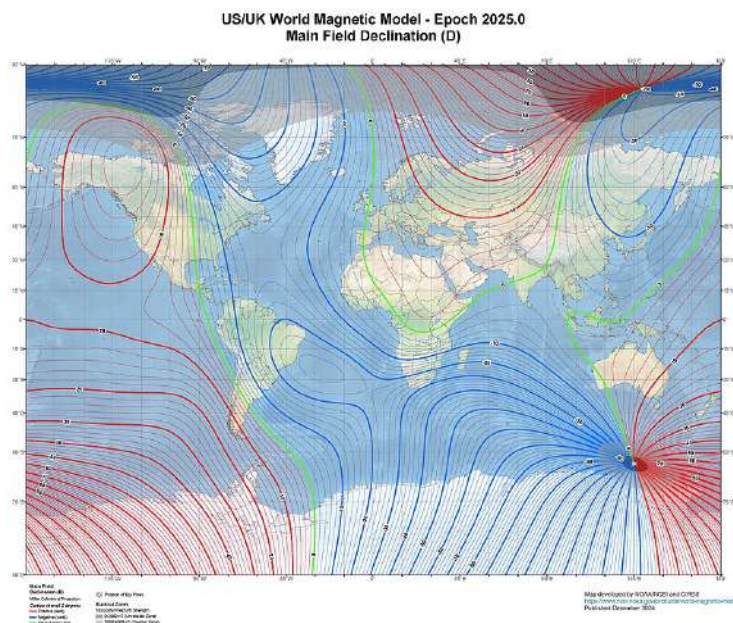
Compléter les cases du tableau suivant exprimant des coordonnées selon les quatre systèmes : (latitude, longitude) avec signe, DD (arrondi au dix-millième de degré), DMS (arrondi à l'arcminute), DM (arrondi au dixième d'arcminute).

Coordonnées	DD	DM	DMS
			$31^\circ 37' 46'' N 7^\circ 58' 52'' W$
		$48^\circ 51,4' N 2^\circ 21,1' E$	
	$40,7128^\circ N 74,0061^\circ W$		
$(-22,9097^\circ, -43,1764^\circ)$			
		$33^\circ 51,4' S 151^\circ 11,6' E$	

2 – Les premiers outils de localisation

Pour se déplacer sans se perdre, on a besoin de repères, que ce soit un bâtiment, une montagne au loin ou même les astres. Mais comment faire lorsqu'on ne peut voir aucun repère ? La découverte des propriétés magnétiques de certaines pierres a conduit, au XI^e siècle, à leur utilisation à des fins d'orientation : une pierre magnétique, mise en flottaison sur de l'eau, tourne et s'oriente en direction du Sud – il s'agit de ce qu'on peut considérer comme les premières boussoles–.

L'utilisation de boussoles dans la navigation maritime a pour la première fois été relatée dans des écrits datant de la fin du XI^e siècle. En haute mer, il n'y a aucun repère visible, un bateau est entouré par de l'eau jusqu'à l'horizon, rendant indispensable l'utilisation d'outils pour se repérer. Cependant, bien que la boussole soit pratique pour indiquer une direction en toutes circonstances, ce n'est pas un outil très précis ; ou plutôt, il requiert une précision phénoménale pour être fiable : par exemple, une erreur de 0,01° sur la direction partant de France fait dévier un bateau jusqu'à 60 km de sa cible en arrivant aux États-Unis. Pire encore, la boussole souffre de la *déclinaison magnétique* : en effet, le *pôle magnétique*, c'est-à-dire le point vers lequel la boussole pointe, n'est pas au même endroit que le *pôle géographique*, celui par lequel passe l'axe de rotation de la Terre. Pour compliquer davantage les choses, les pôles magnétiques se déplacent avec le temps, et l'interaction complexe entre les lignes du champ magnétique terrestre rendent irrégulières les corrections à apporter à la lecture d'une boussole.

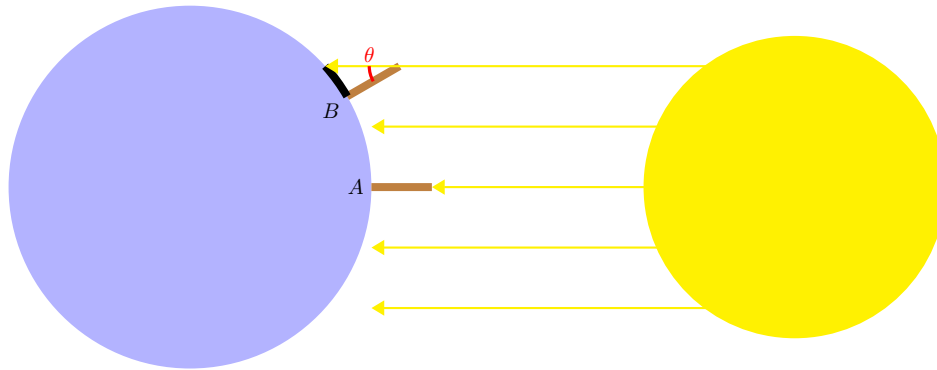


Il était donc nécessaire de trouver un moyen de se repérer à la surface du globe. Bien que les Grecs ne sont certainement pas les premiers à avoir découvert la sphéricité de la Terre, c'est à Ératosthène, au III^e siècle av. J.-C., que l'on attribue les notions de latitude et de longitude, et surtout une mesure de la circonférence de la Terre.

Exercice 11 (Calcul de la circonférence de la Terre selon Ératosthène)

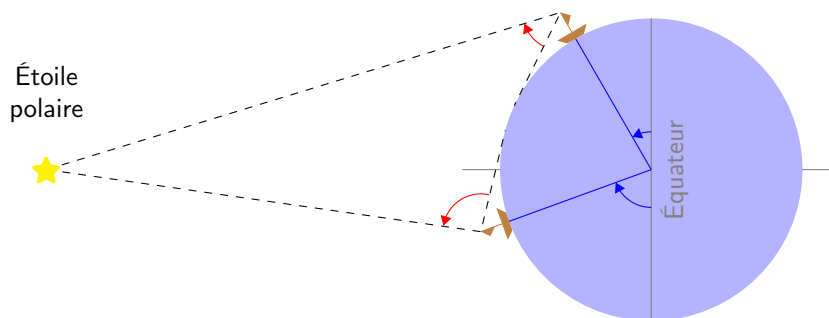
L'idée d'Ératosthène repose sur le constat suivant : le Soleil étant très éloigné de la Terre, les rayons qui nous parviennent sont tous parallèles. Comme la Terre est sphérique, si l'on plante un piquet bien droit à la surface de la Terre, l'ombre qu'il projette est plus ou moins longue selon la latitude. Il existe même une latitude particulière où le piquet est parallèle aux rayons du Soleil, et donc ne projette pas d'ombre.

La méthode d'Ératosthène (à quelques simplifications près) est donc la suivante : il repère un point *A* où le piquet n'a pas d'ombre, marche jusqu'à un point *B* en mesurant la distance parcourue, et mesure la longueur de l'ombre au point *B*. À échelle humaine, la Terre apparaît suffisamment plate pour assimiler l'ombre à un segment (droit). Un peu de trigonométrie permet alors de calculer l'angle formé par le piquet et les rayons de Soleil.



- 1► Reproduire le schéma ci-dessus et le compléter en faisant apparaître le centre O de la Terre et les rayons de la sphère terrestre reliant le centre aux deux piquets.
- 2► Si θ est l'angle mesuré au point B , combien mesure l'angle $\widehat{AOB}$?
- 3► Application : Ératosthène a marché 950 km entre le point A et le point B , et a mesuré un angle de $7,2^\circ$.
 - a► En vous aidant éventuellement de la première question de l'exercice 7, calculer la circonférence de la Terre selon les mesures d'Ératosthène.
 - b► Les mesures modernes s'accordent à dire que le rayon de la Terre est d'environ 6371 km. En déduire la circonférence de la Terre au kilomètre près.
 - c► Calculer l'erreur relative d'Ératosthène (l'écart en pourcentage entre le calcul d'Ératosthène et la valeur réelle).
 - d► Expliquer les sources possibles de cet écart.

La méthode d'Ératosthène permet, dans une certaine mesure, de calculer la latitude sur terre. Sur la mer en revanche, cette méthode est caduque : en effet, il est crucial que le piquet soit parfaitement vertical par rapport à la surface de la Terre, ce qui est impossible à mettre en place en pratique sur le pont d'un bateau qui tangue. Il a donc fallu prendre des points de référence plus stables et prédictibles. L'*astrolabe planisphérique*, que l'on retrouve mentionné dans des écrits datant du II^e siècle av. J.-C., est un outil permettant de mesurer la hauteur des astres dans le ciel. Son principe a été adapté pour la navigation maritime, avec l'invention de l'*astrolabe nautique* au XV^e siècle, suivi de l'*octant* puis du *sextant* au XVIII^e siècle, simplifiant grandement les mesures. Le schéma ci-dessous illustre, avec l'exemple de l'Étoile polaire (plus ou moins alignée avec l'axe Nord-Sud), comment la latitude fait varier la hauteur apparente des astres. En pratique, on mesure l'angle entre l'horizon, à hauteur fixe, et l'astre de référence. Grâce aux cartes célestes, on peut prédire la hauteur apparente des astres en fonction de la latitude ; inversement, on peut donc calculer à l'aide de ces mêmes cartes, la latitude à partir de la hauteur des astres !



Là où le calcul de la latitude relève d'un exercice de géométrie relativement simple, celui de la longitude est beaucoup plus ardu. En effet, le problème est que la Terre tourne : pour la latitude, ce n'est pas un problème, car les parallèles sont alignés avec la rotation de la Terre ; pour la longitude, les méridiens bougent à chaque instant, plus précisément, chaque heure, la Terre tourne de 15° environ. Cette rotation empêche d'avoir un repère immobile comme l'Étoile polaire, qui ne bouge presque pas dans le ciel. Des méthodes ont été développées pour calculer géométriquement la longitude, comme la méthode des distances lunaires, qui exploite le fait que la Lune étant bien plus proche que

les étoiles, sa position ne défile pas à la même vitesse que l'arrière plan. Ces méthodes souffrent grandement des imprécisions de mesure et de calculs – imprécisions qui peuvent avoir des conséquences fatales, comme le désastre naval de Sorlingues –. Au XVIII^e siècle, une nouvelle méthode est développée : elle repose sur l'heure. En effet, supposons que notre horloge soit réglée, au départ du voyage, à midi au zénith, c'est-à-dire au moment où le Soleil est le plus haut dans le ciel. Lorsqu'on voyage vers l'Est ou l'Ouest, le moment où le zénith ait lieu change ⁽¹⁾ : ainsi, en mesurant l'heure, selon notre montre préalablement réglée, du *midi local*, c'est-à-dire le moment du zénith à l'endroit actuel, et l'écart de temps permet de déduire la longitude. Pour simplifier : imaginons que le midi local à lieu alors que notre montre affiche 14h. Cela signifie que le midi arrive 2h plus tard qu'à notre point de départ. Or, comme dit plus tôt, la Terre tourne d'environ 15° par heure (vers l'Est), cela signifie donc que l'on a voyagé de 30° vers l'Ouest.

L'efficacité de cette méthode repose sur la capacité des horloges embarquées à ne pas se dérégler avec le temps. En effet, par exemple, une erreur de 4 secondes entraîne un décalage de plus de 1 km au niveau de l'équateur. Et en général, une horloge se dérègle de plus en plus avec le temps, augmentant donc de plus en plus le décalage. Les difficultés rencontrées par les horlogers résidaient dans deux facteurs : l'instabilité des bateaux rendaient difficilement utilisables les horloges à balancier, et les variations d'humidité et de température faussaient la régularité des montres à ressort. Il faudra attendre la fin du XVIII^e siècle pour que les horloges marines soient suffisamment fiables pour que leur utilisation supplante toutes les autres méthodes.

3 – La localisation par satellite

A – La trilatération

Activité 12

- 1► Alice a caché un trésor quelque part en France. Afin de vous permettre de le retrouver, elle a laissé quelques indices dans le texte ci-dessous. En vous aidant des cartes en Annexe⁽¹⁾ et de ficelles, ainsi que de cartes interactives (OpenStreetMap, Google Maps, ...), trouvez le trésor d'Alice.

« J'ai enterré un trésor à 250 km de Bordeaux, et à 320 km de Paris. J'aurais peut-être dû l'enterrer plus près de Nantes, le chemin du retour va être très long... À peine arrivé sur le lieu du trésor, j'ai entendu le sifflet d'un train. C'était sans doute le train parti de la gare à 14h00 et 0 secondes, renommé pour sa remarquable ponctualité ! Enfin, ce n'est pas l'heure à laquelle j'ai entendu le sifflet, car le son voyageant à 340 mètres par seconde environ, le temps que le son arrive jusqu'à moi, ma montre affichait 14h00, 12 secondes et 12 centièmes (oui, ma montre est très précise)... Je n'ai pas vu le temps passer ; en levant les yeux, j'ai aperçu un nuage menaçant venant de l'est. Quelques instants plus tard, la foudre a frappé le clocher de l'église Saint-Alpinien. Le son fracassant du tonnerre m'est parvenu 9,29 secondes plus tard. J'ai terminé rapidement d'enfouir le trésor, puis j'ai longé la D941 jusqu'à revenir en ville. »

- 2► Un sonar permet de mesurer la distance entre lui-même et un objet distant. Il envoie un court son et mesure le temps que l'écho met à revenir.
 - a► Si on note t le temps écoulé entre l'émission du son et son retour, comment calculer la distance d qui sépare l'objet du sonar ?
 - b► Application numérique : dans l'eau, le son se déplace de 1500 mètre chaque seconde. Si l'écho revient 60 ms après son émission, à quelle distance se trouve l'objet détecté ?
- 3► Pour annoncer le Nouvel an, des stations radio à Nantes, Toulouse et Strasbourg émettent un message de célébration à minuit pile, à la microseconde près. Bob reçoit le message toulousain avec 917 microsecondes de retard, le message nantais 384 μ s plus tard, et le message strasbourgeois 267 μ s encore après. Où Bob a-t-il fêté le Nouvel an ?

(1). C'est le principe des fuseaux horaires !

(1). Afin de disposer de cartes suffisamment grandes et détaillées, celles-ci ont été découpées sur quatre pages. Découpez proprement le bord de chaque morceau de carte, et assemblez-les pour reformer la grande carte.

Trilatération

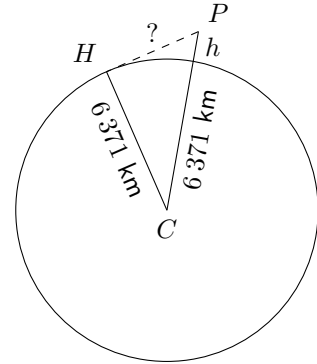
Définition 13

La *trilatération* est une méthode permettant de situer un point en connaissant sa distance par rapport à des points de référence. Dans le plan, il faut 3 points de référence (conduisant à chercher l'intersection de trois cercles), dans l'espace il en faut 4 (conduisant à chercher l'intersection de quatre sphères).

Activité 14

L'objectif de ce problème est de déterminer la distance de l'horizon sur Terre à hauteur humaine. On fera l'approximation d'une Terre parfaitement sphérique. On note C le centre de la Terre, P le point de vue, et H le point à l'horizon. La situation peut donc être représentée par le schéma ci-contre.

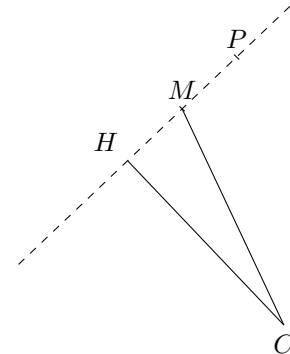
L'objectif est donc de trouver la longueur PH , marquée par un point d'interrogation.



Partie A : mise en place mathématique

L'objectif de cette partie est de montrer que l'angle $\widehat{CHP}$ est droit.

- 1► Justifier que le point H est le point sur la droite (PH) qui est le plus proche de C .
- 2► Pour clarifier la situation, on reproduit une partie du schéma ci-dessus. On place également un point M sur (PH) quelconque pour répondre aux questions suivantes.
 - a► Démontrez la proposition suivante :
Soit $M \in (HP)$.
Si le segment $[CM]$ ne forme pas un angle droit avec la droite (HP) , alors M n'est pas le point de (PH) qui est le plus proche de C .
 - b► Énoncez la contraposée de cette proposition.
 - c► Grâce à la contraposée appliquée à H au lieu de M , que peut-on conclure sur l'angle en H ?



Partie B : application à des antennes

- 1► À l'aide de la partie A, justifier qu'à une hauteur de 50 m, l'horizon se situe à un peu plus de 25 km.
- 2► On dresse une antenne de hauteur 50 m. On considère que l'on capte le signal de l'antenne tant que celle-ci apparaît au-dessus de l'horizon.
Par quelle forme géométrique peut-on approcher la zone géographique dans laquelle on capte le signal de l'antenne ? Calculer sa surface.
- 3► En déduire une minoration grossière du nombre de telles antennes qu'il faudrait pour couvrir toute la surface terrestre. Cela vous semble-t-il réalisable ?
Aire d'une sphère de rayon r : $\mathcal{A} = 4\pi r^2$.

B – L'histoire du GPS

Les systèmes utilisant des antennes émettant des ondes radio permettant de se localiser sont appelés *système de radionavigation*. Des systèmes *terrestres*, c'est-à-dire dont les antennes émettrices se trouvent à la surface de la Terre, existent depuis le début du XX^e siècle. Utiliser des antennes pour se localiser est cependant limité, et pas toujours possible, parce que les surfaces à couvrir sont trop vastes ou trop inaccessibles. Par exemple, l'armée américaine ne pouvait pas installer un réseau d'antennes dans la jungle vietnamienne ou dans le désert irakien en amont ou pendant leurs opérations militaires, ou au fond des océans pour repérer leurs sous-marins. L'idée de mettre les antennes en

orbite afin de surmonter ces difficultés voit le jour en 1957, avec la course à l'espace de la Guerre Froide. En effet, les ingénieurs américains ont remarqué qu'en analysant les signaux envoyés par le satellite Spoutnik 1, ils pouvaient déterminer sa position. Un système de localisation par satellite prendrait alors le problème à l'envers : si on connaît la position des satellites, alors on peut en déduire notre position sur Terre. Cette idée sera concrétisée en 1964 avec les quatre satellites du système *Transit*. Celui-ci permettait de déterminer une position à 200 m près toutes les heures. Son usage était initialement réservé à la marine militaire américaine.

Après plusieurs itérations et améliorations, l'armée américaine lance en 1978 le premier satellite du système *GPS* (*Global Positioning System*)⁽¹⁾. Tout comme le système Transit, le GPS était initialement développé par et pour l'armée américaine. Cette première salve de satellites a essentiellement servi à éprouver le concept : le « vrai » système GPS est déployé entre 1985 et 1990. En 1995, le gouvernement américain ouvre le système GPS aux usages civils, notamment pour l'aviation, mais avec une précision bridée à une centaine de mètres, contre une dizaine de mètres pour les usages militaires – cette limitation sera levée en 2000, permettant le développement de technologies de navigation modernes, celles que l'on appelle « GPS » abusivement–.

La flotte du système est constamment renouvelée et améliorée, les premiers satellites se dégradant, et les technologies progressant. Aujourd'hui, 31 satellites sont en activité, permettant de toujours avoir au moins 4 satellites au-dessus de n'importe quel point du globe. D'autres systèmes de géolocalisation par satellite ont également été déployés : le système soviétique *Glonass* (24 satellites) dans une temporalité similaire aux États-Unis, l'europpéen *Galileo* (30 satellites) à partir de 2017, et le chinois *Beidou* (35 satellites), opérationnel dans la région chinoise et ses alentours à partir de 2001, puis globalement à partir de 2020.

C – Limitations techniques du GPS

Le GPS repose sur trois groupes d'appareils, appelés *segments* : les satellites (*segment spatial*), les téléphone et autres récepteurs GPS (*segment utilisateur*), et des stations au sol (*segment contrôle*).

Comme illustré dans les précédentes activités, le facteur clé de la précision du positionnement GPS réside dans la capacité du système entier à correctement chronométrer les échanges d'informations. Pour parcourir 10 mètres, une onde radio n'a besoin que d'une trentaine de nanosecondes⁽²⁾, ainsi pour permettre une telle précision, les satellites embarquent des horloges atomiques. De là, le principe du fonctionnement du système GPS se résume simplement : chaque satellite émet régulièrement un signal contenant l'heure exacte d'émission dudit signal, et ses coordonnées spatiales⁽³⁾. Lorsqu'il reçoit le signal, le récepteur note l'heure de réception, et en calculant l'écart de temps, permet d'en déduire sa distance au satellite. En répétant l'opération avec plusieurs autres satellites, le récepteur peut calculer sa position⁽⁴⁾. Mais tout ceci est bien trop simple... en effet, cette procédure a plein de défauts :

- ▶ Le satellite a beau mesurer l'heure avec une précision atomique, le récepteur n'est pas doté d'un matériel de la même précision !
- ▶ Depuis Albert Einstein et sa théorie de la relativité⁽⁵⁾, on sait que l'horloge des satellites ne s'écoule pas au même rythme qu'à la surface de la Terre. Cela complique davantage la synchronisation des horloges entre le récepteur et les satellites.
- ▶ Les différentes couches de l'atmosphère freinent et dévient les signaux radio émis par les satellites.
- ▶ Les satellites doivent en permanence connaître leur position exacte dans l'espace pour pouvoir la communiquer aux utilisateurs. Mais eux ne peuvent pas utiliser de GPS pour se repérer...

Le dernier point est réglé grâce au segment contrôle : en effet, comme expliqué plus haut avec l'inspiration du satellite Spoutnik, si l'on connaît les coordonnées de quelques points de référence à la surface de la Terre, alors on peut appliquer le principe du GPS « à l'envers » pour en déduire les coordonnées du satellite. Les stations de

(1). Attention, bien que « GPS » soit entré dans le langage courant comme acronyme désignant toute technologie de géolocalisation, il s'agit en fait d'un système parmi tant d'autres !

(2). Une nanoseconde est un milliardième de seconde.

(3). D'autres informations sont également envoyées, pour en savoir plus, cherchez « Trame NMEA » dans les manuels scolaires ou votre moteur de recherche favori.

(4). Voir le paragraphe sur la trilatération.

(5). Pour les esprits curieux, allez lire le [billet de blog](#) de Science Etonnante, ou allez voir sa vidéo sur la relativité !

contrôle, contrairement aux récepteurs, disposent d'horloges atomiques, pour pouvoir calculer avec la plus grande précision possible la position de chaque satellite, et leur transmettre pour qu'ils restent à jour.

Les autres problèmes sont réglés en considérant l'écart de temps comme l'une des inconnues du problème : on rajoute des équations à résoudre, il faut donc augmenter le nombre de mesures pour pouvoir en trouver la solution. En pratique, on se sert d'au moins quatre satellites ; d'où le fait que les orbites de ces derniers sont telles qu'au moins quatre satellites survolent chaque point du globe.

4 – Les centrales inertielles

Activité 15

Alice part en exploration souterraine, à la recherche de trésors, à l'aide de sa foreuse. Cette dernière ne dispose pas d'un système de positionnement, et dans l'obscurité des profondeurs de la terre, pas moyen de trouver des points de repère ! Le tableau de bord de la foreuse d'Alice affiche trois informations : la vitesse lorsqu'elle avance, un chronomètre, et l'angle de rotation de l'appareil lorsqu'il tourne (un angle négatif signifie une rotation dans le sens horaire, et un angle positif correspond à une rotation dans le sens anti-horaire). Son journal de bord affiche les actions suivantes : 3s à 5m/s, virage à -45° , 5s à 5m/s, virage à $+135^\circ$, 10s à 3m/s, virage à $+60^\circ$, 8s à 1m/s, virage à -130° , 4s à 5m/s, virage à -90° , puis 8s à 4m/s. Par ailleurs, Alice a commencé son forage en creusant à la verticale.

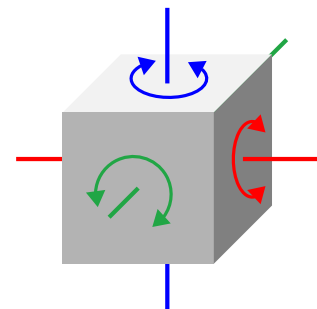
En vous aidant du plan en Annexe, répondez à la question suivante : quels trésors Alice a-t-elle rencontrés en chemin ?

Aussi pratique que le GPS puisse être, il n'est pas infallible : sa précision peut être limitée, et il n'est pas toujours accessible, voire souhaitable. Une *centrale inertielle* est un dispositif permettant de se repérer sans aucune aide extérieure ; c'est comme une boussole beaucoup plus avancée.

Exemple 16

- Les avions en sont équipés notamment pour des raisons de sûreté, car le fait de n'avoir aucun sol en-dessous de soi, et le fait de se déplacer à grande vitesse, augmentent l'imprécision de la mesure de position par GPS.
- L'eau bloque le signal GPS, le rendant inaccessible aux sous-marins. Les sous-marins militaires contraints à une discrétion absolue ne peuvent utiliser aucun outil qui pourrait compromettre leur position.
- Le GPS permet de mesurer des positions sur Terre jusqu'à une certaine altitude, mais les sondes spatiales, satellites et autres vaisseaux au-delà de cette limite n'y ont pas accès.

Dans l'activité précédente, Alice ne mesure que sa propre vitesse et sa rotation, et cela suffit à connaître sa position à chaque instant. Mais Alice est un être bidimensionnel, elle ne peut tourner que dans le sens horaire ou le sens anti-horaire. En trois dimensions, il y a 6 sens de rotation (pour les fans d'aviation : les trois axes de rotation sont ceux du *tangage*, du *roulis* et du *lacet*). De plus, contrairement à une voiture, un avion, un sous-marin ou une sonde spatiale ne dispose pas de roues dont on peut mesurer la vitesse de rotation pour en déduire la vitesse du véhicule. Plutôt que de mesurer une vitesse, les centrales inertielles mesurent l'*accélération*, qui est la variation de la vitesse.



Une centrale inertielle est donc équipée de trois accéléromètres, pour mesurer l'accélération selon les trois directions (droite-gauche, haut-bas, devant-derrrière) et de trois gyromètres pour mesurer la vitesse angulaire selon les trois axes illustrés ci-dessus. ⁽¹⁾

(1). Pour une illustration, voir la vidéo [How do submarines know where they are?](#) de Veritasium sur YouTube.

Remarque 17

Tous les smartphones modernes sont équipés des mêmes éléments qu'une centrale inertielle. À l'aide d'une application tierce, ou directement depuis certains paramètres de votre téléphone, vous pouvez afficher les données de ces capteurs en temps réel. Essayez donc !

5 – Représentation des données géographiques

A – Les cartes géographiques physiques : histoire et limites

i – Des cartes pour la navigation

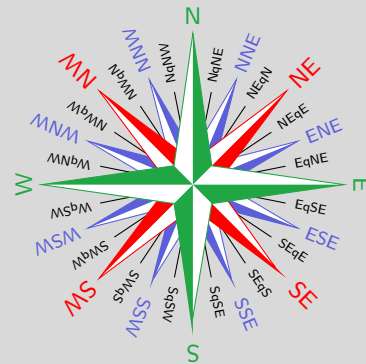
Les premières cartes, du rocher de Bedolina jusqu'au XVII^e siècle, reposaient essentiellement sur l'observation et la mesure du *cap*, c'est-à-dire la direction vers laquelle on regarde.

Définition 18

La *rose des vents* est une représentation des différentes directions. On y retrouve :

- ▶ Les quatre points cardinaux : Nord (N), Sud (S), Est (E) et Ouest (W) (en vert ci-contre),
- ▶ Quatre directions intermédiaires inclinées à 45° par rapport aux points cardinaux : Nord-Est (NE), Sud Est (SE), Sud-Ouest (SW) et Nord-Ouest (NW) (en rouge ci-contre),
- ▶ Huit directions intermédiaires inclinées de 22,5° par rapport aux points cardinaux : Nord-Nord-Est (NNE), Est-Nord-Est (ENE), ... (en bleu ci-contre),
- ▶ Seize directions intermédiaires complétant la division du tour complet en 32 secteurs de 11,25° chacun : Nord-Quart-Nord-Est (NqNE), Nord-Est-Quart-Nord (NEqN), ... (en noir ci-contre).

Un *rhumb* est une mesure d'angle, correspondant à 11,25°. La rose des vents est donc divisée en 32 rhumbs.

Rose des vents, rhumb**Remarque 19**

Pour les retrouver facilement,

- ▶ le nom d'une direction secondaire (NO, NE, SO, SE) est construit ainsi : elles portent les deux noms des deux points cardinaux qui l'entourent, la première lettre est Nord ou Sud.
- ▶ le nom d'une direction tertiaire (ENE, ESE, ...) est construit ainsi : écrivez le nom du point cardinal le plus proche, puis de la direction secondaire (NE, NW, SE, SW) la plus proche.
- ▶ le nom d'une direction « quart de vent » est construit ainsi : écrivez le point cardinal ou la direction secondaire (NE, NW, SE, SW) la plus proche, « q » pour « quart » (ou la fraction directement), puis le deuxième point cardinal ou la deuxième direction secondaire la plus proche.

Définition 20*Portulan*

Un *portulan*, ou *carte à rhumb*, est un type de carte permettant de placer des lieux selon leurs directions relatives. On y trace des lignes selon les rhumbs de la rose des vents à partir de certains points de la carte, donnant un aspect de toile d'araignée à la carte.

Exemple 21

Ci-dessous se trouve un portulan⁽¹⁾ représentant les chefs-lieux en France métropolitaine. Les différentes directions ont été colorées comme la rose des vents plus haut pour faciliter la lecture. On y lit par exemple que pour se rendre de Bordeaux à Lyon, il faut suivre la direction Est-Quart-Nord-Est ; pour aller de Lyon à Toulouse, il faut suivre la direction Sud-Ouest ; pour aller de Créteil à Bar-le-Duc, il faut suivre la direction Est.



ii – Les problèmes de projection et d'échelle

Il est invraisemblable de considérer un lien entre les portulans et les notions de latitude et de longitude. En effet, les portulans n'ont pas vocation à fournir des coordonnées, les lignes de rhumb indiquent uniquement des directions. Dit grossièrement, les portulans sont des cartes routières.

Par ailleurs, les notions de latitude et de longitude, décrites par Ptolémée au II^e siècle dans sa *Géographie*, inconnues de l'Europe latine jusqu'au XV^e siècle, soit postérieurement à l'invention des portulans, au début du XIV^e siècle.

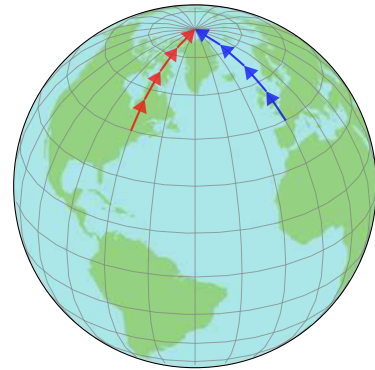
Les portulans soulèvent un problème : des directions identiques sont représentées parallèlement. À première vue, ça ne semble pas choquant, mais pour mieux voir, il faut prendre un peu de recul.

(1). Ou plutôt, une tentative de recreation de la main d'un amateur ! Les vrais portulans choisissent mieux les points d'où partent les lignes directionnelles, pour un rendu plus harmonieux et lisible. Allez voir des exemples en ligne.

Deux personnes, partant de Paris et de New York, se dirigeant vers le Nord, suivent-elles des directions parallèles ? Impossible, puisqu'elles vont se croiser au pôle Nord, alors que le parallélisme suppose par définition l'absence de point de rencontre !

Mais les cartes portulan sont vieilles de plusieurs siècles, de nos jours on doit bien savoir faire des cartes qui corrigent ce problème de parallélisme... non ?

La réponse est : vous n'avez jamais vu, et ne verrez jamais, une carte fidèle à la réalité !



Activité 22 (La trahison des images)

- 1► À l'aide de la carte du monde en Annexe, dire si les affirmations suivantes sont vraies ou fausses :
 - A► Le Canada est aussi grand que le Brésil.
 - B► L'Antarctique est de taille comparable à l'Australie.
 - C► L'Afrique est plus grande que la Russie et le Groenland réunis.
- 2► Rendez-vous sur le site thetruesize.com. Dans la barre de saisie, entrez Russia, Greenland, Antarctica et Canada. Glissez les calques en surbrillance. Avez-vous correctement jugé les affirmations précédentes ?

Information 23

Projections

Il est mathématiquement impossible de produire une carte de la Terre qui respecte à la fois les distances, les angles et les aires.

Le compromis d'un ou plusieurs de ces critères donne lieu à différentes cartes, que l'on dit être différentes *projections*. La plus fréquente est la *projection de Mercator* (1569) (utilisée pour la carte du monde en annexe, ou le site de l'activité précédente) : elle conserve les angles, mais pas les distances ni les aires. Sa propriété de conservation des angles en a fait la projection privilégiée pour la navigation.

Activité 24

Selon vous, quel est le plus court chemin entre Paris et New York, en avion ? Après avoir proposé une réponse à l'aide du planisphère en Annexe, allez sur le site <https://user.numazu-ct.ac.jp/~tsato/webmap/sphere/great-circle/globe-e.html>. Assurez-vous que les cases « Great Circle Arc » et « Rhumb Lines » sont cochées, dans le menu qui apparaît lorsque vous survolez l'icône en haut à droite de l'écran.

Remarque 25 (Le problème de l'échelle)

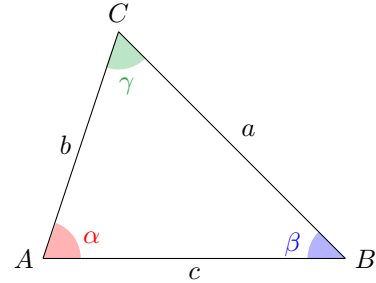
Si vous avez fait de grands voyages en voiture avant la démocratisation du GPS, ou si vous avez fait l'activité utilisant les cartes en Annexe, vous remarquez que les cartes physiques sont limitées : si l'on veut représenter un grande échelle, il faut sacrifier les plus petits détails pour ne pas surcharger la carte ; si l'on veut représenter des détails fins, il faut se limiter à une zone géographique restreinte, sans quoi la carte serait immensément grande.

iii – Les mathématiques au service de la mesure topographique

Exercice 26 (Loi des sinus)

Soit ABC un triangle non aplati. On note ses longueurs et ses angles conformément à la figure ci-contre.

- 1► Rappeler les relations trigonométriques donnant le cosinus, le sinus et la tangente d'un angle dans un triangle rectangle en fonction de deux côtés. (*Indication : CAH SOH TOA...*)
- 2► On note h la hauteur de ABC issue de C .
 - a► Exprimer h en fonction de a et d'un sinus.
 - b► Exprimer h en fonction de b et d'un sinus.
 - c► En déduire que $\frac{a}{\sin(\alpha)} = \frac{b}{\sin(\beta)}$.
- 3► Montrer de même que $\frac{b}{\sin(\beta)} = \frac{c}{\sin(\gamma)}$. (*Indication : utiliser une autre hauteur.*)



On vient de montrer la *loi des sinus* : dans tout triangle non aplati ABC , avec les notations ci-dessus,

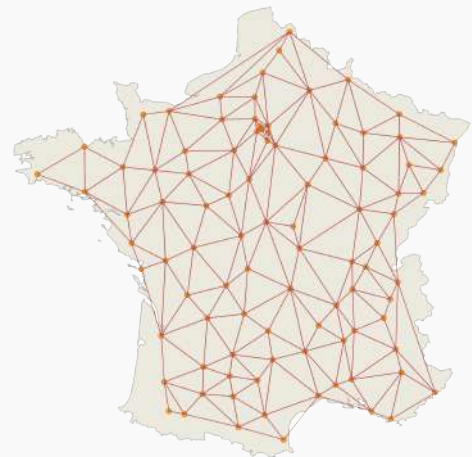
$$\frac{a}{\sin(\alpha)} = \frac{b}{\sin(\beta)} = \frac{c}{\sin(\gamma)}$$

- 4► Établir un protocole (que mesure-t-on, avec quels outils, que fait-on des résultats...) permettant de calculer la distance entre deux lieux.
- 5► Application : Bob souhaite connaître la distance entre la Place de la Concorde et le Palais Garnier. Malheureusement, les ruelles qui relient ces deux endroits sont tortueuses, il est donc impossible de mesurer directement la distance les séparant. Bob remarque la Rue de Rivoli, reliant le Musée du Louvre à la place de la Concorde est remarquablement rectiligne : il mesure 1,32 km en marchant de l'un à l'autre en ligne droite. Il se débrouille également pour mesurer les angles aux sommets du triangle reliant ces trois lieux : l'angle au Louvre mesure $46,53^\circ$ environ, celui au Palais Garnier mesure $66,35^\circ$ environ, et celui à la Place de la Concorde mesure $67,12^\circ$ environ. Aidez Bob à l'aide de la loi des sinus.
- 6► Contrôlez votre résultat à l'aide de l'outil de Mesure de distance d'une carte numérique (OpenStreetMap, Géoportail, Google Maps, ...).

Information 27

Triangulation & Cassini

Aux XVII^e et XVIII^e siècles, la famille Cassini entreprend de cartographier la France continentale dans son entièreté. Pour la première fois, les mathématiques sont utilisées pour cartographier géométriquement : les Cassini utilisent la *triangulation*. Ils découpent la France en un maillage de petits triangles, en mesurent les angles, puis en utilisant la loi des sinus, calculent les longueurs des côtés de ces triangles. Ce travail s'est étendu sur quatre générations, et a fourni une carte à l'échelle 1/86 400, la plus précise de son époque.



iv – Problèmes de conservation, reproduction et péremption

Remarque 28 (Le choix des matériaux)

Autrefois en parchemin, en toile ou en papier, les cartes, comme tout objet, s'abîment au fil de leur utilisation et de leurs conditions de conservation. Le papier, moins cher, est plus fragile, est sensible à l'humidité (et donc difficile à exploiter en mer), et finit par se déchirer à force d'être pliées. Le parchemin, comme la toile, plus robuste et résistant à l'eau de mer, est coûteux et plus long à préparer.

Avant le développement des techniques d'imprimerie, les cartes devaient être (re)produites à la main, ce qui est long et requiert une certaine expertise.

Remarque 29 (Des cartes à renouveler fréquemment)

Une fois terminée, une carte n'évolue plus. Elle représente une réalité à un instant donné. Par exemple,

- ▶ Les routes évoluent sur une échelle de temps de l'ordre de l'année ; certaines ferment, de nouvelles ouvrent, d'autres subissent des travaux qui en modifient le tracé, ...
- ▶ Le tracé des cours d'eau, des côtes et des falaises évoluent sur une échelle de temps de l'ordre de la décennie, à cause de l'érosion,
- ▶ Les frontières évoluent avec les conflits ; par exemple, le XX^e siècle a été le théâtre de nombreuses et fréquentes modifications des frontières au sein de l'Europe.

v – Prendre de la hauteur

Une carte dessinée sur une feuille est par nature un objet bidimensionnel. En général, on ne cherche qu'à y repérer les localisations des lieux et les tracés des routes, la notion d'altitude n'a que peu d'importance au sein des informations représentées. Les montagnes étaient représentées schématiquement ou artistiquement, et ne faisaient office que de repère visuel, sans contenir plus d'information.

Le besoin de répertorier avec précision et de manière continue l'altitude associée à certains lieux a émergé au XVIII^e siècle. Plutôt que l'altitude, il s'agissait en fait de profondeur : contrairement aux montagnes que l'on peut voir à l'œil nu, les fonds marins revêtent un danger particulier car ils sont invisibles aux yeux des marins. Naviguer dans des eaux trop peu profondes est souvent synonyme de naufrage : il était donc primordial de savoir quels endroits sont sûrs pour la navigation, avec une hauteur d'eau suffisante.

Comme beaucoup de choses, ce sont les avancées militaires qui ont fait évoluer les besoins. Pour calibrer ses tirs et ses itinéraires en montagne, l'armée a besoin de connaître l'altitude du lieu où elle progresse. À l'issue de la guerre, ces données sont devenues utiles pour l'ingénierie civile : connaître le dénivelé permet d'évaluer la possibilité de construire des routes, particulièrement des chemins de fer, prévoir des ponts et des tunnels, ...

Au début du XX^e siècle, les débuts de l'aviation marquent l'avènement d'une nouvelle méthode pour la cartographie : munis d'appareils photos, les avions survolent les zones à cartographier en prenant des clichés régulièrement. Ces clichés sont ensuite recollés ensemble tel un puzzle ⁽¹⁾, constituant ainsi une vue aérienne de la zone sur laquelle les cartographes peuvent s'appuyer pour en réaliser une carte, presque comme un calque. La même idée s'étendra au lancement de satellites permettant de capturer des images de la Terre à très grande échelle.

Mais l'échelle, justement, reste un problème : contrairement à ce qu'on pourrait penser, la résolution des appareils photo n'est pas (plus) un problème (aujourd'hui, un satellite situé à des centaines de kilomètres peut discerner un banc au bord de la route !). Le problème est la distorsion et le volume de données ; il s'agit d'un équilibre entre les déformations dues à la perspective et à la rotondité de la Terre. Pour échapper à ce phénomène, on pourrait prendre les clichés à basse altitude avec un faible angle de vue : mais ce faisant, le champ de vue est réduit, et il faut donc

(1). Cette analogie est abusive : en réalité, les clichés se recouvrent les uns les autres. Cela permet de corriger les déformations dues à l'appareil photo, d'autant plus accentuées que le relief en-dessous est important. Si vous disposez d'un appareil photo avec un mode grand angle, faites l'expérience ! En mode grand angle, la scène a un aspect un peu « arrondi », par rapport au mode normal.

prendre de très nombreux clichés pour couvrir toute la surface. Le problème de l'échelle est un problème logistique : si on veut avoir accès à toutes les informations simultanément, à l'échelle d'une ville et à l'échelle d'un pays par exemple, on aura besoin de plusieurs cartes, et donc de plusieurs prises de vue.

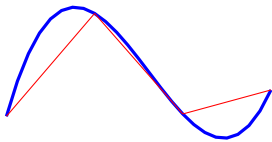
B – Les cartes numériques

i – Des points, des lignes et des polygones – GeoJSON

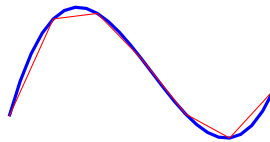
Une carte est, en tout et pour tout, un assemblage de formes géométriques. Quelle est la nature de ces éléments géométriques ?

L'objet géométrique le plus élémentaire est le point. Sur une carte, il repère un site unique : un bâtiment, un arbre, un banc, une fontaine... Un point est uniquement déterminé par ses coordonnées.

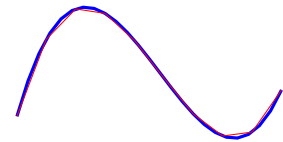
L'objet géométrique le plus simple après le point est la ligne. Parmi les lignes, le segment est le plus simple d'entre toutes : il est uniquement déterminé par les coordonnées de ses extrémités. On aimerait bien représenter les lignes courbes aussi simplement, mais c'est impossible. À ce sujet : vous êtes vous déjà demandé comment un ordinateur trace des lignes courbes ? Réponse : il ne le fait pas ! En réalité, il trace une ligne brisée : si les segments de la ligne brisée sont suffisamment courts, l'illusion est parfaite.



Approximation avec 3 segments

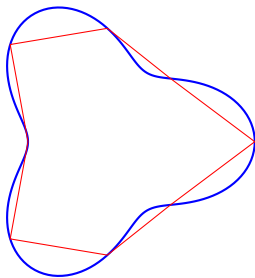


Approximation avec 6 segments

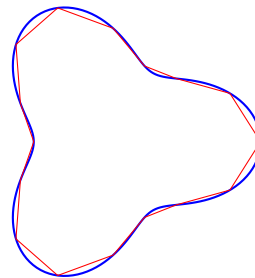


Approximation avec 9 segments

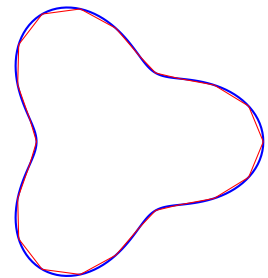
Après les lignes viennent les surfaces. Des carrés, des cercles, des triangles, ou même des formes plus exotiques. De toutes les surfaces, les plus simples sont les polygones : pour les connaître, il suffit de lister les coordonnées de leurs sommets. De même que pour les lignes courbes, on va approcher les surfaces par des polygones. Plus le polygone aura de côtés, meilleure sera l'approximation.



Approximation avec 8 segments



Approximation avec 16 segments



Approximation avec 24 segments

C'est suivant ce principe que l'on a décidé que les données géographiques seraient représentées par trois types de données : les Points, les (Multi)Lignes, et les (Multi)Polygones ⁽¹⁾.

(1). Le « Multi » permet d'englober les formes en plusieurs morceaux : par exemple, les lignes de RER à Paris se séparent en plusieurs branches, il est donc impossible de représenter tout le RER A à l'aide d'une seule ligne brisée. Autre exemple, la France ne peut pas être représentée par un seul polygone : on manquerait la Corse et les territoires d'Outre-mer. Pour autant, ces territoires non continentaux ne sont pas des pays à part ! Ils font partie d'un seul et même objet, l'objet « France ». Celui-ci est simplement constitué de plusieurs morceaux détachés les uns des autres.

Format GeoJSON**Information 30**

Le format *GeoJSON* est un format de données, issu du JSON. Il en partage la structure, et dispose de clés standardisées facilitant l'interopérabilité entre les différents producteurs de données et logiciels exploitant ces données.

Sa structure principale est composée de plusieurs clés donnant des renseignements généraux sur le jeu de données (comme des métadonnées), et d'une clé **"features"** contenant la liste des objets représentant les données géographiques.

Chaque objet géographique est repéré par **"type"** : **"Feature"**, et dispose de deux clés : **"properties"** et **"geometry"**.

La clé **"geometry"** renseigne le type de l'objet (Point, Ligne ou Polygone), ainsi que les coordonnées associées (du point, des nœuds de la ligne brisée, des sommets du polygone).

La clé **"properties"** contient les renseignements sur l'objet représenté : son nom, ce que c'est (une boulangerie, une école, un banc, ...), et bien d'autres informations.

Exemple 31

Consultez l'Annexe pour voir le code GeoJSON des trois objets suivants :

- Le lycée Joliot Curie de Nanterre est représenté par un Point. La clé **"properties"** renseigne le nom de l'établissement, s'il est public ou privé, sa ville, son adresse, sa date d'ouverture... Remarquez que l'on retrouve également une répétition de la latitude et de la longitude, et deux autres coordonnées nommées *x* et *y*. Cela correspond aux coordonnées cartésiennes (celles que l'on voit au collège) dans une certaine projection (précisée par la clé **"epsg"**). Plus de détails sur les projections dans l'Information 23.
- La ligne 3 bis du métro de Paris est représentée par une Ligne. On lit dans la clé **"geometry"**, sous clé **"coordinates"**, que la ligne brisée est définie par 4 points, qui sont en fait les 4 stations de la ligne. La clé **"properties"** renseigne le nom de la ligne, le type de ligne (**"Subway"** pour le métro, **"Train"** pour les TER et RER, **"Bus"** pour les bus, ...), la couleur identitaire de la ligne en hexadécimal, ...
- Le département de la Gironde est représenté par un polygone. La clé **"coordinates"**, sous-clé de **"geometry"**, renseigne les coordonnées de chaque sommet. La frontière de la Gironde étant complexe, cette liste est très longue ! On l'a coupée pour qu'elle rentre dans la page, en ne laissant que le premier et le dernier point ; remarquez que ceux-ci sont identiques : bien heureusement, pour fermer la figure, il faut que la fin rejoigne le début !
La clé **"properties"** ne contient que deux informations : le nom du département et son numéro.

Remarque 32 (Trop d'informations ? Le filtrage de données et les étiquettes)

Particulièrement sur l'exemple du lycée Joliot-Curie, qui pourtant a été élagué de quelques clés, on pourrait croire qu'il y a plus d'informations que nécessaire. Par exemple, si on connaît son code postal, quel intérêt de préciser le nom du département, de la région et de l'académie ? On pourrait retrouver toutes ces informations à l'aide du seul code postal !

Les informations listées dans la clé **"properties"** ont deux fonctions principales : fournir des étiquettes et permettre de filtrer les données.

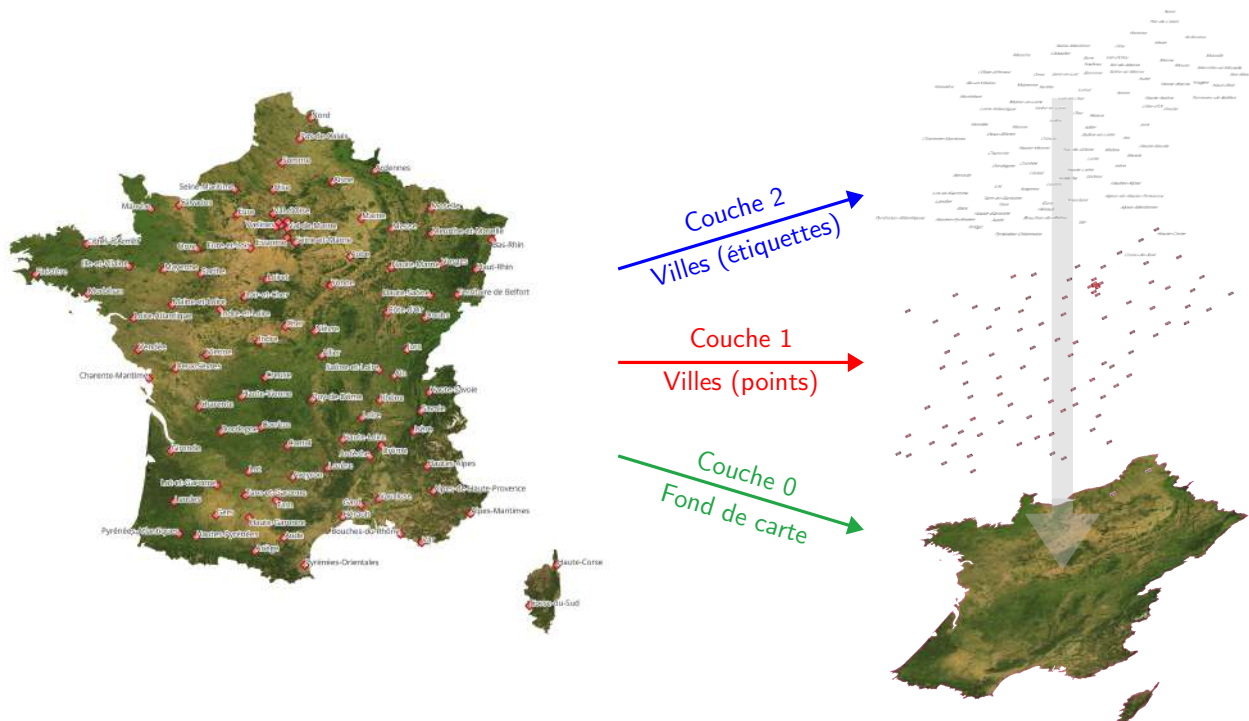
Une étiquette est le petit texte affiché sur la carte pour indiquer à quoi correspond un élément. Sans ça, impossible de savoir, par exemple, quel point correspond à quel lycée !

Lorsqu'on récupère le fichier GeoJSON des établissements scolaires de France, on en obtient la liste exhaustive. Mais si je veux faire une carte des lycées uniquement, il faut que je trouve un moyen de supprimer tous les collèges et toutes les écoles ! Et il va sans dire que l'on ne va pas supprimer ces milliers de données à la main. À la place on met en place un filtre : pour changer d'exemple, si je veux uniquement les établissements scolaires des Hauts-de-Seine, je dis à mon logiciel « conserve uniquement les données dont la clé **"code_département"** vaut **"092"** ». Autre exemple : si je souhaite dessiner une carte de France métropolitaine, je dis à mon logiciel « conserve uniquement les données dont la clé **"code"** n'est pas parmi les valeurs suivantes : (**"971"**, **"972"**, **"973"**, **"974"**, **"976"**) ».

ii – Le principe des couches

Le principe fondateur des cartes numériques est celui des *calques* : une carte est composée de plusieurs calques (plusieurs couches), superposées les unes aux autres. On peut classer les calques en trois catégories :

- Le fond de carte : c'est la couche la plus basse, c'est une image réelle ou schématique qui représente la zone cartographiée.
- Les symboles : les couches intermédiaires contiennent les polygones, les lignes et les points, représentant des régions, des routes, des villes, ...
- Les étiquettes : ce sont les informations textuelles qui donnent des renseignements sur les symboles (le nom de la ville, de la rue, de la région, ...). Les étiquettes sont placées sur la couche supérieure.



Les assistants de navigation rajoutent des couches *dynamiques*, qui évoluent en temps réel : un symbole pour afficher la position actuelle de l'utilisateur, les ralentissements en cours... Les couches dynamiques ne sont pas réservées aux cartes routières : autre exemple, les cartes météo, qui affichent la couverture nuageuse, les températures, ... qui évoluent en temps réel.

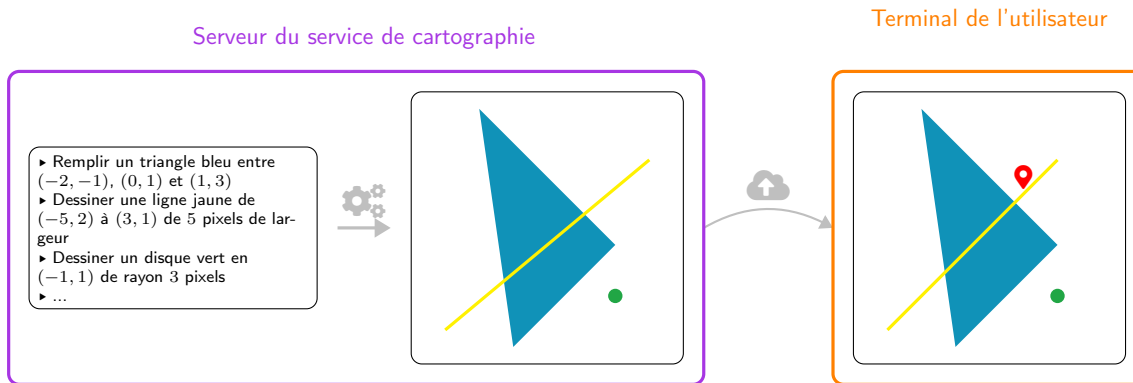
Le système de calques permet de rendre disponible un grand nombre de données sans surcharger la carte : il suffit de choisir les calques en fonction de nos besoins.

Exemple 33

Explorez les différents calques d'OpenStreetMap ou de Google Maps ! Cliquez ou survolez l'icône  pour faire apparaître la sélection disponible.

iii – La disparition du problème de l'échelle

À leurs débuts, les cartes numériques reposaient encore sur des images : le serveur avait à charge d'interpréter les données géographiques de sa base de données, les transformer en une image de carte, et envoyer cette image au terminal de l'utilisateur qui affiche ladite carte.



Chaîne de traitement des informations (ancienne)

Pour éviter d'envoyer une quantité démesurée de données, les applications de cartes numériques avaient recours au *tiling* (*tuilage*, en équivalent français) : l'image de la carte est découpée en petits carrés, qui sont ré-assemblés par l'utilisateur à la réception. Grâce à cette méthode, pas besoin de transmettre toute la carte d'un coup : il suffit de transmettre les tuiles qui entourent la position de l'utilisateur, et de lui envoyer les nouvelles au fur et à mesure qu'il avance.

Le tuilage permet de réduire la quantité de données à transmettre, mais le transfert d'images reste lourd par nature. Par ailleurs, cette méthode ne résout pas du tout le problème de l'échelle en pratique : si l'utilisateur souhaite zoomer dans la carte, à un moment les images des tuiles vont devenir floues ; il faut donc renvoyer de nouvelles tuiles adaptées à cette échelle. Et qui dit renvoyer des tuiles dit encore plus de données à transférer.

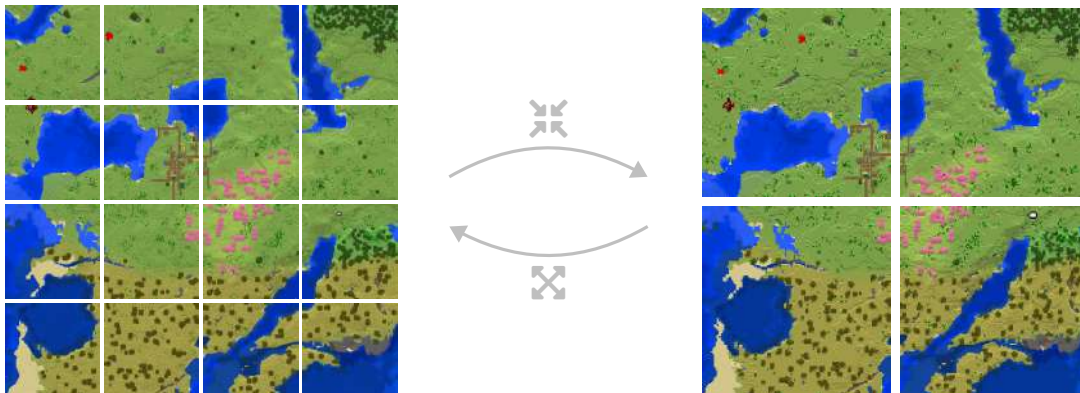


Illustration du principe de tuilage, notamment lors d'un zoom

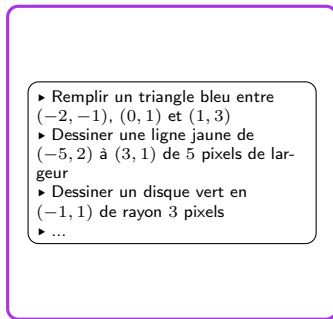
Sur l'illustration, en dézoomant (gauche vers droite), on a besoin de 4 fois moins de tuiles pour afficher la même zone, mais chaque tuile affiche les détails avec moins de précision.

Source des images : Minecraft

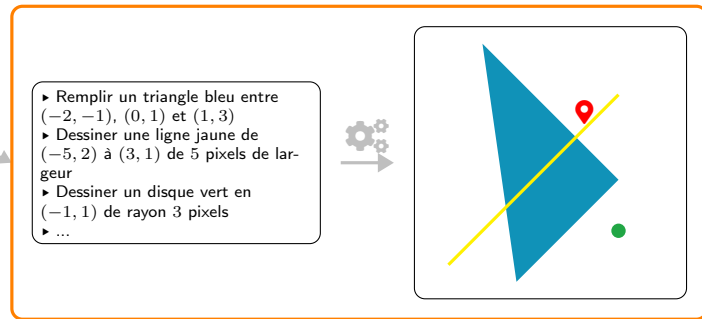
La personnalisation est également plus complexe : si l'on souhaite modifier les calques en place, il faut que le serveur recalculé la nouvelle carte modifiée, puis en transmette les tuiles à l'utilisateur.

Avec la progression de la puissance de calcul embarquée dans les terminaux des utilisateurs, les cartes numériques ont pu changer de paradigme pour économiser des données et rendre le processus plus efficace dans l'ensemble.

Serveur du service de cartographie



Terminal de l'utilisateur



Chaîne de traitement des informations (moderne)

Le serveur procède toujours au tuilage de la carte, mais au lieu de calculer lui-même des images prêtes à afficher, il envoie directement les données géographiques à l'utilisateur, dont le terminal s'occupe de composer l'image finale. Des données brutes étant bien plus légères que des images, les transferts sont d'autant plus rapides, et le goulot d'étranglement qu'était la vitesse de transfert a ainsi disparu.

C – Qui fait les cartes ?

i – Les cartes physiques

Dans l'Antiquité, les savants s'occupent de réaliser les cartes. Que ce soient des cartes astronomiques ou terrestres, la réalisation de ces cartes s'appuie sur des mesures géométriques (la position des étoiles dans la voûte céleste variant d'un lieu à l'autre, cela permet d'estimer notamment la latitude) et sur des récits d'itinéraires de marins et de marchands.

Au Moyen Âge, notamment concernant les portulans, ce sont des marins cartographes qui réalisent les cartes.

À la Renaissance, la cartographie retombe entre les mains des mathématiciens. Elle devient une affaire d'État et de prestige national, tant et si bien que les rois commandent des cartes à divers acteurs (par exemple, Tycho Brahe au Danemark et sa carte du ciel la plus précise jamais réalisée, commandée par le roi Frédéric II, ou les Cassini et leur carte de France, sous l'influence des rois Louis XIV et Louis XV).

À l'Époque contemporaine, la réalisation des cartes tombe sous la supervision de l'armée ; en France, le service de cartographie de l'armée est l'ancêtre de l'IGN (Institut national de l'information géographique et forestière), aujourd'hui chargé de répertorier les données et les représenter dans diverses cartes.

ii – Les cartes numériques

À l'ouverture débridée du GPS au grand public, des entreprises se sont fondées en ayant pour unique but de numériser la Terre : les images satellites ou aériennes fournissent un fond de carte, mais ça ne reste qu'une photo : il s'agit d'associer à chaque position une information (le tracé de chaque route, les lieux remarquables, ...)

Des entreprises comme Navteq et Tele Atlas ont donc déployé des hordes de véhicules équipés de GPS et arpentant chaque route, tout en annotant ces données GPS d'informations supplémentaires (cette rue-là s'appelle comme ça, celle-ci est à sens unique...). Ces entreprises ont ensuite vendu des données, par exemple, aux services de navigation GPS.

Ce travail de fourmi entrepris dans les années 1980-1990 a constitué une première base de données géographiques. Google rachètera en nombre les données de ces entreprises pour lancer Google Maps en 2005, un service mêlant imagerie satellite et données géographiques. En exploitant, par exemple, les données d'annuaires, Google a pu préciser ses cartes avec des lieux utiles au public : les boulangeries, les bureaux de Poste, ... En 2008, Google lance sa flotte

de véhicules « Street View », qui arpentent les rues à la manière des véhicules de Navteq et Tele Atlas, mais en capturant des données photographiques. Des outils d'intelligence artificielle ont permis d'exploiter cet immense volume de données, croisé avec la base préexistante, pour créer des cartes avec la précision que l'on connaît aujourd'hui.

En repassant régulièrement sur des lieux déjà capturés, Google Maps peut également tenir à jour ses cartes en incluant les différentes évolutions des territoires. Mais ce dispositif joue d'un équilibre entre temps et argent : soit on réduit les coûts en diminuant le nombre de véhicules arpenteurs, mais cela signifie des passages moins fréquents et donc des cartes plus susceptibles de rester obsolètes, soit on multiplie le nombre d'arpenteurs au détriment du coût de l'opération. Imaginez avec 7 milliards d'arpenteurs disponibles... c'est faisable, si tout le monde met la main à la pâte ! C'est sur cette idée que repose la *cartographie participative* : si chacun renseigne quelques informations à son échelle, l'incommensurable volume de contributeurs a vite fait de fournir toutes les données dont on a besoin, et même de les tenir à jour ! ⁽¹⁾

C'est en accord avec cette philosophie que le projet OpenStreetMap a été lancé en 2004. Chacun peut ajouter ou mettre à jour des données sur une carte partagée. Les principaux avantages de cette méthode sont le coût presque inexistant (les informations sont ajoutées bénévolement), la réactivité (le grand nombre de contributeurs potentiels permet de mettre à jour les données rapidement), et la protection / l'absence de contrôle des données (on peut ajouter des données anonymement, et aucune entité ne peut censurer les données partagées ; les données sont libres et ouvertes (*Open Data*)) ⁽²⁾. En contrepartie, des erreurs peuvent se glisser dans les informations, pas nécessairement par malveillance : les données saisies peuvent être inexactes ou erronées.

La pratique consistant à fournir volontairement des informations à une base de données commune s'appelle le *crowdsourcing actif* (*production participative active* en équivalent français). Mais toutes les données exploitées par les géants de la cartographie ne sont pas fournies volontairement ⁽³⁾. Google Maps ou Waze affichent en temps réel les bouchons et ralentissements : ils n'attendent pas que les utilisateurs signalent ces perturbations, ils extrapolent ces informations à partir du flux de données qui leur parvient. En effet, si tous les utilisateurs roulent à la même vitesse, inférieure à la limitation en vigueur à cet endroit, c'est qu'il est probable qu'un ralentissement soit en cours. De même, Google peut afficher la fréquentation habituelle des magasins, en fonction de l'heure et du jour : pour ce faire, il détecte les utilisateurs présents dans ledit magasin, quel jour et à quelle heure, pour combien de temps.

C'est ce qu'on appelle le *crowdsourcing passif* (ou *collecte de données passive* en équivalent français). Sans même que nous ayons à ouvrir l'application ou à cliquer sur un bouton, nos smartphones nous désignent comme des capteurs cartographiques en temps réel. Cette collecte invisible pose toutefois la question de la vie privée et du modèle économique de ces outils. Si des services aussi complexes que Google Maps ou Waze sont gratuits, c'est parce que nos données de déplacement ont une valeur immense. Prises isolément, ces données semblent anodines ; mais accumulées jour après jour, elles permettent de reconstituer nos habitudes, nos lieux de travail, nos activités de loisirs, et même nos visites médicales.

Au-delà de compromettre notre vie privée ou d'influencer notre consommation au travers de publicités ciblées, les données de localisation peuvent constituer un enjeu de sécurité. Un exemple parlant est celui de l'application Strava, qui enregistre et publie les performances sportives de ses utilisateurs. En analysant l'historique des comportements de certains utilisateurs, Le Monde ⁽⁴⁾ a pu identifier, malgré l'anonymat, l'un des gardes du corps du Président. Une fois cette personne repérée, on peut en déduire la position exacte du Président à partir de celle de son garde du corps.

(1). C'est le même principe que Wikipédia !

(2). Absence de contrôle ne signifie pas absence de vérification ! Comme pour Wikipédia, des bénévoles ou des programmes automatiques effectuent un travail de modération, mais le principal moteur de correction est la *validation par les pairs* : si tout le monde est d'accord sur une donnée, il est très probable qu'elle soit vraie.

(3). Enfin, pourrait-on dire, tout le monde a lu les Conditions d'utilisation, et consent explicitement à ce partage, n'est-ce pas ?

(4). Source : [Comment suivre le président Macron à la trace | #StravaLeaks Ep 1/2](#) sur YouTube.

iii – Et si c'était vous ?

Activité 34 (TP – découverte d'OpenStreetMap)

Rendez-vous sur OpenStreetMap, et recherchez la commune de Luzarches, puis parcourez la carte. Repérez notamment le golf.

- 1► Quelles informations sont visibles directement sur la carte ?
- 2► Quels types d'objets sont représentés ? On pourra distinguer les routes, bâtiments, espaces verts, équipements sportifs, commerces, etc.
- 3► Effectuez un clic droit sur un objet, puis choisissez « Interroger les objets ». À gauche, cliquez sur un objet. Quelles propriétés s'affichent ?
- 4► Relevez au moins trois propriétés associées à l'objet choisi. Tentez de les traduire en français. (si l'objet n'a pas 3 propriétés, choisissez-en un autre)

Activité 35 (TP – Filtrer les données)

(Ce TP fait suite à l'Activité 34.)

Rendez-vous sur <https://overpass-turbo.eu/>, un outil qui permet d'exploiter les objets disponibles dans OpenStreetMap.

Partie A : prise en main

- 0► Dans Assistant, indiquer `name = "McDonald's" in Ile-de-France` puis exécuter.

La carte des McDonald's d'Île-de-France apparaît !

- 1► Étudier la carte à Paris. Les McDonald's sont-ils répartis régulièrement dans Paris ?
- 2► Dans quels arrondissements semblent-ils les plus nombreux ?
- 3► Sont-ils plutôt situés près : des grandes avenues ? des gares ? des centres commerciaux ? des quartiers touristiques ? des zones de bureaux ?
- 4► Quelles conclusions peut-on en tirer sur les choix de placement des McDonald's ?

Partie B : exploration

Voici d'autres exemples de recherche que vous pouvez faire (avec l'assistant). Sont-elles bien réparties sur le territoire communal ?

<code>shop = bakery in Paris</code>	Les boulangeries à Paris
<code>amenity = bicycle_rental in Paris</code>	Les Vélib à Paris
<code>amenity = school in "Val-d'Oise"</code>	Les écoles du Val-d'Oise
<code>amenity = hospital in "Ile-de-France"</code>	Les hôpitaux d'Île-de-France
<code>leisure = park in Paris</code>	Les parcs et jardins parisiens
<code>cuisine = kebab in Paris</code>	Les kebabs à Paris

Activité 36 (TP – créez votre propre carte)

(Ce TP fait suite à l'Activité 35.)

- 1► Une fois les résultats affichés dans Overpass Turbo, nous allons les récupérer sous forme de fichier.
 - a► Cliquez sur **Exporter**.
 - b► Dans la partie **Données**, choisissez **GeoJSON**.
 - c► Téléchargez le fichier.

- d► Renommez le fichier avec un nom clair, par exemple : `mcdonalds_ile_de_france.geojson`
- 2► Nous allons maintenant créer notre propre carte. Rendez-vous sur umap.openstreetmap.fr
 - a► Cliquez sur **Créer une carte**.
 - b► Donnez un titre à la carte.
 - c► Cliquez sur le bouton d'import de données.
 - d► Choisissez le fichier GeoJSON téléchargé depuis Overpass Turbo.
 - e► Importez les données.
 - f► Les points doivent apparaître sur le fond de carte.
- 3► (*Facultatif*) Modifiez l'apparence de votre carte. Pour cela, cherchez le bouton « calques », modifiez, puis « Propriétés de la forme ». Vous pouvez choisir une icône adaptée au thème.
- 4► (*Pour aller (plus) loin*) Vous pouvez importer plusieurs fichiers GeoJSON dans la même carte uMap. Par exemple, mettez sur la même carte les kebabs et les gares (railway = station **in** Paris).
Chaque jeu de données doit être placé dans un nouveau calque. Que remarque-t-on ?
- 5► Nous allons comparer avec la densité de population. Dans une nouvelle fenêtre, allez sur « Geoportail » et cherchez le fond de carte **Économie > Démographie > Densité**. Que peut-on remarquer en comparant la densité et les écoles du Val-d'Oise ?

6 – Annexes

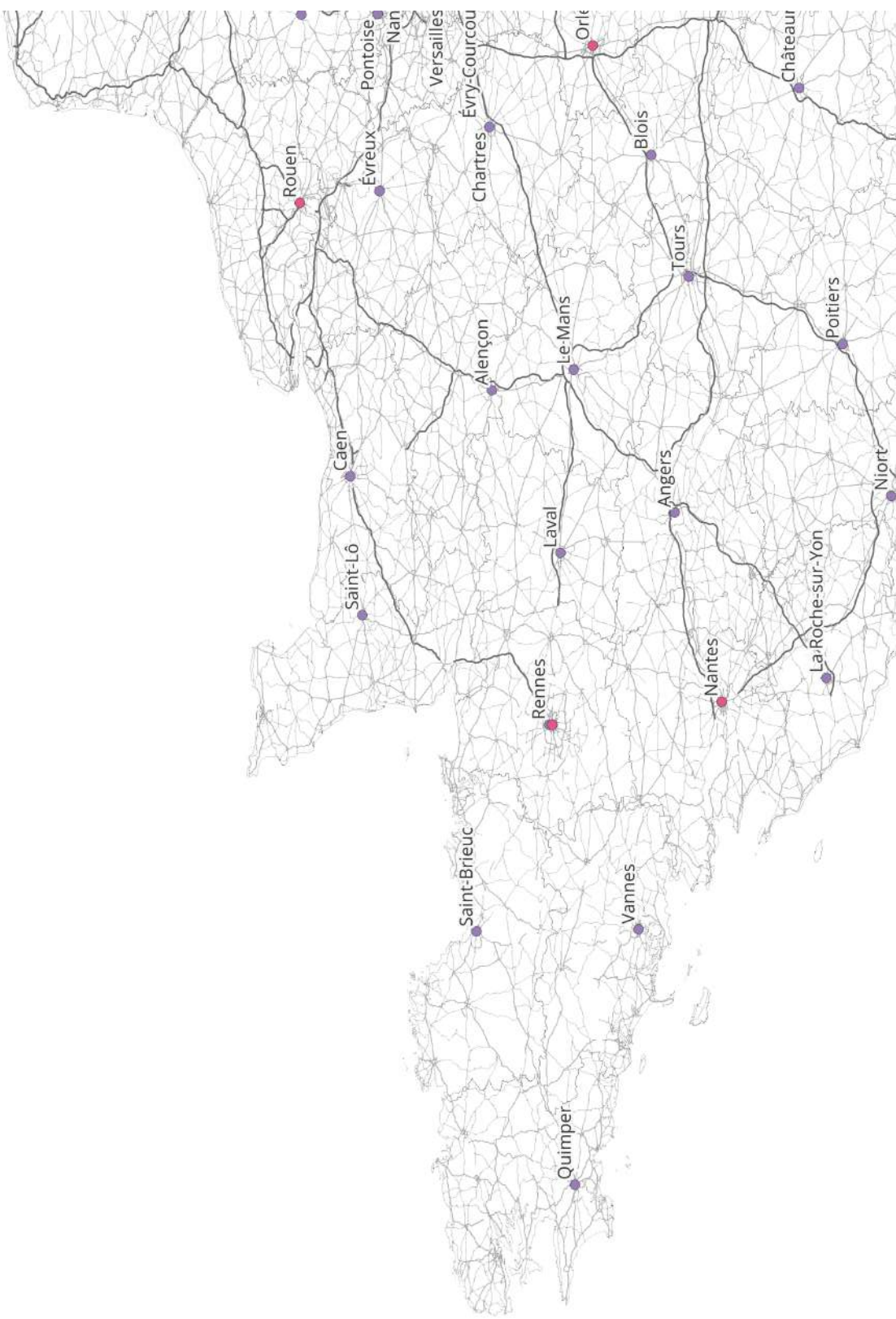
Retrouvez les cartes complètes et en quatre morceaux à l'adresse [.....]

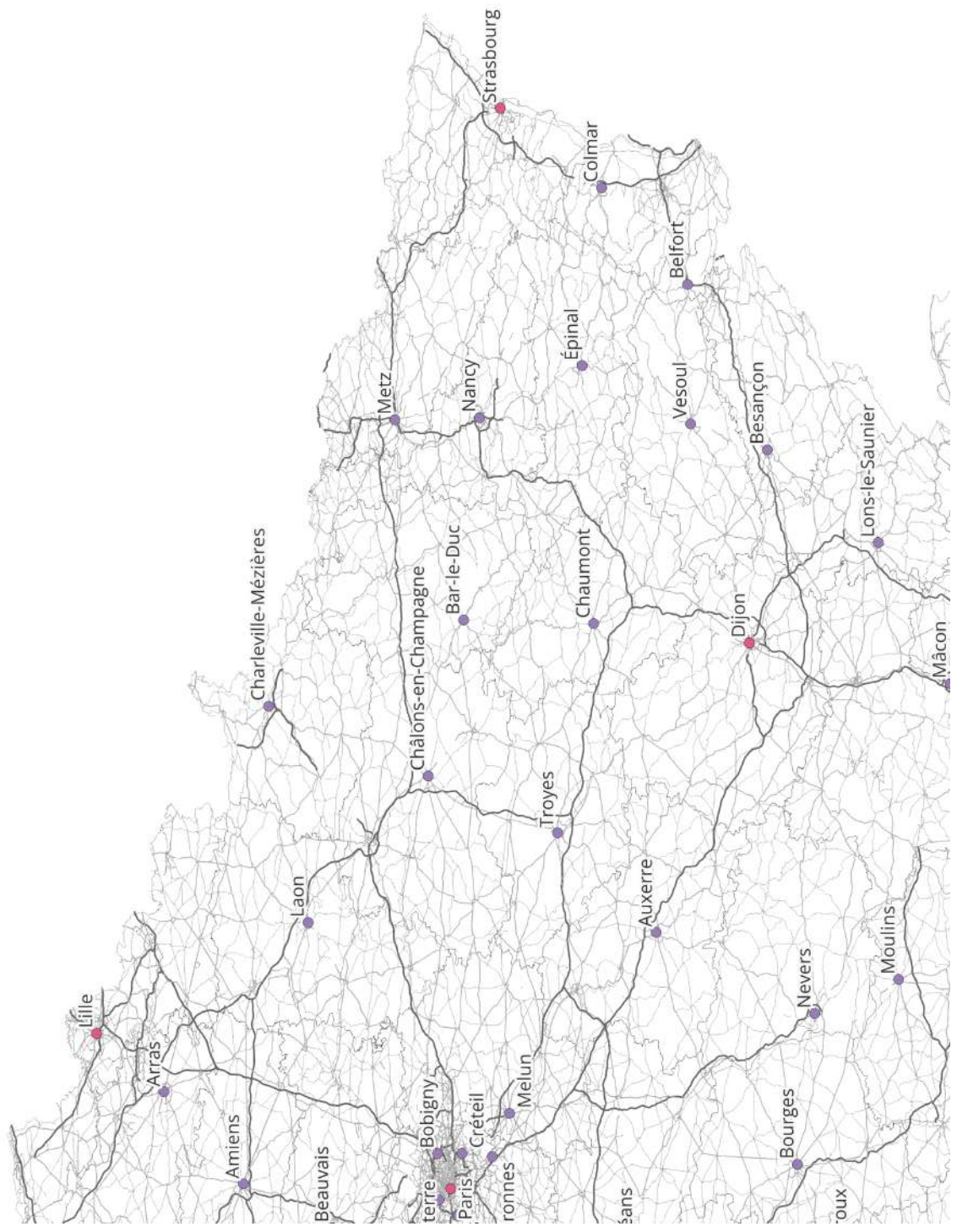
Si vous n'avez pas le matériel physique nécessaire pour mener l'activité, vous pouvez la faire sur GeoGebra à l'adresse suivante [.....] ou en local en suivant le protocole suivant (décrit pour GeoGebra 5) :

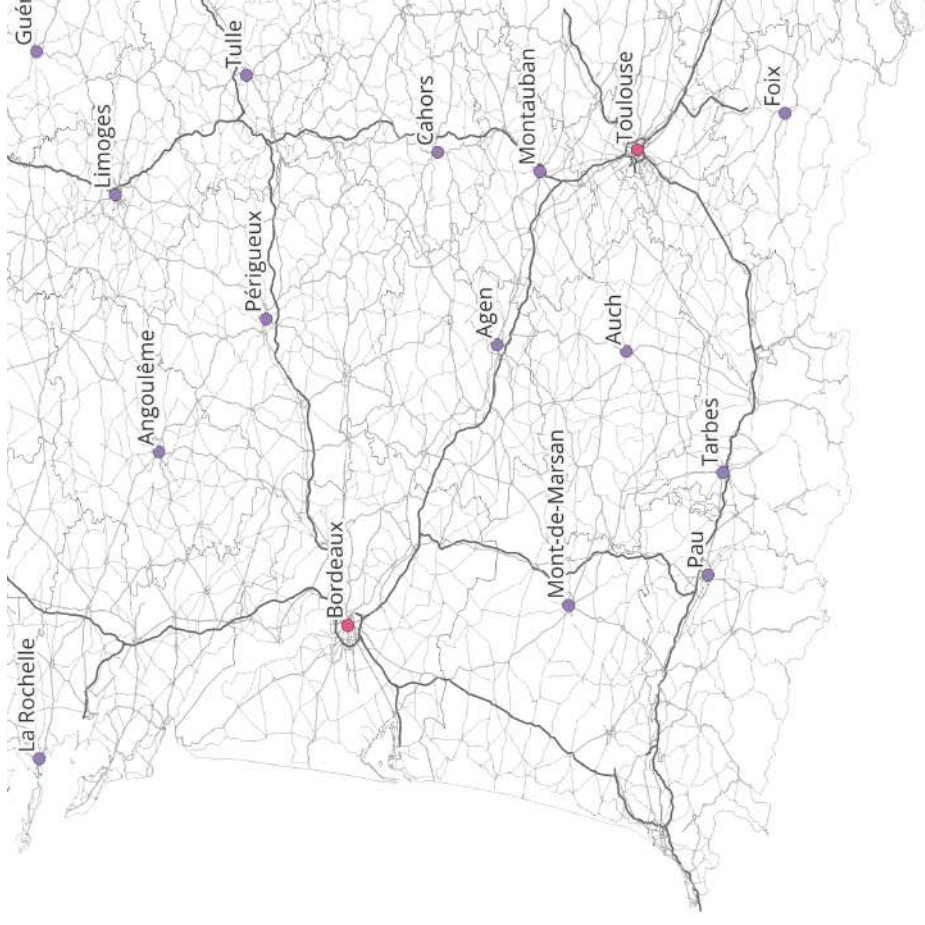
- 1► Masquez les axes et le quadrillage.
- 2► Importez le fond de carte dans GeoGebra.
- 3► Allez dans les propriétés de l'image importée.
- 4► Cochez la case « Image d'arrière-plan ».

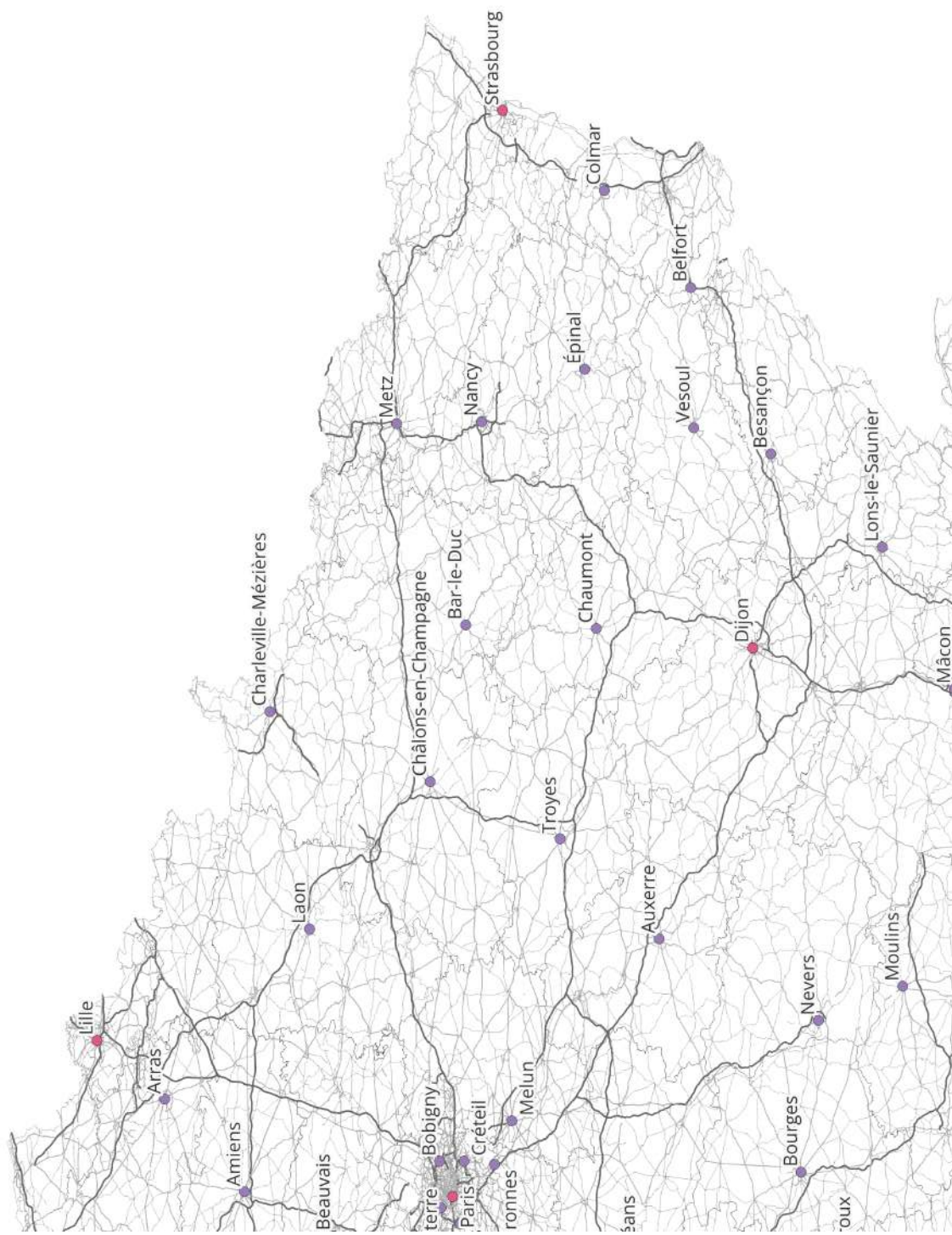
Sources des images en annexe :

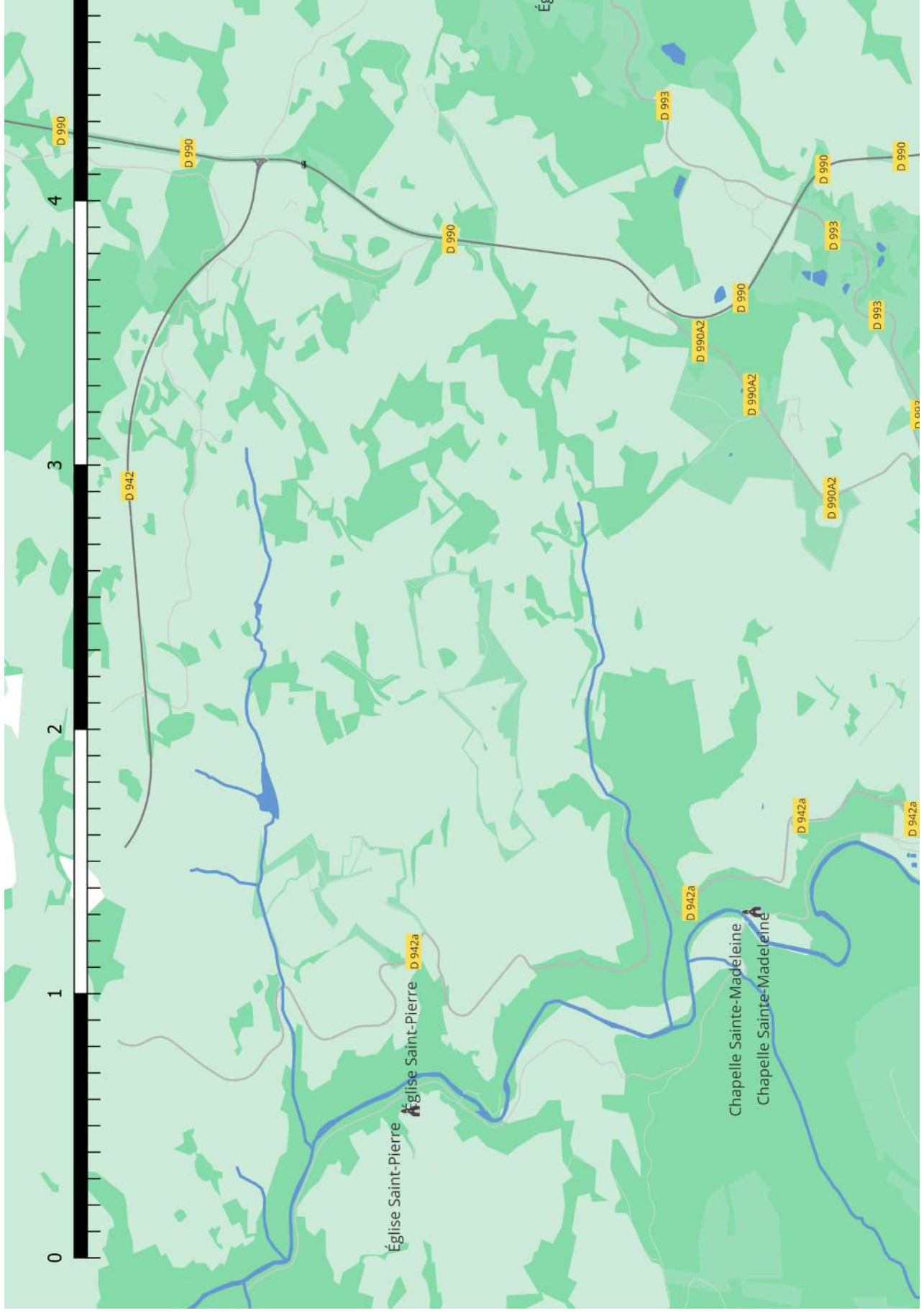
- Carte des alentours d'Aubusson : captures d'écran de Google Maps
- Planisphère : utilise la Blue Marble de la NASA

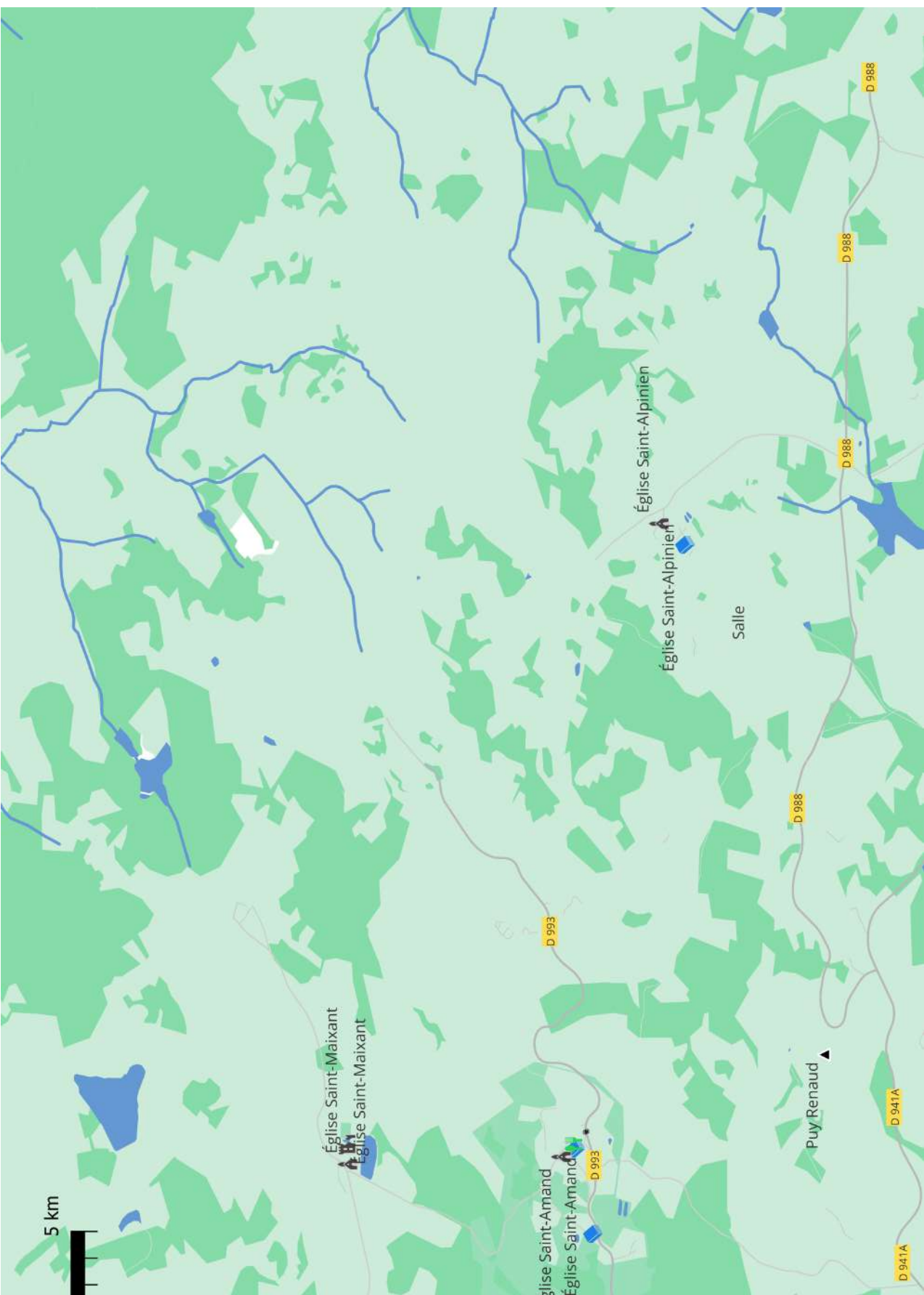


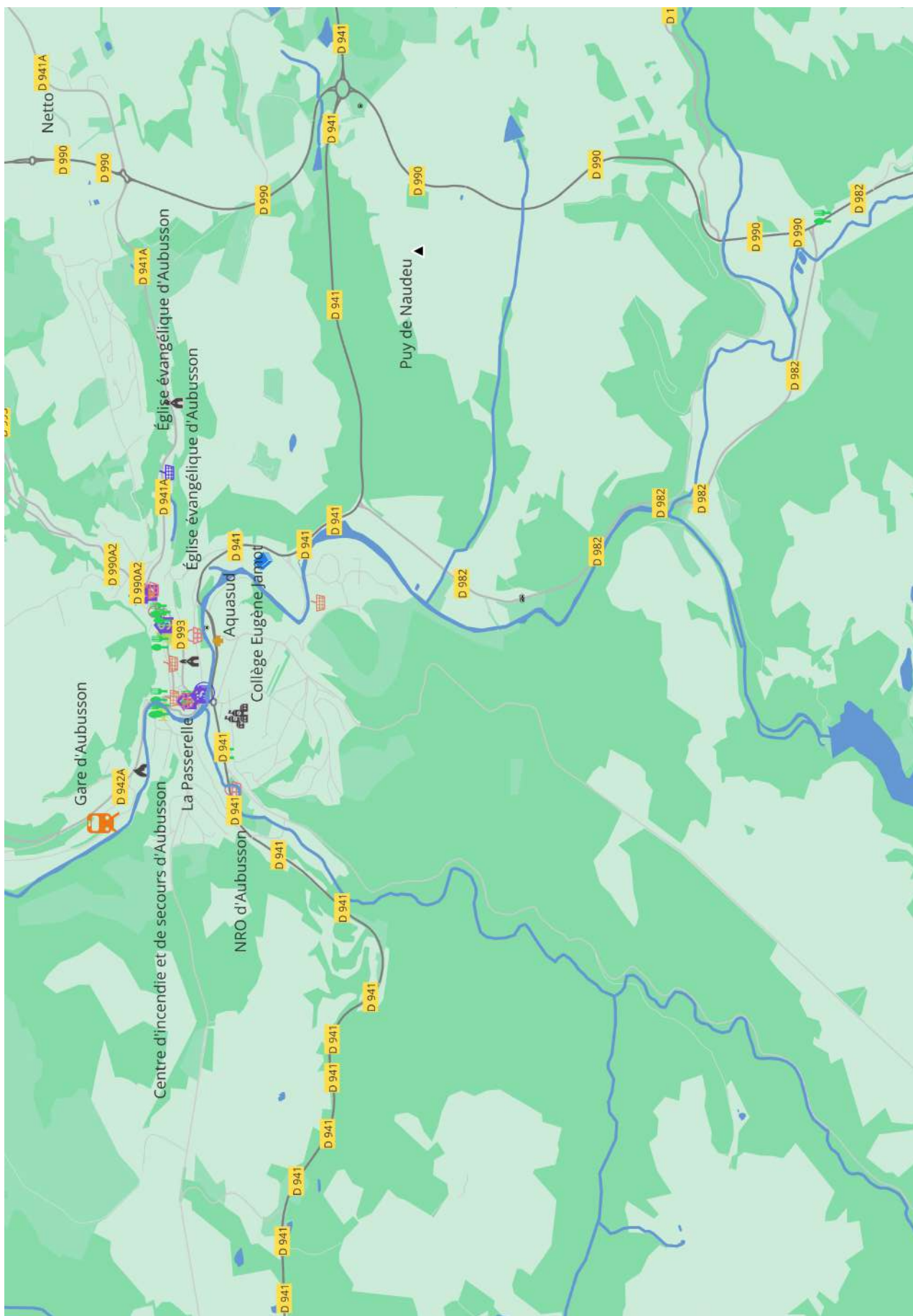


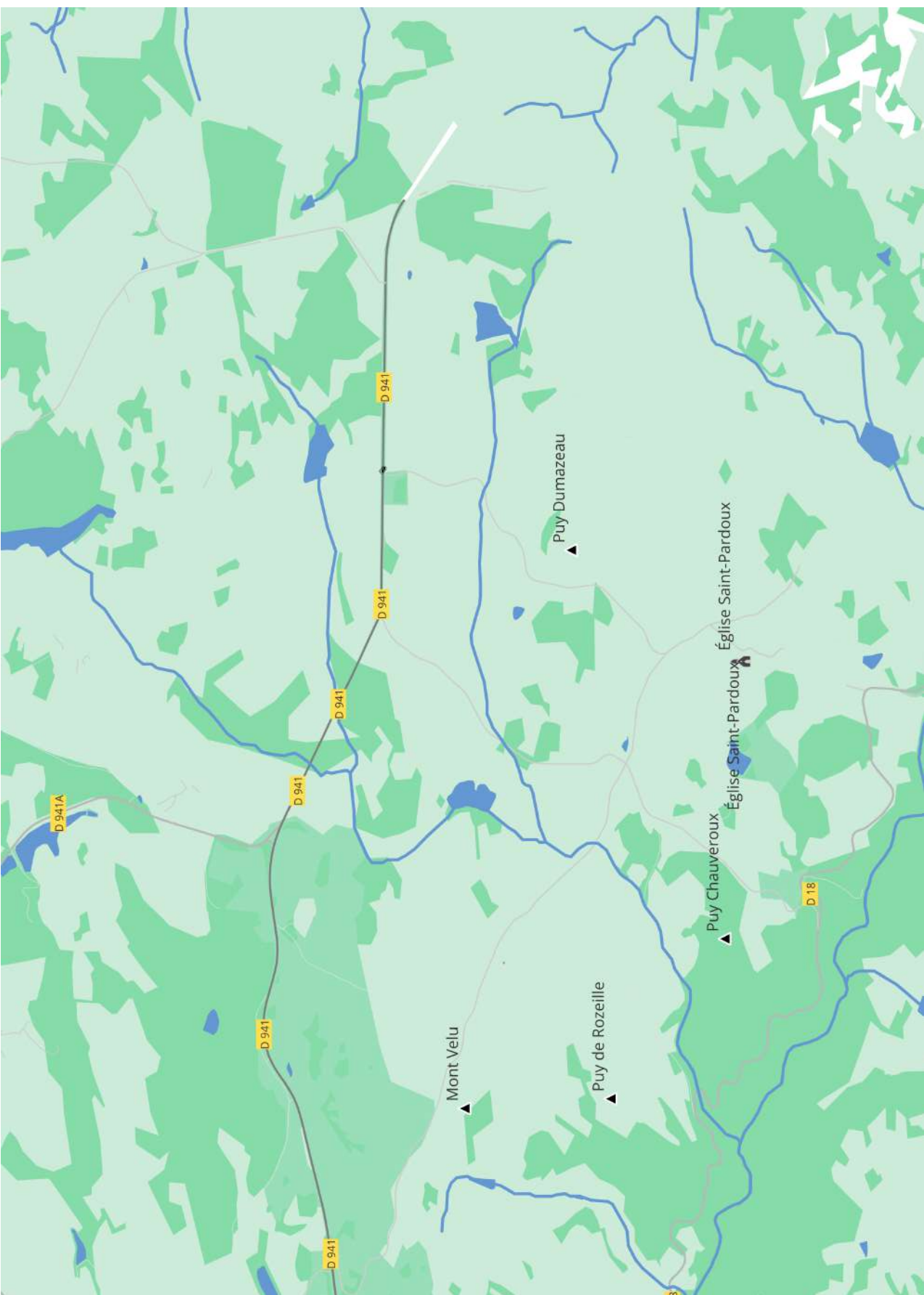


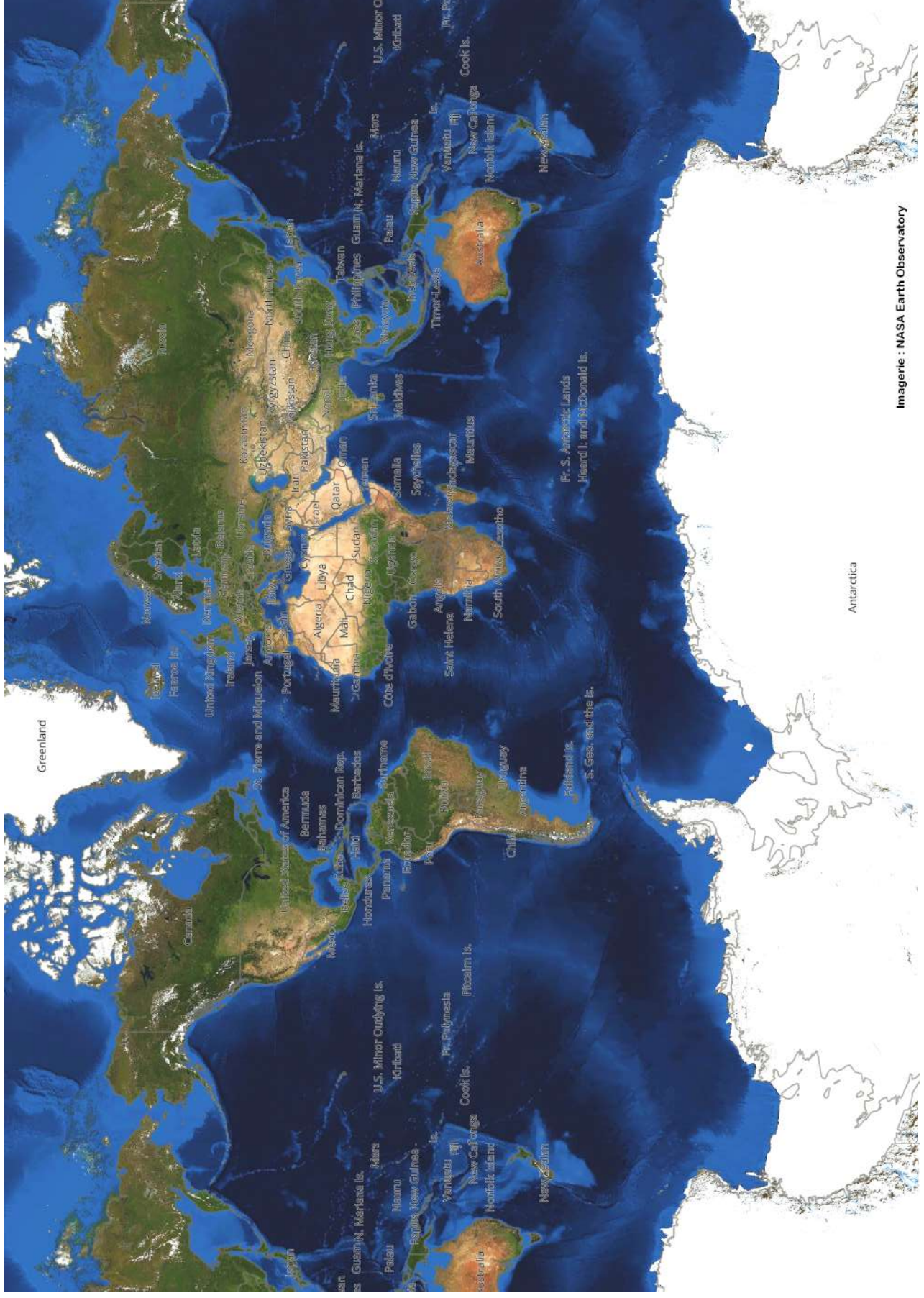










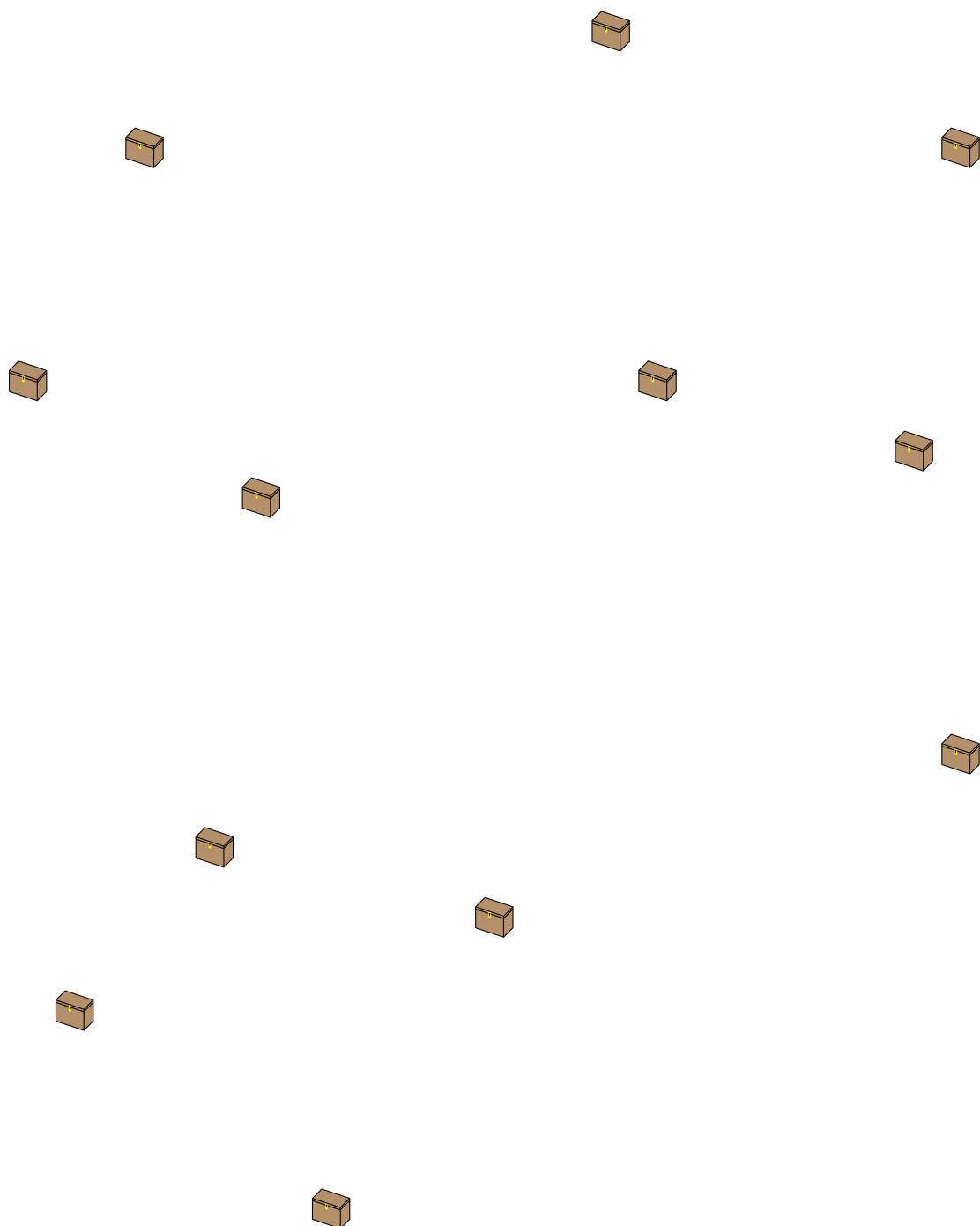


Imagerie : NASA Earth Observatory

1 graduation = 1m



Annexe de l'activité 15



Extrait du code GeoJSON d'un point (le lycée Joliot-Curie de Nanterre)

```
{
  "type": "FeatureCollection",
  "name": "lycees",
  "crs": {
    "type": "name",
    "properties": {
      "name": "urn:ogc:def:crs:OGC:1.3:CRS84"
    }
  },
  "features": [
    ...
    {
      "type": "Feature",
      "properties": {
        "numero_uai": "0920141D",
        "appellation_officielle": "Lycée polyvalent Joliot-Curie",
        "secteur_public_prive_libe": "Public",
        "adresse_uai": "92 avenue Frédéric et Irene Joliot Cur",
        "code_postal_uai": "92014",
        "libelle_commune": "Nanterre",
        "coordonnee_x": 641896.8,
        "coordonnee_y": 6865954.2,
        "epsg": "EPSG:2154",
        "latitude": 48.890974840699883,
        "longitude": 2.2075839125947363,
        "code_departement": "092",
        "code_region": "11",
        "code_academie": "25",
        "code_commune": "92050",
        "libelle_departement": "Hauts-de-Seine",
        "libelle_region": "Ile-de-France",
        "libelle_academie": "Versailles",
        "code_ministere": "06",
        "libelle_ministere": "MINISTERE DE L'EDUCATION NATIONALE",
        "date_ouverture": "1965-05-01T00:00:00",
      },
      "geometry": {
        "type": "Point",
        "coordinates": [
          2.207583912594736,
          48.890974840699883
        ]
      }
    },
    ...
  ]
}
```

Extrait du code GeoJSON d'une ligne (la ligne de métro 3 bis à Paris)

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "coordinates": [
          [
            [
              2.398537305704572,
              48.865032564464
            ],
            [
              2.40160636134298,
              48.86842046799948
            ],
            [
              2.4044981831688688,
              48.87184337222297
            ],
            [
              2.4070619733807948,
              48.87656859902798
            ]
          ]
        ],
        "type": "LineString"
      },
      "properties": {
        "route_id": "IDFM:C01386",
        "route_short_name": "3B",
        "route_long_name": "3B",
        "route_type": "Subway",
        "route_color": "6EC4E8",
        "route_url": null,
        "id_ilico": "C01386",
        "operatorname": "RATP",
        "networkname": null,
        "url": "https:\\\\www.ratp.fr\\sites\\default\\files\\lines-assets\\plan-
          de-ligne\\metro\\plan-de-ligne_metro_ligne-3b.pdf",
        "long_name_first": "3",
        "geo_point_2d": {
          "lon": 2.4030809868334333,
          "lat": 48.87060226732235
        }
      }
    }
  ]
}
```

Extrait du code GeoJSON d'un polygone (département de la Gironde)

```
{
  "type": "FeatureCollection",
  "features": [
    ...
    {
      "type": "Feature",
      "geometry": {
        "type": "Polygon",
        "coordinates": [
          [
            [
              -0.7188015186962935,
              45.327417794908314
            ],
            ...
            [
              -0.7188015186962935,
              45.327417794908314
            ]
          ]
        ]
      },
      "properties": {
        "code": "33",
        "nom": "Gironde"
      }
    },
    ...
  ]
}
```

8 Pourquoi (se) repérer ?

Table des matières

1	Repérer les lieux	135
2	Repérer les données	135

1 – Repérer les lieux

2 – Repérer les données

Partie 1 : L'Histoire — De la survie à la conquête du monde Repérer les ressources et le territoire :

Les premières cartes de l'humanité (ex : les peintures de la grotte de Lascaux représentant les constellations, ou la carte de Bedolina gravée dans la roche) : un besoin vital de repérer le gibier, l'eau et les voies de passage.

Naviguer et explorer (Se repérer sans repères) :

Comment faire quand il n'y a pas de paysage (en pleine mer) ? L'orientation grâce aux astres (Étoile Polaire, Soleil), puis l'invention de la boussole et des portulans.

La grande crise scientifique de la Longitude (XVIIIe siècle) : Pourquoi ne pas savoir où l'on est en mer coulait des flottes entières.

Cadastrer et administrer :

La carte comme outil de pouvoir : repérer pour taxer, faire la guerre ou tracer des frontières (ex : les cartes de Cassini sous Louis XIV).

Partie 2 : Les enjeux sociétaux — Quand le numérique transforme le besoin De la carte collective à l'espace centré sur « Moi » :

Avant, se repérer demandait de lire une carte orientée au Nord et d'analyser le paysage autour de soi. Aujourd'hui, la carte pivote avec nous et place l'individu au centre absolu de l'univers virtuel.

La mutation des mobilités : de la quête à l'optimisation :

On ne cherche plus seulement son chemin, on cherche le chemin optimal instantané (éviter les bouchons, trouver le Fast-Food le plus proche ouvert H24). Le repérage est devenu un outil de consommation et de gestion du temps.

La souveraineté et le contrôle des populations :

Qui a le droit de repérer quoi ? L'enjeu stratégique des cartes (pourquoi Google floute les bases militaires, pourquoi les États veulent leur propre système souverain comme Galileo).

Le marché de la géolocalisation :

Aujourd'hui, ce ne sont plus seulement les humains qui se repèrent : ce sont les entreprises qui nous repèrent. La position géographique est devenue la métadonnée commerciale ultime.

9 Les algorithmes et les positions

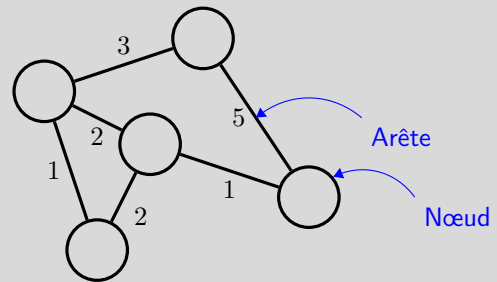
1 – Les algorithmes d'itinéraire

A – Notion de graphe

Graphe

Définition 1

Un *graphe* est une manière de représenter de manière schématique les relations entre différentes données. Les données sont représentés sur les *nœuds* du graphe, et les connexions entre les nœuds sont appelées des *arêtes*. Sur chaque arête, on peut ajouter un nombre appelé *pondération*, qui peut représenter une distance, un temps, un coût, ... selon la situation. On parle alors de *graphe pondéré*.

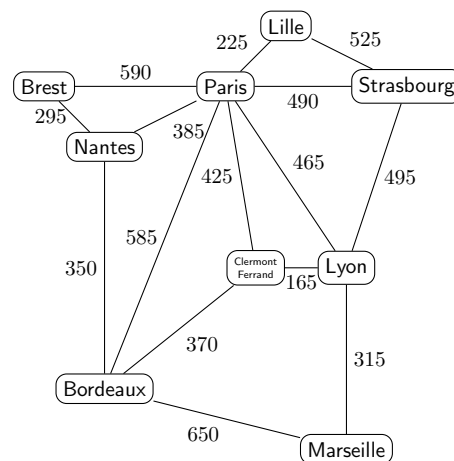
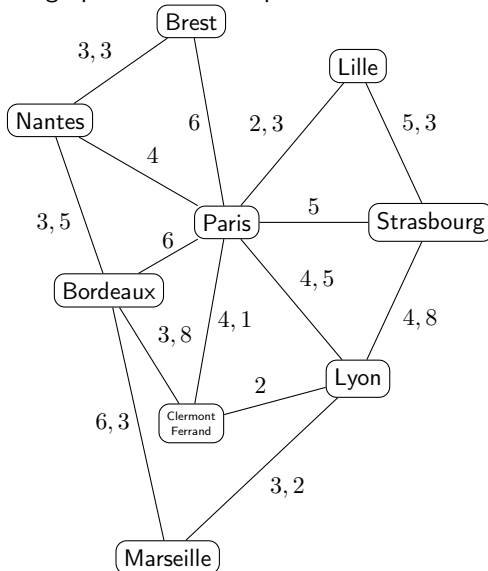


Exemple 2

Le réseau routier peut-être représenté par un graphe, dont les villes sont les nœuds, et les routes sont les arêtes. Les pondération sur les arêtes peuvent, par exemple, représenter la distance en km entre deux villes voisines. Dans la première représentation, les nœuds ont été placés de manière à ce que la longueur d'une arête représente approximativement la distance de la pondération, comme une vraie carte à l'échelle.

Dans la seconde, les pondérations sont les temps de trajet en voiture, en heures. Cette fois-ci, on a arrangé le graphe de sorte à équilibrer les longueurs des arêtes, et renforcer la position centrale de Paris : la longueur d'une arête n'est donc plus représentative de sa pondération.

Un graphe étant une représentation schématique, il n'a pas besoin d'être fidèle à des positions réelles.



Algorithmes d'itinéraire

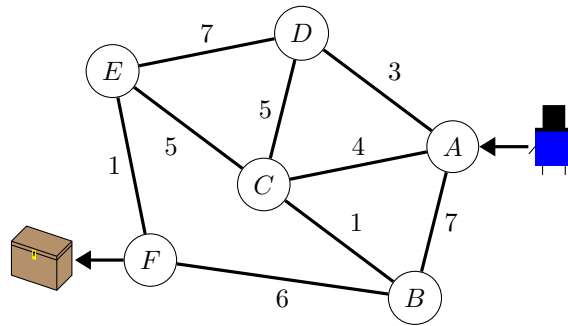
Information 3

Les *algorithmes* d'itinéraires servent à trouver le plus court chemin entre deux nœuds d'un graphe pondéré. Il existe de nombreux algorithmes d'itinéraire, mais le plus célèbre d'entre eux est sûrement l'*algorithme de Dijkstra*, pour sa simplicité et sa relative efficacité.

Expliquons le fonctionnement de l'algorithme de Dijkstra au travers d'un exemple illustré.

Carole part explorer un donjon. Elle entre par la salle *A* et a pour objectif d'atteindre le trésor dans la salle *F*. Mais comme tout trésor qui se respecte, celui-ci est piégé : une fois retiré de son piédestal, un mécanisme s'enclenche et une horde de monstres surgit de la salle au trésor, et envahit tous les couloirs. Carole souhaite donc faire preuve de prudence : à tout moment de son exploration, elle doit être en mesure de s'échapper du donjon le plus rapidement possible, c'est-à-dire en suivant le chemin le plus court. Autrement, elle risque d'être prise en tenaille par des monstres qui auraient progressé plus rapidement dans le labyrinthe.

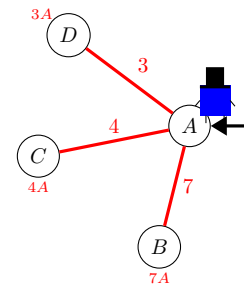
Le graphe ci-contre représente le plan du donjon, chaque nœud est une salle du donjon, et les pondérations sur les arêtes représentent le temps de marche entre deux salles (en heures). Carole ne dispose pas de ce plan : elle ne sait pas quelle salle est reliée à quelle autre, quelle distance les sépare, ni où se trouve la salle au trésor ; elle va devoir le découvrir au fur et à mesure de son exploration. Pour cela, Carole tient un journal d'exploration.



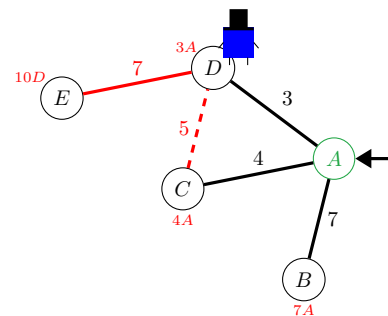
Carole opte pour la stratégie d'exploration suivante :

- 1► Elle établit son camp dans la première salle ;
- 2► Elle explore les couloirs reliés à cette salle, en notant le temps de trajet et la salle de laquelle elle vient ;
- 3► Elle établit un nouveau camp dans la salle la plus proche de *A* parmi les salles qu'elle a précédemment découvert mais pas encore exploré ;
- 4► Elle répète les étapes précédentes à partir de la nouvelle salle.

Lors du premier raid, Carole établit son camp dans la salle *A*. Elle remarque trois couloirs, qu'elle explore. Le premier mène à la salle *B* : elle est à 7h de la salle *A*, et son camp est établi en *A*, elle note donc 7A. De même, elle note 4A à côté de la salle *C* et 3A à côté de la salle *D*.



Selon ses trouvailles, la salle la plus proche de *A* est la *D* : c'est donc ici qu'elle établit son deuxième camp. D'ici, elle remarque trois couloirs, qu'elle explore. Le premier mène évidemment à salle *A* d'où elle vient, inutile de le parcourir à nouveau. Vers le Sud, Carole découvre un couloir qui mène à la salle *C*, qu'elle a déjà vu : elle a déjà mis 3h pour venir en *D*, et ce couloir lui rajoute 5h de trajet, soit 8h au total depuis le départ. Comme son camp actuel est en *D*, Carole s'apprête à écrire « 8D », mais elle voit qu'elle a déjà écrit « 4A » : elle en conclut que passer par *D* est plus long pour aller en *C*, elle garde donc 4A comme annotation.



Comme le couloir entre *C* et *D* est inutile, elle le trace en pointillés ; ça ne lui sera pas utile pour l'exploration, mais ça rend la carte plus lisible. Le troisième couloir mène en *E*, une nouvelle salle : en tenant compte des 3h pour aller

de A et D , et des 7h pour aller de D et E , elle note $10D$.

Carole a maintenant connaissance des salles A à E . Elle a déjà établi un camp et exploré les salles A et D , inutile d'y retourner. Cela lui laisse donc les salles B , C et E : d'après ses notes, la plus proche de A est la salle C (4h contre 7h et 10h) : c'est donc ici qu'elle établit son troisième camp.

Elle y retrouve les deux couloirs menant aux salles A et D qu'elle connaît déjà. Parmi les deux nouveaux couloirs, l'un mène vers la salle B en 1h. La mention « $4A$ » associée à la salle C où elle se trouve signifie que le trajet de A à B lui a pris 5h en passant par C : c'est moins que les 7h précédemment trouvées !

Elle met donc à jour ses notes avec la mention $5C$. De même, le trajet vers E en passant par C est plus court qu'en passant par D , elle remplace donc le $10D$ par $9C$.

Comme précédemment, elle indique par des pointillés les chemins qui ne sont pas rentables.

Des salles encore non explorées, la B est plus proche de l'échappatoire que la E , c'est donc là-bas qu'elle établit son quatrième camp. Elle y retrouve les couloirs vers les salles A et C , déjà explorés. Un troisième couloir attire son attention... après 6h de marche, elle tombe sur la salle au trésor ! La mention « $5C$ » marquée au camp B permet à Carole de conclure que, en suivant le chemin passant par B , il lui faut au total 11h pour relier l'entrée et la salle au trésor.

Mais Carole est prudente : ce n'est pas parce qu'elle a atteint la salle F en passant par B qu'il n'existe pas de route plus rapide. En effet, elle n'a pas encore exploré la salle E , qui pourrait aussi être reliée à la F . Elle établit donc un cinquième camp là-bas ; et bingo ! Un couloir relie les salles E à F en 1h. La mention « $9C$ » associée à la salle E permet à Carole de conclure qu'en passant par là, elle peut rejoindre la sortie en 10h au lieu des 11h en passant par B . Elle écrit donc $10E$ pour finaliser son plan.

Voici comment lire le chemin optimal une fois l'exploration terminée : on part du point d'arrivée, et remonte les salles en suivant la lettre indiquée dans l'annotation. Ici, la salle d'arrivée est la F : elle dispose de l'annotation « $10E$ », il faut donc passer par la salle E . Cette dernière est notée « $9C$ », il faut donc passer la salle C . Pour celle-ci, on lit « $4A$ » sur le plan de Carole, il faut donc venir de la salle A .

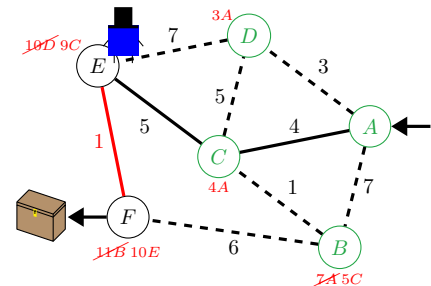
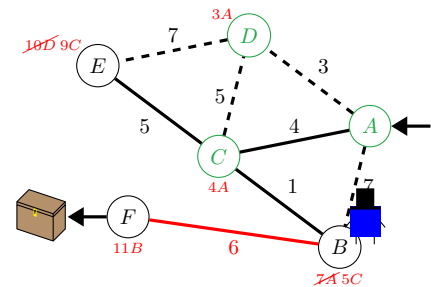
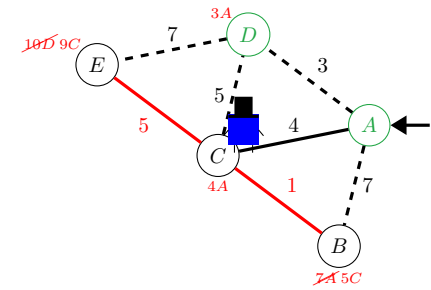
Le chemin le plus court est donc (lu à l'envers) : $F \leftarrow E \leftarrow C \leftarrow A$.

Remarque 4

Souvent, l'algorithme de Dijkstra est présenté dans un tableau, plutôt que d'annoter le graphe. Le fonctionnement est le même : chaque ligne représente un camp, et chaque colonne représente une salle. Le premier camp est établi dans la salle A , on met un symbole  pour le rappeler. Les distances sont annotées comme précédemment dans les colonnes des salles voisines. Les  servent à indiquer que la salle a été explorée et qu'il n'y a plus besoin de s'en soucier.

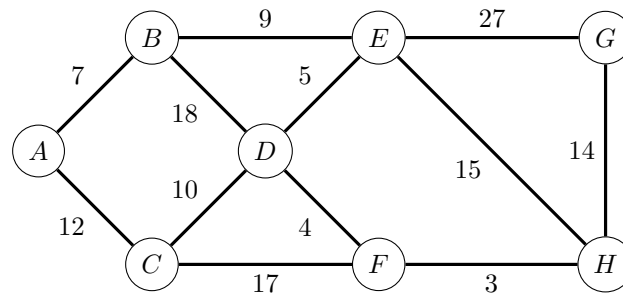
La lecture finale est la même : une fois toutes les salles explorées, on remonte les lettres en partant de l'arrivée.

	A	B	C	D	E	F
Camp n°1		7A	4A	3A		
Camp n°2		7A	4A		10D	
Camp n°3		5C			9C	
Camp n°4					9C	10B
Camp n°5						9E



Exercice 5

Le graphe ci-dessous représente les coûts de trajet (essence, péage, ...) selon la route empruntée. Par exemple, emprunter la route entre *A* et *B* vous coûtera 7€. Quelle trajet devez-vous emprunter pour faire le plus d'économies pour voyager entre *A* et *G* ?



B – Impacts sur les pratiques humaines

Les premiers systèmes de navigation embarqués sont antérieurs à l'invention du GPS. Dès les années 1930 avec l'*Iter Avto*, on voit apparaître des assistants de navigation. L'ouverture du GPS débridé au début du XXI^e siècle amorce la démocratisation des assistants numériques. Jusqu'en 2010, les assistants de navigation nécessitaient un appareil dédié (TomTom, Garmin, ...) : cette contrainte matérielle, ainsi que la robustesse du réseau sans fil, étaient les derniers freins avant la complète adoption de cette technologie, désormais accessible sur n'importe quel smartphone.

L'essor des assistants de navigation a de nombreux avantages :

- ▶ N'importe qui peut se repérer et suivre un itinéraire qu'il ne connaît pas sans se perdre ;
- ▶ Le conducteur peut porter son attention sur la route sans avoir à s'arrêter pour consulter la carte, ou pire, détourner les yeux de la route pour la consulter ;
- ▶ Les voyageurs peuvent directement se rendre de leur point de départ à leur destination, sans avoir besoin de s'arrêter en route pour demander le chemin, ou trouver une carte (papier) locale ;
- ▶ Les assistants de navigation peuvent minimiser le temps de trajet, mais aussi les coûts (notamment en péage), en consommation d'essence (et donc en empreinte carbone), ou même prévoir des arrêts stratégiques les moins fréquents possibles pour recharger la batterie des véhicules électriques si leur autonomie n'est pas suffisante.

Cependant, les assistants de navigation viennent avec leur lot d'inconvénients :

- ▶ À force de reléguer aux assistants cet effort de calcul d'itinéraire, les utilisateurs deviennent dépendants : ils deviennent incapables de se repérer sans l'aide des assistants. Des études maintenant célèbres menées par Eleanor Maguire ont montré que les chauffeurs de taxis londoniens, qui doivent mémoriser 25 000 rues pour obtenir leur licence, ont un cerveau plus développé dans la zone de la mémoire. La dépendance aux assistants de navigation diminue la capacité de notre cerveau à se construire une carte mentale et à mémoriser les lieux.
- ▶ Les instructions d'un assistant sont autosuffisantes : pour caricaturer, on pourrait presque conduire ou marcher les yeux fermés. De ce fait, il n'est plus nécessaire d'observer les alentours attentivement pour y trouver des repères. Il n'est pas nécessaire non plus de prendre des routes non prévues par l'itinéraire : il n'y a plus besoin d'explorer les environs. Ainsi, on peut passer à côté de points d'intérêt, de commerces, ... dont on n'aura pas connaissance à moins de chercher spécifiquement à s'y rendre. Pour les commerces, cela peut signifier une baisse de la fréquentation.

Une conséquence qui paraît bénéfique à première vue mais qui révèle ses effets pervers *a posteriori* est la répartition du trafic : typiquement, en cas de ralentissement sur votre itinéraire initial, votre assistant de navigation va tenter de trouver une route alternative, normalement plus lente, mais tout de même plus rapide étant donné la perte de temps due aux bouchons. On aurait tendance à dire que répartir le flux sur plusieurs itinéraires fluidifie le trafic, mais cela ne tient pas compte du fait que les routes alternatives ne sont pas conçues pour recevoir autant de trafic : une petite rue résidentielle calme et isolée d'habitude peut se retrouver inondée par des centaines de véhicules qui ont suivi leur

assistant de navigation qui leur a suggéré d'éviter le ralentissement sur la voie rapide à côté, afin de gagner quelques minutes. Résultat, les habitants se retrouvent avec un bouchon devant leur porte, ainsi que les pollutions sonores et atmosphériques qui vont avec.

2 – La sensibilité des données de position

Activité 6 (Dis moi où tu vas je te dirai qui tu es)

Dans cette activité, vous vous mettez dans la peau d'un courtier en données. Vous avez récolté une sélection de données sur plusieurs utilisateurs (Annexes B et suivantes). À partir de ces informations, dressez les profils des différents individus auxquels appartiennent ces données, afin de trouver quelles publicités sont les plus pertinentes pour chacun (Annexe A).

Information 7

Niveaux d'accès aux données

Les applications ont accès à trois (principaux) niveaux d'utilisation de la puce GPS d'un smartphone :

- ▶ **Toujours** : l'application peut accéder aux données même lorsqu'elle est fermée.
- ▶ **Pendant l'utilisation** : l'application ne peut accéder aux données que lorsqu'elle est active (ouverte et à l'écran).
- ▶ **Jamais** : l'application ne peut pas accéder du tout aux données sous aucun prétexte.

Il est courant que des niveaux intermédiaires soient proposés, par exemple « En arrière plan » : l'application peut accéder aux données lorsqu'elle est ouverte, mais pas nécessairement affichée à l'écran. Il est également possible d'exiger à l'application de redemander l'autorisation pour chaque utilisation. Dans le cas des données de position, l'utilisateur peut également choisir de donner accès à la position exacte (au mètre près) ou approximative.

Information 8

Conseils d'hygiène numérique

Il est important de réserver les différents niveaux d'accès à des utilisations pertinentes :

- ▶ **Toujours** : à restreindre le plus possible
- ▶ **Pendant l'utilisation** : pour le GPS, la livraison, ...
- ▶ **Jamais** : toutes les applications qui n'ont pas besoin de votre position pour fonctionner normalement (réseaux sociaux⁽¹⁾, jeux, ...)

Si votre calculatrice vous demande d'utiliser vos données GPS, méfiez vous !

Information 9

RGPD

En Europe, le RGPD (Règlement Général sur la Protection des Données) protège l'utilisateur. Ses quatre grands principes, appliqués dans le cadre des données de géolocalisation, sont :

- ▶ **Consentement libre et éclairé** : une application doit demander l'autorisation pour accéder aux données de la puce GPS. Une application n'a pas le droit de refuser de fonctionner en cas de refus d'accès à ces données.
- ▶ **Minimisation des données** : une application n'a le droit de collecter que le strict nécessaire à son bon fonctionnement. Par exemple, une application de météo n'a pas besoin de connaître votre position au mètre près pour vous donner des prévisions : une position à l'échelle de votre quartier ou de votre commune suffit.
- ▶ **Limitation de la conservation** : une application ne peut pas conserver indéfiniment vos données de géolocalisation.

(1). Comme illustré dans l'activité précédente, il peut être dangereux de partager sa position sur les réseaux sociaux, car cela permet de relier énormément d'informations à votre sujet !

- Accès et effacement : tout le monde peut demander à consulter ou supprimer définitivement les données qui le concernent.

Remarque 10 (Des données cachées)

Les photos peuvent aussi être des marqueurs de position : que ce soit parce qu'on reconnaît le paysage, ou parce que les coordonnées GPS ont été inscrites dans les métadonnées EXIF lors de la capture.

3 – Annexes

A – Publicités ciblées

- Publicités pour des moyens de transport
 - i▸ Une voiture (10 000 €)
 - ii▸ Un vélo cargo (3 000 €)
 - iii▸ Un lot de deux trottinettes électriques (600 €)
- Publicités ciblant les enfants
 - i▸ Un console de jeux
 - ii▸ Un set de cartes à collectionner
 - iii▸ Un jeu de blocs en bois
- Publicités pour des biens animaliers
 - i▸ Un harnais pour chien
 - ii▸ Une litière pour chat
 - iii▸ Un aquarium
- Publicités pour de la nourriture
 - i▸ Publicité de fast food
 - ii▸ Publicité pour des produits végétariens
 - iii▸ Publicité pour des produits bio

B – Données de géolocalisation

Profil 1 : - habite en bordure extérieure de périph - prend la voiture pour aller au travail () - a un enfant en crèche

C – Données de livraison

D – Publications sur les réseaux sociaux

Métadonnées EXIF d'une photo mettant en scène les 3 acteurs ensemble, permettant de les identifier définitivement parce qu'un jour ils se sont retrouvés tous les trois au même endroit

Thème d'étude V

Réseaux sociaux

Programme

Les réseaux sociaux

Introduction

Les réseaux sociaux sont des applications basées sur les technologies du Web qui offrent un service de mise en relation d'internautes pour ainsi développer des communautés d'intérêts.

Repères historiques

- ▶ 1995 : Classmates est l'un des premiers réseaux sociaux qui permettent aux étudiants de rester en relation ;
- ▶ 2003 : apparition de Myspace, aujourd'hui en perte de vitesse, et de LinkedIn (racheté depuis par Microsoft), à vocation professionnelle ;
- ▶ 2004 : apparition de Facebook, d'abord réservé aux étudiants de l'université Harvard, puis ouvert au grand public en 2006 ;
- ▶ 2006 : apparition de Twitter, qui permet l'échange de courts messages, limités au départ à 140 puis à 280 caractères (on parle de microblogage) ;
- ▶ 2009 : lancement de la messagerie instantanée WhatsApp (rachetée depuis par Facebook) qui se substitue à l'utilisation des SMS et MMS chez beaucoup d'utilisateurs ;
- ▶ 2010 : arrivée d'Instagram (racheté depuis par Facebook), qui permet le partage de photos et de vidéos ;
- ▶ 2011 : début de Snapchat qui permet, sur plateformes mobiles, le partage de photos et de vidéos, avec une limitation de durée ;
- ▶ 2018 : on estime à 3,2 milliards le nombre d'utilisateurs actifs des réseaux sociaux.

En 2018, les réseaux sociaux utilisés en France sont états-unis, toutefois il en existe bien d'autres : en Chine, par exemple, apparaît en 2009 l'application de microblogage Weibo avec plus de 350 millions d'utilisateurs actifs en 2018 ; en 2012 naît l'application de messagerie Weixin (développée par Tencent) qui compte en 2018 plus d'un milliard de comptes utilisateurs.

Les données et l'information

Les différents réseaux sociaux permettent l'échange d'informations de natures différentes : textes, photos, vidéos. Certains limitent strictement la taille des informations, d'autres autorisent la publication, mais de façon limitée dans le temps. Certains permettent l'adjonction d'applications tierces (plug-ins) qui peuvent ajouter des fonctionnalités supplémentaires.

Toutes les applications de réseautage social utilisent d'importantes bases de données qui gèrent leurs utilisateurs, l'ensemble des données qu'ils partagent, mais aussi celles qu'ils consentent à fournir (sans toujours le savoir), y compris sur leur vie personnelle.

Les algorithmes et les programmes

De très nombreux algorithmes sont mis en œuvre par les applications de réseautage social. Toutes les applications s'appuient sur des services de mise en relation avec des internautes membres du réseau, relations ou amis communs : des algorithmes opérant sur les graphes et sur les bases de données sont au cœur de ces services.

À l'aide d'algorithmes de recommandation, les réseaux sociaux suggèrent aux utilisateurs des amis, des contenus, des annonces promotionnelles. Ils permettent aussi aux plateformes sociales d'étudier les comportements de leurs utilisateurs à des fins commerciales, politiques ou d'amélioration du service.

Impacts sur les pratiques humaines

Le développement des réseaux sociaux introduit un nouveau type de liens sur le Web, qui ne relève pas de l'hypertexte : il s'agit de l'abonnement à des relations/des amis et de la possibilité de recommander de l'information en fonction du réseau ainsi constitué.

L'objectif annoncé des applications de réseautage social est de mettre les individus en relation les uns avec les autres. Quelle est la réalité ? L'expérience de Milgram (1967) semble indiquer la constitution de « petits mondes » où chacun est au plus à six liens de distance d'un autre. Peut-on éviter les phénomènes de communautés liés à des recommandations se renforçant les unes les autres pouvant aller jusqu'à un appauvrissement de la pensée critique ? Ces questions font référence au concept de bonding (renforcement de liens existants au sein d'un même groupe) versus bridging (construction de nouveaux liens non redondants).

Les affaires de fuite de données personnelles mettent en avant les questions liées aux modèles économiques des applications de réseautage social symbolisés par le slogan « quand c'est gratuit, c'est vous le produit ». Les réseaux sociaux peuvent être le support d'une cyberviolence, par le biais de photographies partagées sans consentement ou impossibles à retirer, par la diffusion de fausses nouvelles, de dénonciations ou de calomnies. Des pratiques, des outils et des services permettent de se protéger, lutter et dénoncer de tels agissements.

Sensibilisés au collège dans le cadre de l'éducation aux médias et à l'information, les lycéens doivent acquérir les démarches nécessaires pour se protéger et une conduite appropriée dans le cadre d'usages scolaires pour se préparer au monde professionnel. Les espaces numériques de travail (ENT) constituent le cadre privilégié de cet apprentissage.

Contenus	Capacités attendues
Identité numérique, e-réputation, identification, authentification	Connaître les principaux concepts liés à l'usage des réseaux sociaux.
Réseaux sociaux existants	Distinguer plusieurs réseaux sociaux selon leurs caractéristiques, y compris un ordre de grandeur de leurs nombres d'abonnés. Paramétrer des abonnements pour assurer la confidentialité de données personnelles. Modèle économique des réseaux sociaux Identifier les sources de revenus des entreprises de réseautage social.
Rayon, diamètre et centre d'un graphe	Déterminer ces caractéristiques sur des graphes simples. Notion de « petit monde » Expérience de Milgram Décrire comment l'information présentée par les réseaux sociaux est conditionnée par le choix préalable de ses amis.
Cyberviolence	Connaître les dispositions de l'article 222-33-2-2 du code pénal. Connaître les différentes formes de cyberviolence (harcèlement, discrimination, sexting...) et les ressources disponibles pour lutter contre la cyberviolence.

Exemples d'activités

- ▶ Construire ou utiliser une représentation du graphe des relations d'un utilisateur. S'appuyer sur la densité des liens pour identifier des groupes, des communautés.
- ▶ Sur des exemples de graphes simples, en informatique débranchée, étudier les notions de rayon, diamètre et centre d'un graphe, de manière à illustrer la notion de « petit monde ».
- ▶ Comparer les interfaces et fonctionnalités de différents réseaux sociaux.
- ▶ Dresser un comparatif des formats de données, des possibilités d'échange ou d'approbation (bouton like), de la persistance des données entre différents réseaux sociaux.
- ▶ Analyser les paramètres d'utilisation d'un réseau social. Analyser les autorisations données aux applications tierces.
- ▶ Discuter des garanties d'authenticité des comptes utilisateurs ou des images.
- ▶ Lire et expliquer les conditions générales d'utilisation d'un réseau social.
- ▶ Consulter le site nonauharcèlement.education.gouv.fr.

10 Les réseaux sociaux

Table des matières

Thème d'étude VI

Objets connectés

Programme

Informatique embarquée et objets connectés

Introduction

Embarquer l'informatique dans les objets a beaucoup d'avantages : simplifier leur fonctionnement, leur donner plus de possibilités d'usage et de sûreté, et leur permettre d'intégrer de nouvelles possibilités à matériel constant par simple modification de leur logiciel.

Après avoir transformé les chaînes de montage des automobiles et les avions dans les années quatre-vingt-dix, l'informatique intervient maintenant dans des domaines toujours plus nombreux : automobile, réseau ferroviaire et transports urbains, domotique, robotique, loisirs, etc., conduisant à un nouvel internet des objets.

Pour les avions par exemple, l'informatique gère le vol en commandant finement des servomoteurs électriques, plus légers et plus fiables que les vérins hydrauliques, les réacteurs, la navigation et le pilotage automatique, et permet l'atterrissage automatique par temps de brouillard. Elle a eu un impact décisif sur l'amélioration de la sécurité aérienne.

Les objets informatisés avaient autrefois des interfaces homme-machine (IHM) dédiées, souvent dépendantes d'une liaison filaire directe. Mais les technologies du Web intégrées au téléphone portable permettent maintenant d'y rassembler les interfaces des objets du quotidien, ce qui en simplifie et uniformise l'usage. Les objets informatisés deviennent ainsi connectés.

Repères historiques

- ▶ 1967 : premier système embarqué de guidage lors de la mission lunaire Apollo ;
- ▶ 1971 : premier processeur produit par Intel ;
- ▶ 1984 : sortie de l'Airbus 320, premier avion équipé de commandes électriques informatisées ;
- ▶ 1998 : mise en service du métro informatisé sans conducteur Météor (ligne 14 à Paris) ;
- ▶ 1999 : introduction de l'expression « internet des objets » par Kevin Ashton ;
- ▶ 2007 : arrivée du smartphone.

On estime à 50 milliards le nombre d'objets connectés en 2020.

Les données et l'information

Dans les systèmes informatiques embarqués, l'information provient soit des IHM soit des capteurs, pour contrôler automatiquement ou manuellement le fonctionnement physique par des actionneurs et transmettre des informations aux utilisateurs. Le flux d'informations à travers les IHM permet ainsi une interaction continue entre l'homme et la machine.

Les algorithmes et les programmes

Le développement des logiciels embarqués est délicat, car il pose souvent des questions de temps-réel, c'est-à-dire de respect de temps de réponse imposé. Ceci conduit à des méthodes de programmation spécifiques.

Les machines

Les microprocesseurs sont beaucoup plus nombreux dans les objets que dans les ordinateurs et téléphones, mais ils sont souvent plus petits, moins chers et moins rapides. Les capteurs et actionneurs reposent sur des technologies physiques et électroniques variées, allant quelquefois vers l'électronique de puissance. Un problème essentiel est la réduction de la consommation électrique, surtout pour les appareils sur pile.

Impacts sur les pratiques humaines

L'impact de l'informatisation des objets devient considérable, surtout depuis que leurs interfaces s'unifient. Le but est de fabriquer des machines d'utilisation facile permettant des fonctionnalités améliorées, voire complètement nouvelles comme la voiture autonome. Celle-ci utilise à la fois des techniques de systèmes embarqués pour son fonctionnement et sa navigation et de l'intelligence artificielle pour l'analyse en temps-réel de l'environnement à l'aide de capteurs variés (caméras, radars, lidars, etc.).

Comme l'informatique embarquée interagit avec le monde physique en exposant quelquefois des vies humaines ou des équipements critiques (réseaux électriques par exemple), elle est soumise à de fortes contraintes de sûreté (absence d'erreurs) et de sécurité (résistance aux attaques). En avionique, ferroviaire ou autres applications critiques, des processus lourds de certification externe sont utilisés. Cependant, dans beaucoup de systèmes embarqués moins critiques, la sécurité reste souvent un point faible, et les objets connectés sont de plus en plus utilisés comme robots pour lancer des attaques sur internet.

Contenus	Capacités attendues
Systèmes informatiques embarqués	Identifier des algorithmes de contrôle des comportements physiques à travers les données des capteurs, l'IHM et les actions des actionneurs dans des systèmes courants.
Interface homme-machine (IHM)	Réaliser une IHM simple d'un objet connecté.
Commande d'un actionneur, acquisition des données d'un capteur	Écrire des programmes simples d'acquisition de données ou de commande d'un actionneur.

Exemples d'activités

- ▶ Identifier les évolutions apportées par les algorithmes au contrôle des freins et du moteur d'une automobile, ou à l'assistance au pédalage d'un vélo électrique.
- ▶ Réaliser une IHM pouvant piloter deux ou trois actionneurs et acquérir les données d'un ou deux capteurs.
- ▶ Gérer des entrées/sorties à travers les ports utilisés par le système.
- ▶ Utiliser un tableau de correspondance entre caractères envoyés ou reçus et commandes physiques (exemple : le moteur A est piloté à 50% de sa vitesse maximale lorsque le robot reçoit la chaîne de caractères « A50 »).

11 Informatique embarquée

Table des matières

Lunette connectées, aspirateurs robots : problème de la confiance

Voitures autonomes

https://youtu.be/GArLwf5Wo2w?si=h-p_DhrfbGtNUne7

Thème d'étude VII

Intelligences artificielles

Programme

12 L'histoire des IA

Turing

Voir vidéo Science étonnante + 3b1b

13

Les enjeux modernes

2.1. Données, biais et représentations

Le rôle central des données d'entraînement : « ce qui entre produit ce qui sort ».

Les biais algorithmiques : comment les stéréotypes et discriminations du monde réel sont reproduits par les machines.

La souveraineté numérique, le respect de la vie privée (RGPD) et la concentration du marché.

2.2. Enjeux environnementaux et matériels

La réalité physique derrière le virtuel : infrastructures, centres de données, cartes graphiques.

L'empreinte écologique : consommation d'énergie, d'eau pour le refroidissement et extraction des ressources matérielles.

2.3. Démocratie, information et société

La manipulation des contenus : désinformation, deepfakes et hallucination des modèles.

Les questions éthiques et juridiques : droit d'auteur, propriété intellectuelle et protection de la création humaine.

Impact sur les métiers et égalité filles-garçons dans la tech.

14 Vers un usage pertinent

3.1. Anatomie et fonctionnement d'un prompt

Comprendre comment interroger efficacement une IA (contexte, rôle, consignes, format).

L'IA comme partenaire de travail ou assistant, et non comme un substitut à la pensée réflexive.

3.2. Évaluation critique des résultats

Apprendre à douter : vérifier les sources, repérer les erreurs factuelles (« hallucinations ») et analyser la neutralité des réponses.

Méthodologie d'hygiène numérique face aux contenus générés (textes, images, audios).

3.3. Intégration dans le cadre scolaire et citoyen

Les règles de bon usage dans le cadre du Cadre de Référence des Compétences Numériques (CRCN).

Développer son autonomie numérique pour ne pas subir l'outil, mais le maîtriser.

Algorithmique et programmation

Programme

La démarche algorithmique est, depuis les origines, une composante essentielle de l'activité mathématique. Au cycle 4, en mathématiques et en technologie, les élèves ont appris à écrire, mettre au point et exécuter un programme simple. Une consolidation des acquis du cycle 4 est proposée autour de deux idées essentielles :

- la notion de fonction ;
- la programmation comme production d'un texte dans un langage informatique.

Dans le cadre de cette activité, les élèves s'exercent à :

- décrire des algorithmes en langage naturel ou dans un langage de programmation ;
- en réaliser quelques-uns à l'aide d'un programme simple écrit dans un langage de programmation textuel ;
- interpréter, compléter ou modifier des algorithmes plus complexes.

Un langage de programmation simple d'usage est nécessaire pour l'écriture des programmes informatiques. Le langage choisi est Python, langage interprété, concis, largement répandu et pouvant fonctionner dans une diversité d'environnements. Les élèves sont entraînés à passer du langage naturel à Python et inversement.

L'algorithmique a une place naturelle dans tous les champs des mathématiques et les problèmes ainsi traités doivent être en relation avec les autres parties du programme (fonctions, géométrie, statistiques et probabilité, logique) mais aussi avec les autres disciplines ou la vie courante.

À l'occasion de l'écriture d'algorithmes et de petits programmes, il convient de transmettre aux élèves l'exigence d'exactitude et de rigueur, et de les entraîner aux pratiques systématiques de vérification et de contrôle. En programmant, les élèves revisitent les notions de variables et de fonctions sous une forme différente.

Histoire des mathématiques

Les textes évoqués dans la thématique « Nombres et calculs » indiquent une préoccupation algorithmique tout au long de l'Histoire. Lorsqu'un texte historique a une visée algorithmique, transformer les méthodes qu'il présente en un algorithme, voire en un programme, ou inversement, est l'occasion de travailler des changements de registre qui donnent du sens au formalisme mathématique.

Variables et instructions élémentaires

Contenus

- Variables informatiques de type entier, booléen, flottant, chaîne de caractères.
- Affectation (notée $\leftarrow$ en langage naturel).
- Séquence d'instructions.
- Instruction conditionnelle.
- Boucle bornée (**for**), boucle non bornée (**while**).

Capacités attendues

- Choisir ou déterminer le type d'une variable (entier, flottant ou chaîne de caractères).
- Concevoir et écrire une instruction d'affectation, une séquence d'instructions, une instruction conditionnelle.
- Écrire une formule permettant un calcul combinant des variables.

-
- ▶ Programmer, dans des cas simples, une boucle bornée, une boucle non bornée.
 - ▶ Dans des cas plus complexes : lire, comprendre, modifier ou compléter un algorithme ou un programme.

Notion de fonction

Contenus

- ▶ Fonctions à un ou plusieurs arguments.
- ▶ Fonction renvoyant un nombre aléatoire. Série statistique obtenue par la répétition de l'appel d'une telle fonction.

Capacités attendues

- ▶ Écrire des fonctions simples ; lire, comprendre, modifier, compléter des fonctions plus complexes. Appeler une fonction.
- ▶ Lire et comprendre une fonction renvoyant une moyenne, un écart type. Aucune connaissance sur les listes n'est exigée.
- ▶ Écrire des fonctions renvoyant le résultat numérique d'une expérience aléatoire, d'une répétition d'expériences aléatoires indépendantes.

1 Algorithmique

Table des matières

1	Les variables informatiques	159
2	Les structures conditionnelles	160
A	Les blocs <i>SI... SINON SI...</i> , <i>SINON...</i>	160
B	Les opérateurs logiques	161
3	Les structures itératives	161
A	La boucle inconditionnelle (ou bornée)	161
B	La boucle conditionnelle (ou non bornée)	162
4	Les fonctions	163
	Exercices	164

Programme

Utiliser les variables et les instructions élémentaires

Contenus

- Variables informatiques de type entier, booléen, flottant, chaîne de caractères.
- Affectation (notée $\leftarrow$ en langage naturel).
- Séquence d'instructions.
- Instruction conditionnelle.
- Boucle bornée (for), boucle non bornée (while).

Capacités attendues

- Choisir ou déterminer le type d'une variable (entier, flottant ou chaîne de caractères).
- Concevoir et écrire une instruction d'affectation, une séquence d'instructions, une instruction conditionnelle.
- Écrire une formule permettant un calcul combinant des variables.
- Programmer, dans des cas simples, une boucle bornée, une boucle non bornée.
- Dans des cas plus complexes : lire, comprendre, modifier ou compléter un algorithme ou un programme.

Notion de fonction

Contenus

- Fonctions à un ou plusieurs arguments.
- Fonction renvoyant un nombre aléatoire. Série statistique obtenue par la répétition de l'appel d'une telle fonction.

Capacités attendues

- Écrire des fonctions simples ; lire, comprendre, modifier, compléter des fonctions plus complexes. Appeler une fonction.
- Lire et comprendre une fonction renvoyant une moyenne, un écart type. Aucune connaissance sur les listes n'est exigée.
- Écrire des fonctions renvoyant le résultat numérique d'une expérience aléatoire, d'une répétition d'expériences aléatoires indépendantes.

Définition 1**Algorithme**

Un *algorithme* est une suite finie d'instructions qui permet de faire quelque chose.

Exemple 2

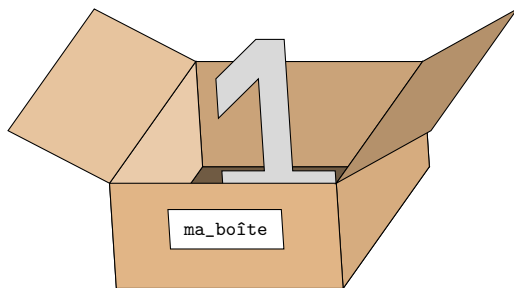
Une recette de cuisine est une sorte d'algorithme : c'est une suite d'instructions qui transforme des ingrédients en un plat. Les directions données par un GPS forment un algorithme, qui permettent d'aller d'un endroit à un autre.

1 – Les variables informatiques

Définition 3

- Une *variable* (informatique) est un élément dans la mémoire de l'ordinateur, qui possède un nom et contient une valeur.
- Le *type* d'une variable est la nature de la valeur qu'elle contient : un nombre entier, décimal, un mot, une image ...
- Lorsqu'on donne une certaine valeur à une variable, on dit qu'on lui *affecte* cette valeur.

On illustre souvent une variable par une *boîte* : une boîte qui porte un nom permettant de l'identifier, et contenant une certaine valeur que l'on peut sortir et remettre dans la boîte à volonté.



Cette boîte (cette variable), qui s'appelle `ma_boîte` contient la valeur 1.
On peut écrire `ma_boîte = 1`.

En *langage naturel*, l'affectation d'une valeur à une variable est représentée par une flèche. Par exemple, $X \leftarrow 5$ signifie « on affecte la valeur 5 à la variable X ». La variable X est donc de type *entier*.

Exemple 4

Voici ci-contre un exemple d'algorithme en langage naturel. Détaillons son fonctionnement par étapes :

- 1► On commence par affecter la valeur 2 à une variable qu'on appelle a.
- 2► Ensuite, on affecte la valeur 3 à une variable qu'on appelle b.
- 3► Ensuite, on affecte à une variable c la valeur de $a + b$: ici, on lui affecte la valeur 5, puisque $5 = 2 + 3$.
- 4► Enfin, on affiche la valeur contenue dans la variable c, autrement dit on affiche 5.

```

1  a ← 2
2  b ← 3
3  c ← a + b
4  AFFICHER c

```

Exercices conseillés : 23, 24

Remarque 5

Une *variable* s'appelle ainsi car son contenu est amené à varier, à changer le long du déroulement de l'algorithme. On peut donc réaffecter une nouvelle valeur à une variable déjà existante : auquel cas, l'ordinateur oublie

purement et simplement l'ancienne valeur, et ne garde que la nouvelle. C'est comme si on vidait le contenu de la boîte pour le remplacer.

On peut également redéfinir une variable à partir d'elle-même : par exemple, $a \leftarrow 2 * a$ indique que l'on remplace la valeur de a par son double.

Exercices conseillés : 25, 26, 27, 28

2 – Les structures conditionnelles

A – Les blocs *SI... SINON SI..., SINON...*

Définition 6

- ▶ Une *condition* est la valeur de vérité d'une phrase pouvant dépendre d'une variable. Une condition est donc soit Vraie, soit Fausse.
- ▶ Une *structure conditionnelle* est une suite de *blocs d'instructions* que l'on exécute si une certaine condition est Vraie, et que l'on n'exécute pas sinon.
- ▶ Les blocs d'instructions sont signalés en décalant les instructions d'un cran vers la droite : on dit qu'on les *indente* (l'espace est appelée un *indentation*).
- ▶ Une variable qui vaut soit Vrai, soit Faux est dite *booléenne* (ou de *type booléen*).

Exemple 7

Cet algorithme contient une structure conditionnelle (composée de deux blocs d'instructions) qui dépend de la variable *âge* : si celle-ci est supérieure ou égale à 18, alors l'algorithme affiche le texte "Vous êtes majeur", et dans le cas contraire (c'est-à-dire si *âge* est strictement inférieur à 18), il affiche le texte "Vous êtes mineur". En langage naturel, la structure conditionnelle est délimitée, en langage naturel, par SI et FIN SI.

```

1  SI âge ≥ 18:
2      AFFICHER "Vous êtes majeur"
3  SINON:
4      AFFICHER "Vous êtes mineur"
5  FIN SI

```

Remarque 8 (Enchaîner les test)

On peut aussi enchaîner les tests les uns après les autres avec le mot-clé SINON SI... :

```

1  SI x ≥ 5:
2      AFFICHER "x est plus grand que 5:"
3  SINON SI x ≥ 3:
4      AFFICHER "x est plus grand que 3"
5  SINON:
6      AFFICHER "Je ne sais pas..."
7  FIN SI

```

Cette structure conditionnelle est constituée de trois blocs. Tout d'abord, l'algorithme va vérifier si $x \geq 5$: si c'est effectivement le cas, alors l'algorithme affiche "x est plus grand que 5" et ignore les autres blocs d'instruction de la structure conditionnelle. Dans le cas contraire, il lance le test suivant, et ainsi de suite. Le bloc SINON est exécuté seulement quand tous les tests précédents ont échoué.

Notez bien cette subtilité : l'algorithme exécute un et un seul bloc d'instructions dans chaque structure conditionnelle, même si parfois les conditions se superposent. Par exemple avec $x = 6$, l'algorithme ci-dessus affiche "x est plus grand que 5" car c'est le premier test réussi – qu'importe que le deuxième soit aussi valide.

Remarque 9

Le bloc **SINON** : est optionnel : si on ne souhaite exécuter aucune instruction dans le cas où tous les tests échouent, il suffit de ne pas écrire le bloc **SINON** :

Exercices conseillés : 29, 30, 31, 32, 33

B – Les opérateurs logiques

Définition 10

Soient A et B deux variables booléennes.

- ▶ Le booléen A **ET** B est le booléen qui est Vrai si, et seulement si A et B sont simultanément Vrais.
- ▶ Le booléen A **OU** B est le booléen qui est Vrai si, et seulement si au moins l'une des deux variables A ou B est vraie Vrais.
- ▶ Le booléen **NON** A est le booléen qui est Vrai si, et seulement si A est Faux.

On résume cette définition dans des *tables de vérité* :

A	B	A ET B
V	V	V
V	F	F
F	V	F
F	F	F

A	B	A OU B
V	V	V
V	F	V
F	V	V
F	F	F

A	NON A
V	F
F	V

Remarque 11

Parfois, on entoure les conditions de parenthèses, soit pour améliorer la lisibilité de l'algorithme, soit parce que c'est nécessaire.

Exercice conseillé : 34

3 – Les structures itératives

Définition 12

- ▶ Une *structure itérative*, ou *boucle*, est une suite d'instruction qu'on répète un certain nombre de fois.
- ▶ Chaque exécution du bloc d'instruction est appelée *itération* de la boucle.

A – La boucle inconditionnelle (ou bornée)

Définition 13

Une *boucle inconditionnelle*, ou *boucle bornée*, est un bloc d'instructions que l'on répète un nombre de fois déterminé à l'avance.

Une boucle inconditionnelle est délimitée, en langage naturel, par **POUR** et **FIN POUR** (à ce titre, on l'appelle également « *boucle Pour* »).

Exemple 14

Dans l'algorithme suivant, le *compteur* n prend successivement les valeurs 1, puis 2, puis 3, ... jusqu'à 10. Pour chaque valeur de n , l'algorithme exécute les instructions du bloc d'instructions repéré par une indentation : ici, on affiche le double de n (on a donc affiché la table de 2!).

```

1  POUR n ALLANT DE 1 A 10:
2      AFFICHER 2 * n
3  FIN POUR

```

Exercices conseillés : 35, 36, 37, 38

B – La boucle conditionnelle (ou non bornée)

Définition 15

Une *boucle conditionnelle*, ou *boucle non bornée*, est un bloc d'instructions que l'on répète un nombre de fois inconnu en avance.

Une boucle conditionnelle est délimitée, en langage naturel, par TANT QUE et FIN TANT QUE (à ce titre, on l'appelle aussi « *boucle Tant que* »).

Exemple 16

Dans cet algorithme, la variable x prend successivement les valeurs 0, puis 2, puis 4, et ainsi de suite, jusqu'à ce que la condition $2 * x < 10$ ne soit plus vérifiée (autrement dit, ici jusqu'à ce que x dépasse la valeur 5).

```

1  x ← 0
2  TANT QUE 2*x < 10:
3      x ← x + 2
4  FIN TANT QUE

```

Remarque 17

Il est possible de ne pas exécuter le bloc d'instruction d'une boucle Tant que si sa condition est fausse dès le départ. Par exemple, l'algorithme suivant n'affichera rien.

```

1  x ← 10
2  TANT QUE x ≤ 5:
3      AFFICHER "x est plus petit que 5"
4  FIN TANT QUE

```

Définition 18

Incrémenter une variable (de type entier), c'est augmenter sa valeur de 1.

Remarque 19

Il est possible de bricoler une boucle Pour à partir d'une boucle Tant que en créant un compteur que l'on incrémente à la main à chaque itération. En revanche, il n'est pas possible de bricoler une boucle Tant que à partir d'une boucle Pour!

En pratique, il est préférable d'utiliser une boucle Pour dès que c'est possible, car une boucle Pour est plus lisible et plus sûre d'utilisation. En effet, la boucle Tant que ci-contre tourne indéfiniment!

```

1  x ← 10
2  TANT QUE x > 0:
3      x ← x * 2
4  FIN TANT QUE

```

Exercices conseillés : 39, 40, 41, 42

4 – Les fonctions

Définition 20

- ▶ Une *fonction* est un bloc d'instructions qui porte un nom, et qui est exécuté uniquement sur demande.
- ▶ Une fonction peut admettre en entrée des *arguments* (ou *paramètres*), et peut, après avoir exécuté ses instructions, *renvoyer* un résultat.
- ▶ Dans l'algorithme principal, quand on utilise une fonction, on dit qu'on *appelle* cette fonction.

Une fonction est délimitée, en langage naturel, par **FONCTION** et **FIN FONCTION**. Si une fonction admet des arguments, on les précise entre parenthèse, séparés par des virgules, après le nom de la fonction (on laisse une paire de parenthèses vide s'il n'y a pas d'arguments). Le bloc d'instructions définissant une fonction est signalé par une indentation.

Pour appeler une fonction, on écrit son nom dans l'algorithme, suivi de ses paramètres entre parenthèses (ou une paire de parenthèses vide).

Exemple 21

Dans de nombreux jeux ou certaines activités d'initiation à la programmation, un personnage peut faire certaines actions, déclenchées à l'aide de fonctions. Par exemple `avancer()`, `droite()`, `poser()`. Certains tests sont aussi effectués grâce à des fonctions qui renvoient un booléen, par exemple `est_bleu()`, `mur_devant()`, `est_libre()`, ...

Exemple 22

Dans les exemples suivants, on propose une fonction à gauche, et un exemple

- 1▶ La fonction `BonjourMonde` suivante ne prend pas d'argument en entrée et ne renvoie aucune valeur. Elle se contente d'afficher "Bonjour, monde !" et "Hello, world !".

```
1  FONCTION BonjourMonde():
2      AFFICHER "Bonjour, monde !"
3      AFFICHER "Hello, world !"
4  FIN FONCTION
```

```
1  AFFICHER "Salutation du monde en
2      cours..."
3  BonjourMonde()
4  AFFICHER "Le monde est salué !"
```

- 2▶ La fonction `Doubler` suivante prend comme argument un nombre, et renvoie son double.

```
1  FONCTION Doubler(n):
2      RENVOYER 2 * n
3  FIN FONCTION
```

```
1  AFFICHER "Le double de 5 est:"
2  AFFICHER Doubler(5)
3  AFFICHER "Et le double de 8 est:"
4  AFFICHER Doubler(8)
```

- 3▶ La fonction `Additionner` suivante prend comme argument trois nombres, et renvoie leur somme après avoir affiché un petit texte.

```
1  FONCTION Additionner(a, b, c):
2      AFFICHER "On additionne
3          tout ça !"
4      RENVOYER a + b + c
5  FIN FONCTION
```

```
1  AFFICHER "Prenons les nombres 1, 2
2      et 3..."
3  AFFICHER Additionner(1, 2, 3)
```

Exercices conseillés : 43, 44, 45, 46, 47

Exercices

Variables et affectation

Exercice 23

Donnez les valeurs affichées dans les deux algorithmes ci-dessous (remarque : le symbole * indique une multiplication) :

```
1  facteur_1 ← 8
2  facteur_2 ← 5
3  resultat ← facteur_1 * facteur_2
4  AFFICHER resultat
```

```
1  x ← 6
2  y ← 7
3  z ← x * y - 2 * x
4  AFFICHER z - 3 * y
```

Exercice 24 (Déclic 2022)

On considère l'algorithme ci-contre, où la variable x est un nombre donné en amont. Les affirmations suivantes sont-elles vraies ou fausses ? Justifier.

```
1  a ← x + 1
2  b ← a*a
3  c ← 3*b
```

- 1► Si la variable x contient la valeur 2 avant l'exécution de l'algorithme, alors la variable c contiendra 27.
- 2► Si la variable x contient la valeur -3 avant l'exécution de l'algorithme, alors la variable c contiendra -12.
- 3► La variable c contient toujours un réel positif.

Exercice 25 (Tableau d'avancement)

Compléter le tableau suivant représentant les valeurs des variables x et y à chaque étape de l'algorithme :

```
1  x ← 3
2  y ← 2
3  x ← 2*x - 1
4  y ← y*y + 3
5  x ← x - y
6  y ← x*y
```

Étape	x	y
1		
2		
3		
4		
5		
6		

Exercice 26

Donnez les valeurs affichées dans les deux algorithmes ci-dessous (remarque : le symbole / indique une division) :

```
1  a ← 12
2  a ← a - 4
3  b ← 2
4  c ← a / b
5  b ← 10
6  AFFICHER c
```

```
1  a ← 5
2  b ← 0
3  b ← a
4  a ← a / b
5  AFFICHER a
```

Exercice 27 (Déclic 2022)

On considère le programme de calcul suivant : « choisir un nombre X , prendre son double, soustraire 1, puis élever au carré. »

Parmi les algorithmes ci-dessous, indiquer ceux pour lesquels la variable X contient le résultat du programme de calcul à la fin de l'exécution.

Algo A

```
1  A ← 2*X
2  B ← A - 1
3  X ← B*B
```

Algo B

```
1  X ← 2*X
2  X ← X - 1
3  X ← X*X
```

Algo C

```
1  X ← 2*X
2  X ← 2*X - 1
3  X ← (2*X - 1)*(2*X - 1)
```

Exercice 28

Écrire un algorithme permettant d'échanger les valeurs de deux variables a et b .

Structures conditionnelles et logique

Exercice 29 (Déclic 2022)

On considère l'algorithme ci-contre où N est une variable entière.

```

1  SI  $N \geq 10$ :
2       $X \leftarrow 2*N$ 
3  SINON:
4       $X \leftarrow N + 3$ 
5  FIN SI
```

- 1► Si N vaut 5, alors X vaut-elle 10, 8 ou 5 ?
- 2► Si N vaut 10, alors X vaut-elle 20, 13 ou 10 ?
- 3► Si N vaut 15, alors X vaut-elle 30, 18 ou 15 ?

Exercice 30

Qu'affichent les deux algorithmes suivants pour $x = 2$? Et pour $x = 8$?

Algo A :

```

1  SI  $x \leq 5$ :
2      AFFICHER "Plus petit que 5"
3  SINON SI  $x \leq 10$ :
4      AFFICHER "Plus petit que 10"
5  FIN SI
```

Algo B :

```

1  SI  $x \leq 5$ :
2      AFFICHER "Plus petit que 5"
3  FIN SI
4
5  SI  $x \leq 10$ :
6      AFFICHER "Plus petit que 10"
7  FIN SI
```

Exercice 31

Les variables a et b représentent des nombres. Que fait l'algorithme suivant ?

```

1  SI  $a > b$ :
2       $M \leftarrow a$ 
3  SINON:
4       $M \leftarrow b$ 
5  FIN SI
6  AFFICHER M
```

Exercice 32

L'algorithme suivant affiche un message en fonction du score dans un jeu, représenté par la variable score.

```

1  SI score = 100:
2      AFFICHER "Score parfait !"
3  SINON SI score  $\geq 75$ :
4      AFFICHER "Excellent !"
5  SINON SI score  $\geq 50$ :
6      AFFICHER "Bravo !"
7  SINON:
8      AFFICHER "Dommage !"
9  FIN SI
```

Qu'affiche ce programme pour les valeurs de score suivantes ?

► 100 ► 90 ► 70 ► 50 ► 24 ► 0

Exercice 33

Écrire un algorithme qui affiche la plus grande valeur entre trois variables a , b et c .

Exercice 34

Qu'affiche l'algorithme suivant...

Lorsque jour = "Mardi" et meteo = "Pluie" ?

Lorsque jour = "Dimanche" et meteo = "Soleil" ?

```

1  SI jour = "Lundi" ET meteo = "Pluie":
2      AFFICHER "Au boulot..."
3  SINON SI (NON jour = "Dimanche") OU meteo = "Pluie":
4      AFFICHER "N'oublions pas le parapluie"
5  SINON SI (NON jour  $\neq$  "Dimanche") OU (NON meteo  $\neq$  "Pluie"):
6      AFFICHER "Direction la plage !"
7  SINON:
8      AFFICHER "Que vais-je faire aujourd'hui..."
9  FIN SI
```

Boucle bornée

Exercice 35 (Déclic 2022)

Dans l'algorithme ci-dessous, la variable N désigne un entier plus grand que 2.

```

1  S ← 0
2  POUR i ALLANT DE 1 A N:
3      S ← S + 10
4  FIN POUR

```

1► Dans cette question on choisit $N = 3$ avant d'exécuter l'algorithme.

a► Recopier et compléter le tableau d'avancement suivant :

i	S	N
	0	
1		
2		
⋮	⋮	⋮

b► En déduire la valeur de S à la fin du programme.

2► Déterminer la valeur finale de S lorsqu'on donne à N la valeur...

a► 4 b► 5 c► 6 d► 10

Exercice 36

Déterminer la valeur de la variable S à l'issue de cet algorithme :

```

1  S ← 0
2  POUR n ALLANT DE 1 A 10:
3      S ← S + n
4  FIN POUR

```

Exercice 37 (Math'x)

Déterminer l'affichage des deux scripts suivants :
Algo A :

```

1  a ← - 5
2  POUR k ALLANT DE 1 A 5:
3      a ← 10*a - 8
4  FIN POUR
5  AFFICHER a

```

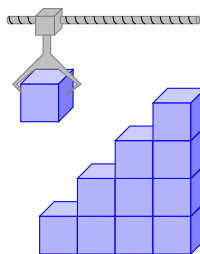
Algo B :

```

1  a ← - 5
2  POUR k ALLANT DE 1 A 5:
3      a ← 10*a - 8
4      AFFICHER a
5  FIN POUR

```

Exercice 38 (Défi)



Une pince mécanique dépose des blocs (on supposera que le prochain bloc apparaît magiquement entre ses crochets après avoir posé le bloc précédent). La pince dispose de trois commandes :

- Écrivez `droite()` dans le programme pour que la pince se décale d'un cran vers la droite.
- Écrivez `gauche()` dans le programme pour que la pince se décale d'un cran vers la gauche.
- Écrivez `poser()` pour que la pince pose un bloc.

Objectif : écrire un programme qui construit un escalier à N marches (sur l'illustration, $N = 4$).

Boucle non bornée

Exercice 39

- 1► Déterminez la valeur de la variable e à l'issue de cet algorithme.
- 2► Interpréter cet algorithme.

(Remarque : la notation $**$ indique l'opération Puissance. Par exemple, $3**2 = 3^2 = 9$.)

```

1  e ← 1
2  TANT QUE 2**e < 500:
3      e ← e + 1
4  FIN TANT QUE

```

Exercice 40 (Suite de Syracuse)

On construit une suite de nombres de la manière suivante : on choisit un nombre, et :

- si ce nombre est pair, alors on le divise par 2,
- sinon, on le triple et on ajoute 1,

```

1  ...
2  TANT QUE ...:
3      SI N%2 ...:
4          N ← ...
5      SINON:
6          N ← ...
7      FIN SI
8      ...
9  FIN TANT QUE

```

et on recommence avec le nouveau résultat, ainsi de suite.

La conjecture de Syracuse stipule que quelque soit le nombre que l'on choisit au départ, on finit par atteindre 1.

Compléter le programme ci-dessus pour qu'il calcule et affiche les nombres successifs de la suite. On rappelle que $N\%2$ vaut le reste de la division euclidienne de N par 2.

Exercice 41 (Objectif bac)

On modélise l'évolution du nombre d'abeilles (en milliers) dans un rucher. Au départ, il y a 1 000 abeilles. À cause de la pollution, 20% du rucher se meurt chaque mois. Pour pallier leur disparition, on introduit 20 milliers d'abeilles à la fin de chaque mois. On donne l'algorithme correspondant à la fin de l'énoncé.

- 1► Recopier et compléter le tableau d'avancement suivant :

abeilles	n	abeilles > 500 ?
	1	
1000	1	Vrai
⋮	⋮	⋮

- 2► Déterminez la valeur de la variable n à l'issue de cet algorithme.
- 3► Interpréter cette valeur.

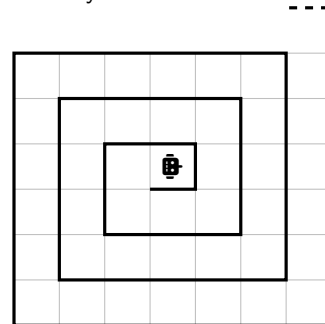
```

1  n ← 1
2  abeilles ← 1000
3  TANT QUE abeilles > 500:
4      abeilles ← 0.8*abeilles + 20
5      n ← n + 1
6  FIN TANT QUE

```

Exercice 42 (Défi)

Votre robot est au centre d'un « labyrinthe » en forme de spirale comme illustré ci-dessous. Vous devez mettre à jour son programme pour le faire sortir de là, mais problème... il ne vous reste de la place que pour 6 lignes de code, et vous ne savez pas quelle est la taille du labyrinthe !



Votre robot dispose des commandes suivantes :

- Effectuez le test `SI mur_devant()` : (ou analogue) pour savoir si votre robot fait face à un mur et réagir en conséquence.
- Écrivez `avancer()` pour que votre robot avance d'un pas.
- Écrivez `droite()` pour que le robot tourne sur lui-même vers la droite. De même avec `gauche()` pour tourner à gauche.
- Effectuez le test `SI est_libre()` : (ou analogue) pour savoir si le robot est sorti du labyrinthe.

Comment allez-vous vous sortir de cette situation ?

Fonctions

Exercice 43 (Déclic 2022)

On considère la fonction `mystere` ci-contre.

- 1► Quels sont les paramètres de cette fonction ?
- 2► Quels sont les renvois des appels suivants ?

a► `mystere(3,4)` b► `mystere(12,5)`

```

1  FONCTION mystere(x, y):
2      SI x < 10:
3          RENVOYER 3*x - y
4      SINON:
5          RENVOYER x + y
6      FIN SI
7  FIN FONCTION

```

```

1  FONCTION OperationMystere(a, b):
2      q ← 0
3      TANT QUE a > b:
4          q ← q + 1
5          a ← a - b
6      FIN TANT QUE
7      RENVOYER q, a - b * q
8  FIN FONCTION

```

Exercice 44

On définit une fonction `rectangle` qui renvoie l'aire et le périmètre d'un rectangle dont les dimensions sont données en paramètres.

Compléter cette fonction.

```

1  FONCTION rectangle(a, b):
2      aire = ...
3      peri = ..
4      RENVOYER ..., ...
5  FIN FONCTION

```

Exercice 46

Programmer une fonction `f` qui prend en paramètre un nombre x et renvoie $3x + 2$ si $x < 1$, et $5 - 2x$ sinon.

Exercice 45

On définit la fonction `OperationMystere` ci-dessous, qui prend deux nombres en paramètres et renvoie deux nombres.

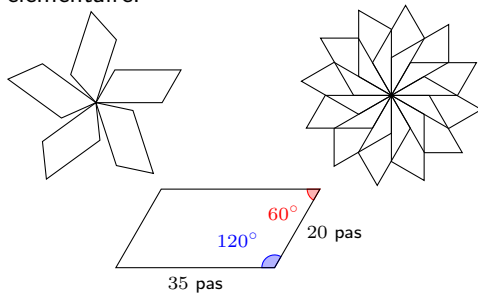
- 1► Que renvoient les appels suivants ?
 - a► `OperationMystere(10, 3)`
 - b► `OperationMystere(3, 10)`
 - c► `OperationMystere(16, 2)`
- 2► À quoi sert cette fonction ?

Exercice 47

Un logiciel de dessin par programmation dispose des commandes suivantes :

- ▶ `poser_crayon()` pour commencer à dessiner,
- ▶ `lever_crayon()` pour arrêter de dessiner,
- ▶ `avancer(x)` pour avancer de x unités vers l'avant,
- ▶ `droite(angle)` pour tourner le crayon d'un angle de angle degrés dans le sens horaire .
- ▶ `gauche(angle)` similaire à la précédente, mais dans le sens inverse,

On veut dessiner un motif de fleur à partir d'un pétale élémentaire.



Un pétale

- 1▶ Compléter la fonction `Petale` ci-dessous qui dessine un pétale élémentaire du motif.

```

1  FONCTION Petale():
2      ...
3      POUR i ALLANT DE ... A ...:
4          avancer(...)
5          gauche(...)
6          avancer(...)
7          gauche(...)
8      FIN POUR
9      ...
10  FIN FONCTION

```

- 2▶ Écrire la fonction `Fleur` qui prend en paramètre le nombre de pétales à dessiner et dessine la fleur avec des pétales également espacés.

Problèmes de géométrie

On reprend les fonctions de l'exercice 47.

Exercice 48

- 1► Écrire une fonction `Carre` qui trace un carré de côté 50 unités.
- 2► Écrire une fonction `Rectangle` qui trace un rectangle long de 100 unités et large de 50 unités.
- 3► Écrire un programme réalisant la frise suivante :



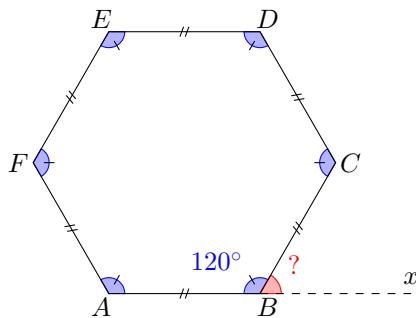
- 4► Écrire un programme réalisant la frise suivante :



Exercice 49

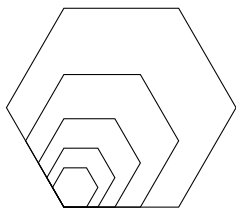
Un hexagone régulier est un polygone à 6 côtés de même longueur, et dont tous les angles mesurent 120° .

- 1► Quelle est la mesure de l'angle $\widehat{CBx}$ ci-dessous ?



Dans chacune des questions suivantes, la fonction `Hexagone` prend en paramètre la variable `cote`, et trace un hexagone tel que ci-dessus dont les côtés mesurent `cote` unités.

- 2► Dans cette question, on souhaite construire la figure suivante. Le plus petit hexagone a un côté de longueur 32, et chaque hexagone a un côté 1,5 fois plus grand que le précédent.



- a► Écrire la fonction `Hexagone` de sorte que l'hexagone soit tracé en partant du point `A`

et en tournant dans le sens inverse des aiguilles d'une montre (autrement dit on part de `A`, puis on va en `B`, puis en `C`, etc. jusqu'à revenir en `A`).

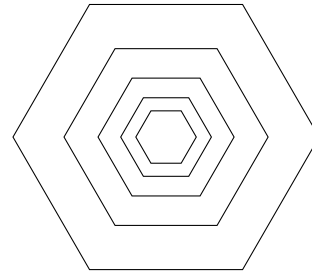
- b► Écrire une fonction `Grossissement` prenant un paramètre entier `n`, et dessinant la figure demandée avec `n` hexagones.

- 3► Dans cette question, on souhaite construire la figure suivante. La longueur des côtés de chaque hexagone est 30 unités.



- a► Modifier la fonction `Hexagone` de la question précédente pour débiter du point `F` au lieu du point `A`.
- b► Dans les hexagones de la figure à tracer, quelle distance sépare deux sommets opposés ?
- c► En déduire une fonction `Frise` prenant un paramètre entier `n`, et dessinant la frise demandée avec `n` hexagones.

- 4► Dans cette question, on souhaite construire la figure suivante. Le plus petit hexagone a un côté de longueur 32, et chaque hexagone a un côté 1,5 fois plus grand que le précédent.



- a► Modifier la fonction `Hexagone` de la question précédente de sorte que l'hexagone soit tracé selon la logique suivante :
 - On part d'un point qui sera le centre de l'hexagone,
 - On décale le crayon sur le sommet de droite,
 - On trace l'hexagone,
 - On ramène le crayon au centre.
- b► Écrire une fonction `Tunnel` prenant un paramètre entier `n`, et dessinant la figure demandée avec `n` hexagones.

Des boucles un peu plus corsées

Exercice 50

Que fait la fonction suivante ?

```

1  FONCTION mystere(n):
2    POUR i ALLANT DE 1 A n-1:
3      POUR j ALLANT DE 1 A n-1:
4        SI i+j = n:
5          AFFICHER i, " + ", j, " = ", n
6        FIN SI
7      FIN POUR
8    FIN POUR
9  FIN FONCTION

```

Exercice 51

En vous inspirant de l'exercice 50, écrire une fonction qui trouve toutes les manières d'écrire un entier n comme le produit de deux entiers différents de 1.

Exercice 52

En utilisant une boucle POUR i ALLANT DE 0 A 10, afficher un compte à rebours de 10 à 0.

Exercice 53

L'objectif de cet exercice est d'écrire une fonction qui détermine si un nombre est premier. La procédure est la suivante : pour savoir si un nombre n est premier, on vérifie qu'il n'est divisible par aucun entier entre 2 et $n - 1$.

- ▶ Écrire le squelette de la fonction `est_premier` qui prend en paramètre un entier n .
- ▶ Compléter la phrase suivante : « n est divisible par k si, et seulement si le ... de la division euclidienne de ... par ... vaut ... »
- ▶ En déduire un test étant vrai lorsque n n'est pas divisible par une variable k en langage naturel.

Le corps de notre fonction sera le suivant, à compléter avec les prochaines questions :

```

1  k <- ...
2  TANT QUE ...:
3    k <- ...
4  FIN TANT QUE
5  SI ...:
6    RENVOYER ...
7  SINON:
8    RENVOYER ...
9  FIN SI

```

- ▶ Le boucle doit s'arrêter lorsque l'on trouve un diviseur. Compléter les deux premières lignes.
- ▶ Quel est le rôle de la variable k ? Compléter la ligne 3.

Étudions les deux cas de figure, où n est soit premier, soit non.

- ▶ Dans cette question, supposons n premier.
 - a▶ Rappeler la définition d'un nombre premier.
 - b▶ Quel est le plus petit diviseur positif de n qui n'est pas 1 ?
 - c▶ En déduire la valeur de k à la sortie de la boucle.
- ▶ Dans cette question, supposons n non premier.
 - a▶ On note a le plus petit diviseur de n positif et différent de 1. Comparer a et n le plus précisément possible.
 - b▶ Que renvoie le test de la boucle pour les valeurs de k entre 2 et a .
 - c▶ En déduire la valeur de k à la sortie de la boucle.
- ▶ Déduire des deux questions précédentes un test à effectuer sur la valeur finale de k pour savoir si n est premier ou non.
- ▶ Compléter les dernières lignes du corps de la fonction pour qu'elle renvoie VRAI si n est premier, et FAUX sinon.
- ▶ Bonus : simplifier la fin de la fonction pour remplacer tout le bloc SI ... SINON ... en une seule ligne RENVOYER ...

Jeux élémentaires

Exercice 54 (Jeu des bâtonnets – ★★☆☆☆)

Dans le jeu des bâtonnets, deux joueurs s'affrontent. Un certain nombre de bâtonnets sont disposés sur la table, et tour à tour les deux joueurs prennent 1, 2 ou 3 bâtonnets. Le joueur qui est forcé de prendre le dernier bâtonnet perd.

- 1► On veut écrire une fonction `afficher_batons` qui prend en paramètre le nombre de bâtonnets et affiche ces bâtonnets sous la forme de traits verticaux entrecoupés d'espaces.

```
| | | | |
```

- a► Les bâtonnets et les espaces représentent un frise. Quel est son motif élémentaire ?
- b► On note n le paramètre de la fonction `afficher_batons`. Combien de motifs élémentaires va-t-on tracer ?
- c► En déduire le type de boucle à utiliser.
- d► Une seule instruction se trouvera à l'intérieur de la boucle, celle qui affiche un motif élémentaire. L'implémenter.

- 2► Définir une variable `nbBatons` qui vaut 15.

On va désormais implémenter la *boucle (principale) de jeu*, c'est-à-dire ce qui doit se passer tant qu'il reste des bâtonnets.

- 3► Sait-on à l'avance le nombre de tours que le jeu durera ? En déduire le type de boucle que l'on va utiliser, et l'implémenter.
- 4► À chaque tour de jeu (autrement dit à chaque tour de boucle), on doit réaliser les actions suivantes :
 - Demander au joueur combien de bâtons il souhaite retirer,
 - Mettre à jour le nombre de bâtons restant,
 - Afficher les bâtons restant.

Compléter le bout de code suivant :

```
1  ENTRER retirer
2  nbBatons <- ...
3  ...
```

- 5► Vous avez un support élémentaire pour le jeu. Laissez parler votre créativité, et améliorez certains éléments ! Tenir compte du tour de chaque joueur, déterminer qui gagne et qui perd, afficher un message de victoire...

Exercice 55 (Le juste prix – ★★★★★)

Le jeu du juste prix est le suivant : le joueur doit deviner un prix en faisant des propositions, et le présentateur dit au joueur si sa proposition est au-dessus ou en-dessous du prix recherché.

- 1► Définir une variable `juste_prix` qui prend un entier aléatoire entre 1 et 100. On utilisera la fonction `ENTALEA` qui prend deux paramètres a et b , et tire un entier aléatoire entre a et b inclus.
- 2► Définir une variable `proposition`, qui va contenir les propositions successives du joueur. On fera attention à bien choisir sa valeur initiale.

On va désormais implémenter la *boucle (principale) de jeu*, c'est-à-dire ce qui doit se passer tant que le joueur n'a pas gagné.

- 3► Sait-on à l'avance le nombre de tours que le joueur prendra pour gagner ? En déduire le type de boucle que l'on va utiliser, et l'implémenter.
- 4► À chaque tour de jeu (autrement dit à chaque tour de boucle), on doit réaliser les actions suivantes :
 - On récupère la proposition du joueur et on l'affiche,
 - Si sa proposition est plus grande que le juste prix, on dit au joueur "C'est moins",
 - Si sa proposition est plus petite que le juste prix, on dit au joueur "C'est plus".

Compléter le bout de code suivant :

```
1  ENTRER proposition
2  AFFICHER ...
3  SI ...:
4      AFFICHER ...
5  SINON SI ...:
6      AFFICHER ...
7  FIN SI
```

- 5► Ajouter un message de victoire quand le joueur a gagné.
- 6► Bonus 1 : définir une fonction qui prend en paramètre un entier n et qui déroule le jeu en faisant deviner un nombre entre 1 et n .
- 7► Bonus 2 : faire en sorte que le joueur n'ait droit qu'à 10 essais.

Exercice 56 (Jeu d'argent – ★★★★★)

Dans un jeu d'argent, on mise 1 euro pour jouer, et on lance deux dés à 6 faces : si on tombe sur un double 6, on gagne 30 euros, sinon on ne gagne rien et la mise est perdue.

On souhaite évaluer les risques de ce jeu : on va simuler un certain nombre de joueurs, et voir combien d'entre eux ressortent perdants au bout de 10 tours de jeu.

- 1► Écrire une fonction `lancer` qui ne prend aucun paramètre, simule de lancer de deux dés à 6 faces et renvoie "gagnant" si le lancer est gagnant (paire de 6) et "perdant" s'il est perdant perdant.

On utilisera la fonction `ENTALEA` qui prend deux paramètres `a` et `b`, et tire un entier aléatoire entre `a` et `b` inclus.

- 2► a► Créer le squelette d'une fonction `partie` qui prend en paramètre une variable `capitalDepart`.
Les questions suivantes servent à remplir le corps de la fonction.
- b► Initialiser une variable `argent` avec la valeur de `capitalDepart`. Cela correspond à l'argent dont on dispose avant de jouer.
- c► Le jeu est composé de 10 tours. Quel type de boucle va-t-on utiliser ? L'implémenter.
- d► À chaque tour de jeu, on :
- mise 1 euro,
 - lance les deux dés,
 - gagne de l'argent ou non selon le résultat.

Compléter et expliquer le code suivant, et le simplifier si possible :

```
1  argent <- ... # mise
2  SI lancer() == "gagnant":
3      argent <- ...
4  SINON:
5      ...
6  FIN SI
```

- e► Renvoyer l'argent du joueur à la fin de la partie.

- 3► Afin de répondre à la question initiale, on va procéder ainsi :

- on initialise un compteur `gagnants` à 0, qui comptera le nombre de gagnants,

- on crée une boucle qui fait 1000 parties et incrémente `gagnants` pour chaque partie gagnante,

- on calcule la proportion de parties gagnantes.

On suppose que chaque joueur dispose de 100 euros au départ. Compléter le programme suivant :

```
1  gagnants <- 0
2  POUR i ALLANT DE 1 A ...:
3      resultat <- partie(...)
4      SI ...:
5          gagnants <- gagnants + 1
6      FIN SI
7      AFFICHER gagnants/...
8  FIN POUR
```

- 4► Interpréter ce résultat.

2 Programmer en Python

Table des matières

1	Les interfaces	175
A	La console	175
B	Exécuter un script	175
C	Le Notebook Jupyter	175
2	Les logiciels	176
A	Basthon	176
B	EduPython	177

Programme

Utiliser les variables et les instructions élémentaires

Contenus

- ▶ Variables informatiques de type entier, booléen, flottant, chaîne de caractères.
- ▶ Affectation (notée $\leftarrow$ en langage naturel).
- ▶ Séquence d'instructions.
- ▶ Instruction conditionnelle.
- ▶ Boucle bornée (for), boucle non bornée (while).

Capacités attendues

- ▶ Choisir ou déterminer le type d'une variable (entier, flottant ou chaîne de caractères).
- ▶ Concevoir et écrire une instruction d'affectation, une séquence d'instructions, une instruction conditionnelle.
- ▶ Écrire une formule permettant un calcul combinant des variables.
- ▶ Programmer, dans des cas simples, une boucle bornée, une boucle non bornée.
- ▶ Dans des cas plus complexes : lire, comprendre, modifier ou compléter un algorithme ou un programme.

Notion de fonction

Contenus

- ▶ Fonctions à un ou plusieurs arguments.
- ▶ Fonction renvoyant un nombre aléatoire. Série statistique obtenue par la répétition de l'appel d'une telle fonction.

Capacités attendues

- ▶ Écrire des fonctions simples ; lire, comprendre, modifier, compléter des fonctions plus complexes. Appeler une fonction.
- ▶ Lire et comprendre une fonction renvoyant une moyenne, un écart type. Aucune connaissance sur les listes n'est exigée.
- ▶ Écrire des fonctions renvoyant le résultat numérique d'une expérience aléatoire, d'une répétition d'expériences aléatoires indépendantes.

Introduction

Python est un langage de programmation, sorti dans sa première version en 1991, devenu l'un des langages les plus populaire pour sa simplicité écrasante et ses possibilités illimitées. Un langage de programmation n'est rien d'autre qu'un moyen de traduire un algorithme qu'un humain peut comprendre en une suite d'instructions qu'un ordinateur peut comprendre. Python est un langage *interprété*, c'est-à-dire qu'un *interpréteur* va traduire votre code en langage machine à *chaque fois* que vous l'exécutez (par opposition aux langages *compilés* qui sont traduits une bonne fois pour toutes).

1 – Les interfaces

A – La console

La *console* de Python est un espace dans lequel vous pouvez entrer du code et l'interpréter au fil de l'eau. On écrit les lignes de code devant les `>>>`, et on presse la touche Entrée pour valider. Elle est idéale pour tester des codes élémentaires, avec une seule ligne ou une seule boucle par exemple. Les résultats s'affichent systématiquement. Elle fonctionne également comme une calculatrice.

```
>>> 2 + 5
7
>>> 2 - 5
-3
>>> a = 6
>>> a = a + 5
>>> a
11
```

B – Exécuter un script

Pour des projets plus complexes, on préférera écrire un *script* à part, et l'*exécuter* une fois tout le script rédigé. Un script Python est un fichier avec l'extension `.py`, bien que, dans les éditeurs que nous verrons plus bas, souvent le système de fichiers est caché.



Exemple avec un script complet à gauche et son exécution à droite

C – Le Notebook Jupyter

Un *Notebook Jupyter* (ou *Notebook* pour faire court) est à mi-chemin entre la console et le script. Il est constitué de *cellules*, qui sont des morceaux de scripts que l'on peut exécuter indépendamment, et dont les résultats s'affichent en-dessous. C'est idéal pour présenter des résultats intermédiaires et produire des documents mêlant du code et du texte, voire des images.

```

Entrée[1]: from turtle import *
           from random import randint

           print("Les modules ont été importés")

           Les modules ont été importés

Entrée[2]: alice = Turtle()

           print("La tortue s'appelle Alice")

           La tortue s'appelle Alice

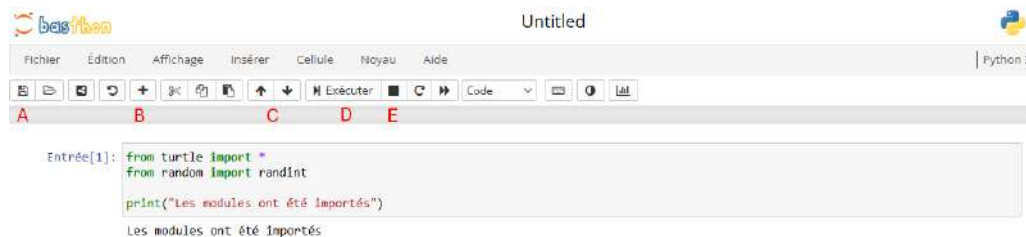
Entrée[3]: for _ in range(200):
           alice.forward(10)
           alice.left(randint(-90,91))
           done()

```



Exemple de Notebook avec trois cellules

Bien que son implémentation peut différer d'un logiciel à l'autre, les commandes d'un Notebook Jupyter sont toujours sensiblement les mêmes :



- A► Sauvegarder le projet (Raccourci clavier : Ctrl + S)
- B► Ajouter une cellule en-dessous (Raccourci clavier : a)
- C► Déplacer une cellule vers le haut ou le bas
- D► Exécuter la cellule sélectionnée (Raccourci clavier : Ctrl + Entrée pour simplement exécuter la cellule, Shift + Entrée pour exécuter la cellule et sélectionner la suivante)
- E► Interrompre le programme quand une cellule est coincée

2 – Les logiciels

Python n'est pas le nom d'un logiciel permettant de coder en Python. Il ne désigne que le langage en lui-même, autrement dit le programme qui transforme et exécute le code que l'on écrit en Python. Pour écrire ledit code, il faut un *éditeur de code* ou un *environnement de développement intégré* (*IDE* en anglais).

A – Basthon

Le site basthon.fr permet de coder en Python de toutes les manières possibles, listées dans la précédente section. En mode console on peut écrire le script à gauche et l'exécuter avec le bouton Exécuter en bas à gauche, ou directement écrire dans la console à droite, où s'affichent les résultats.



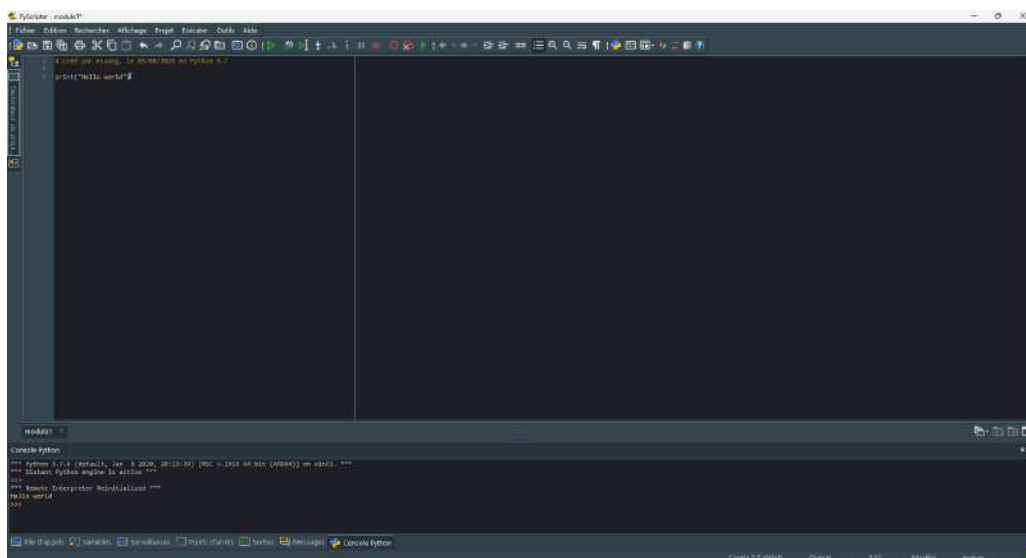
Bien que pratique pour de petits projets à la volée, cette méthode a quelques inconvénients : elle ne dispose pas de toutes les fonctionnalités disponibles dans les autres méthodes, notamment tout ce qui touche aux graphiques et aux dessins, et sa puissance de calcul est limitée. Un programme trop exigeant peut faire planter la page.

Basthon propose également un Notebook, qui donne accès aux fonctionnalités graphiques comme illustré sur les captures de la précédente section, mais qui souffre du même problème de performances. En pratique, il sera bien suffisant pour l'année de seconde.

Le site Capytale de l'académie de Paris, à l'adresse <https://capytale2.ac-paris.fr>, utilise Basthon.

B – EduPython

EduPython est un logiciel tout-en-un qui permet de tout faire : écrire directement dans la console, faire des scripts complets, utiliser les fonctionnalités graphiques mentionnées précédemment, et faire des Notebooks. Il est en général préinstallé sur les ordinateurs de lycée, et sert parfois dans les autres disciplines. Son interface est certes plus complète que Basthon, mais elle est d'autant plus austère.



Au lancement du logiciel, vous disposez d'un éditeur de code à gauche, de la console en bas, et d'un bouton vert triangulaire dans la barre de navigation en haut pour exécuter votre script.

3 Python : variables et types

1 – Affectation en Python

Pour affecter une valeur à une variable (déjà existante ou non) en Python, on utilise le symbole = :

Langage naturel

```
1 var ← 3
```

Scratch

mettre var à 3

Python

```
var = 3
```

Remarque 1

Vous ne pouvez pas nommer vos variables n'importe comment :

- ▶ Un nom de variable doit commencer par une lettre (ou un tiret bas, mais ça a une signification particulière...). Par exemple, *truc ou 2bidule ne sont pas des noms valides.
- ▶ Un nom de variable peut contenir des chiffres ou des tirets bas, mais pas d'autres caractères spéciaux. Par exemple, liste1, nombre_2, de3gil et carte_4_monde sont des noms valides.
- ▶ On ne peut pas utiliser un mot clé réservé par Python : par exemple, on ne peut pas nommer une variable `if`, `for`, `while`, ... La liste complète des mots-clé réservés se trouve à la fin du chapitre.
- ▶ Un nom de variable ne doit pas contenir de caractère accentué.
- ▶ Les majuscules et minuscules ne sont pas équivalentes : ainsi, Truc et truc désignent deux variables différentes. (On dit que les noms de variables sont *sensibles à la casse*).

Lorsque vous choisissez un nom pour une variable, préférez l'explicite à l'implicite : par exemple, si une variable contient un prix, appelez-la `prix` plutôt que `juste p...` L'essentiel, c'est que vous puissiez déterminer d'un coup d'œil ce que représente une variable dans un code. Exemple avec une fonction (voir le paragraphe sur le sujet) qui calcule un intérêt :

Les variables sont mal nommées

```
1 def itr(t, c, a):
2     for i in range(a):
3         c = c * t
4     return c
5
6 itr(1.1, 1000, 5)
```

Les variables sont bien nommées

```
1 def interets(taux, capital, annees):
2     for i in range(annees):
3         capital = capital * taux
4     return capital
5
6 interets(1.1, 1000, 5)
```

Vous pouvez retenir la règle suivante : lorsque votre variable n'a aucune interprétation particulière, vous pouvez la nommer avec juste une lettre. Sinon, nommez-la de manière à décrire ce qu'elle représente. Toutefois, évitez les noms à rallonge !

Exception : quand vous tapez du code sur une calculatrice lycée, vous pouvez vous permettre de nommer vos variables avec juste une lettre... c'est déjà assez pénible comme ça.

Remarque 2 (Pour les experts)

Il arrive très fréquemment de devoir redéfinir une variable à partir d'elle-même. Par exemple, lui ajouter 1 (autrement dit, l'*incrémenter*) :

Langage naturel

```
1 var ← var + 1
```

Scratch



Python

```
var = var + 1
```

En Python, il existe une notation raccourcie quand on redéfinit une variable à partir d'elle-même par une opération simple (addition, soustraction, multiplication, division). Par exemple, `var = var + 1` se raccourcit en `var += 1`. Autre exemple, `var = var * 5` se raccourcit en `var *= 5`.

2 – Les différents types

Voici une liste non exhaustive des types de données que vous serez les plus susceptibles de rencontrer dans un programme Python :

Type (en français)	Type Python	Exemple
Entier (relatif)	<code>int</code>	<code>n = 2</code>
Flottant	<code>float</code>	<code>x = 2.3</code>
Chaîne de caractères	<code>str</code>	<code>texte = "Bonjour monde !"</code>
Booléen	<code>bool</code>	<code>A = True</code>

Pour connaître le type d'un objet `x`, on utilise la fonction `type` : par exemple, `type(12)` affiche `<class 'int'>`.

Remarque 3 (Pour les experts)

Python est un langage que l'on dit *orienté objet*. Pour résumer très grossièrement, ça veut dire que tout en Python est un *objet*. Il y a les objets de type `int`, les objets de type `bool`, ... Une fonction renvoie toujours un objet – quand elle ne renvoie rien, elle renvoie en fait un objet de type `NoneType`. La fonction `type` renvoie un objet de type `type` : ce n'est pas une chaîne de caractère (autrement dit, ce n'est pas un objet de type `str`) ! Pour vérifier, par exemple, si un objet est de type `int`, on ne peut pas écrire le test (voir le paragraphe sur les tests) `type(x) == 'int'` : on doit plutôt écrire `isinstance(x, int)`, ce qui renvoie un booléen.

Exercice conseillé : 21

A – Les entiers

Dans Python, les nombres entiers sont les nombres sans virgule, positifs comme négatifs.

Remarque 4

Bien que mathématiquement, 12 et 12,0 sont égaux, dans Python, ils n'ont pas le même statut : le premier est de type `int`, le second est de type `float`.

Les nombres entiers en Python sont principalement utilisés dans deux contextes : pour faire des compteurs, et pour faire de l'arithmétique. Pour l'utilisation arithmétique, voici les opérations que l'on peut effectuer entre des entiers que l'on rencontre le plus souvent :

Opération	Notation Python	Exemple
Addition	+	>>> 2 + 3 5
Soustraction	-	>>> 2 - 3 -1
Multiplication	*	>>> 2 * 3 6
Puissance	**	>>> 2 ** 3 8
Quotient de la division euclidienne	//	>>> 13 // 4 3
Reste de la division euclidienne	%	>>> 13 % 4 1

Remarque 5

On peut calculer une division entre deux entier, mais le résultat sera toujours de type `float`, même si mathématiquement il est entier. Par exemple, l'instruction `12/3` renvoie `4.0` et non `4`.

B – Les flottants

Un nombre de type flottant est un nombre décimal qui contient une virgule apparente. Par exemple, `2.3` est de type `float` (attention, Python utilise la convention anglaise pour les nombres décimaux, on doit donc écrire un point et non une virgule!). Cependant, il faut bien noter une subtilité, où Python se détache du monde mathématique. En effet, `12.0` est de type `float`, bien que mathématiquement il est égal à un nombre entier. C'est pourquoi on définit les nombres de type flottant comme des nombres décimaux *avec une virgule apparente* (on rappelle qu'un nombre entier est un type particulier de nombre décimal!).

Le type `float` supporte les opérations d'addition (symbole `+`), de soustraction (symbole `-`), de multiplication (symbole `*`), de puissance (symbole `**`) et de division (symbole `/`).

Remarque 6

En Python, comme dans de nombreux langages de programmation, on ne travaille qu'avec des nombres décimaux qui ont un certain nombre de chiffres après la virgule (par exemple, on peut représenter tous les nombres qui ont au plus 55 décimales après la virgule). En effet, pour stocker toutes les décimales d'un nombre non décimal... il faudrait une infinité d'espace mémoire.

Python travaille donc *toujours* avec des arrondis : essayez donc de calculer `0.1 + 0.2` avec Python...

Voir l'activité ?? pour plus d'explications.

Remarque 7

Pour transformer un nombre entier en nombre flottant, on peut soit rajouter un simple point si on donne le nombre explicitement (exemple : `12.`, qui est un raccourci pour `12.0`), soit écrire `float(x)` pour transformer une variable `x`.

Remarque 8

La syntaxe `int(3.4)` renvoie `3` (qui est de type `int`). Attention cependant, la syntaxe `int(-3.4)` renvoie `-3` (qui est aussi de type `int`).

Ainsi, appliquer la fonction `int` à un nombre flottant ne renvoie pas sa partie entière : elle renvoie le nombre tronqué.

Remarque 9

Python comprend l'écriture scientifique d'une certaine manière l'écriture scientifique : par exemple, $1,5 \times 10^3$ s'écrit `1.5e3`. Autre exemple, $3,95 \times 10^{-8}$ s'écrit `3.95e-8`. Dernier exemple, $8,113 \times 10^{275}$ s'écrit `8.113e275`. Il suffit d'écrire la mantisse, puis un e, puis l'exposant sans parenthèses.

C – Les chaînes de caractères

Les chaînes de caractère (*string* en anglais, d'où le type `str`) sont délimitées par des guillemets simples ou doubles. Exemples : `"Bonjour monde !"` et `'Hello world !'` sont des chaînes de caractères.

Remarque 10

La fonction `input` renvoie toujours une chaîne de caractères ! Même si on tape un nombre, par exemple `1.2`, le résultat renvoyé sera la chaîne de caractères `'1.2'`.

Remarque 11

Appeler les fonctions `int` ou `float` sur une chaîne de caractère qui représente un nombre transformera cette dernière en le nombre qu'elle contient. Par exemple, `float('1.2')` renvoie `1.2` (de type `float`). En revanche, cela ne fonctionnera pas si vous essayez de convertir autre chose qu'un nombre : Python n'y arrivera pas et vous affichera une erreur.

Passons en revue quelques fonctions et opérations utiles avec les chaînes de caractères :

Fonction/Opération	Description	Exemple
Concaténation	Coller bout à bout deux chaînes de caractères	<pre>>>> "Bonjour" + "Monde" "BonjourMonde"</pre>
Répétition	Répéter bout à bout une chaîne de caractères un certain nombre de fois	<pre>>>> "Hello"*3 "HelloHelloHello"</pre>
Longueur	Obtenir le nombre de caractères	<pre>>>> len("monde !") 7</pre>

Remarque 12 (Pour les confirmés)

On peut également accéder à un caractère spécifique d'une chaîne de caractères :

```
>>> texte = "Bonjour"
>>> texte[2]
'n'
```

Ainsi on accède au caractère en position 2... ce qui correspond à la troisième lettre ! En effet, comme la quasi-totalité des langages de programmation, *Python compte les positions (les indices) à partir de 0*.

Remarque : on ne peut pas changer un caractère d'une chaîne déjà existante : autrement dit, la syntaxe `texte[2] = 'e'` ne fonctionne pas (on dit que les chaînes de caractères sont *immuables*).

Remarque 13 (Pour les experts)

Le *slicing* correspond à l'action de récupérer une sous-chaîne : par exemple, de "Bonjour" je souhaite récupérer "njour". Pour ça, on utilise la syntaxe suivante : `texte[2:6]`. Le premier nombre correspond à la position du premier caractère (elle est donc incluse dans le découpage), et le second nombre correspond à la position du dernier caractère +1 (autrement dit, c'est la position qui suit la dernière de la sous-chaîne), elle est donc exclue. Pour récupérer les 3 premiers caractères (donc jusqu'à la position 2!), c'est-à-dire "Bon", on peut écrire `texte[0:3]`, mais il existe la notation raccourcie `texte[:3]`.

Pour récupérer tous les caractères à partir du quatrième (donc en position 3!), on peut écrire `texte[4:7]`, mais il existe la notation raccourcie `texte[4:]`.

On peut facilement accéder au dernier caractère par la syntaxe `texte[-1]`. On peut se servir de cela pour récupérer, par exemple, toute la chaîne sauf le dernier caractère, on peut écrire `texte[:-1]`.

Dernière possibilité de *slicing* : la syntaxe `texte[1:6:2]` va extraire la sous-chaîne partant de la position 1 jusqu'à la position 5, en prenant un caractère sur deux.

Dans la même veine, une astuce utile les la syntaxe suivante : la syntaxe `texte[6:1:-1]` va donner la sous-chaîne des caractères entre les positions 2 et 6, mais à l'envers... en particulier, `texte[::-1]` permet de renverser la chaîne.

Remarque 14 (Pour les experts)

Souvent, on a besoin d'inclure une variable dans une chaîne de caractères. La manière naïve de le faire est la suivante :

```
print("La somme de " + str(x) + " et " + str(y) + " vaut " + str(x + y))
```

(Remarque, il faut impérativement convertir les variables `x` et `y` en chaînes de caractères *via* la fonction `str`!) C'est lourd, et c'est pénible à écrire. Une manière plus pratique est la suivante :

```
print("La somme de {0} et {1} vaut {2}".format(x, y, x+y))
```

C'est ce qu'on appelle *formater* une chaîne de caractères. Ici, 0, 1, et 2 entre accolades renvoie à l'ordre des arguments de la fonction `format` (n'oubliez pas qu'on compte à partir de 0!)

Depuis la version 3.6 de Python, il existe une manière encore plus rapide :

```
print(f"La somme de {x} et {y} vaut {x+y}")
```

Le petit `f` devant la chaîne de caractère (que l'on appelle en conséquence un *f-string*) indique à Python qu'il doit remplacer ce qui est entre accolades par les variables qui s'y trouvent.

Une autre syntaxe très pratique est la suivante :

```
print(f"La valeur de x est environ {x:.2f}")
```

Le `:.2f` signifie qu'on tronque le nombre avec deux chiffres après la virgule (le 2 indique le nombre de décimales, le `f` est pour *floating (point)*, il indique la virgule). Il y a plein d'autres petites astuces de formatage du genre, cherchez-les sur Internet.

D – Les booléens

Une variable booléenne (autrement dit de type `bool` ne peut prendre que deux valeurs : soit `True`, soit `False`.

Remarque 15

Quand on applique la fonction `bool` à un objet quelconque, on le transforme en booléen. Mais quelle est la valeur de vérité de 12 ou de "Bonjour" ?

Python a fait le choix suivant : quand une variable numérique vaut 0 ou 0., et quand une chaîne de caractères est vide (c'est-à-dire juste ""), la valeur de vérité est `False`. Dans tous les autres cas, elle vaut `True`.

Mais il est très (très) rarement utile de devoir convertir une variable quelconque en booléen...

Remarque 16

Un test mathématique, ou d'égalité, ou d'identité, renvoie un booléen. Par exemple, `5 < 9` renvoie un booléen (`True` en l'occurrence). Ainsi, écrire `var = 5 < 9` affecte à la variable `var` la valeur `True`. On peut mettre des parenthèses pour améliorer la lisibilité (exemple : `var = (5 > 9)` – ici, `var` prend la valeur `False` puisque 5 n'est pas plus grand que 9), mais ce n'est pas nécessaire.

3 – Entrer du texte en Python

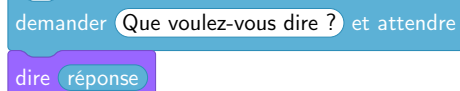
En Python, la fonction permettant d'entrer du texte est `input`.

Exemple 17

Langage naturel

```
1 Afficher "Que voulez-vous dire ?"
2 entree ← ...
3 Afficher entree
```

Scratch



Python

```
1 entree = input("Que voulez-vous dire ?")
2 print(entree)
```

La fonction `input` renvoie une chaîne de caractères, autrement dit du texte (même si on entre un nombre!).

Remarque 18

La fonction `input` renvoie toujours une chaîne de caractères.

Par exemple, si vous demandez à l'utilisateur d'entrer un nombre avec `input`, et que celui-ci entre 2, la fonction `input` renvoie "2", de type `str`, et non 2, de type `int`.

Cela est d'autant plus problématique quand on veut en exploiter le résultat : par exemple, le test `"2" > 1` ne renvoie ni vrai ni faux, il est invalide et soulève un erreur.

On doit donc convertir le résultat de `input` dans le type attendu. Par exemple, si on attend de l'utilisateur qu'il rentre un entier, on peut écrire :

```
x = input("Entrez un nombre")
x = int(x)    # Convertir en int
```

Ou, de manière plus compacte :

```
x = int(input("Entrez un nombre"))
```

4 – (Bonus) Liste des mots-clés en Python

Il existe 35 *mots-clés stricts* ; on ne peut pas nommer une variable ou une fonction avec ces noms, ils sont réservés au fonctionnement de Python.

- | | | | | |
|-------------------------|------------------------|-----------------------|-------------------------|----------------------|
| ▸ <code>and</code> | ▸ <code>del</code> | ▸ <code>from</code> | ▸ <code>None</code> | ▸ <code>True</code> |
| ▸ <code>as</code> | ▸ <code>elif</code> | ▸ <code>global</code> | ▸ <code>nonlocal</code> | ▸ <code>try</code> |
| ▸ <code>assert</code> | ▸ <code>else</code> | ▸ <code>if</code> | ▸ <code>not</code> | ▸ <code>while</code> |
| ▸ <code>break</code> | ▸ <code>except</code> | ▸ <code>import</code> | ▸ <code>or</code> | ▸ <code>with</code> |
| ▸ <code>class</code> | ▸ <code>False</code> | ▸ <code>in</code> | ▸ <code>pass</code> | ▸ <code>yield</code> |
| ▸ <code>continue</code> | ▸ <code>finally</code> | ▸ <code>is</code> | ▸ <code>raise</code> | |
| ▸ <code>def</code> | ▸ <code>for</code> | ▸ <code>lambda</code> | ▸ <code>return</code> | |

Il existe en plus 3 *mots-clés souples*, qui correspondent à des fonctionnalités de Python, mais on peut tout de même nommer des variables ou des fonctions avec ces noms :

- | | | |
|---------------------|----------------------|---------------------|
| ▸ <code>case</code> | ▸ <code>match</code> | ▸ <code>type</code> |
|---------------------|----------------------|---------------------|

Exercices

Exercice 19

Recopier et compléter le tableau d'avancement suivant :

Instruction	Valeur de a	Valeur de b
a = 1		
b = 5		
a = b-3		
b = 2*a		
a = b		
b = a		

Exercice 20

La Terre a un rayon d'environ 6 400km. Compléter les instructions Python suivantes pour que la variable `equateur` contienne la longueur de l'équateur de la Terre en km. La valeur de π est supposée stockée dans une variable appelée `pi`.

```
rayonTerre = ...
equateur = ...
```

Exercice 21

Compléter le tableau suivant :

Exemple de variable	Type (en français)	Type (en Python)
2	Entier	<code>int</code>
		<code>str</code>
		<code>bool</code>
3.14		
"c"		
	Chaîne de caractères	
	Flottant	

Exercice 22

Compléter le tableau suivant :

Opération	Résultat	Type du résultat
$3 + 19$		
$2.5 + 3.1$		
$2.5 + 5$		
"SNT" + "NSI"		
$17//5$		
$15/5$		

Exercice 23

Traduire l'expression $x^3 - 4x^2 + \frac{1}{3}x - 5$ en Python.

Exercice 24

Que vaut la variable x à la fin de ces instructions ?

```
x = 5
x = x*2
x = x - 1
x = x**2
```

4 Python : fonctions et modules

1 – Les fonctions

Une fonction en Python se construit avec le mot-clé `def`, et éventuellement renvoie une valeur avec le mot-clé `return` :

Exemple 1

Langage naturel

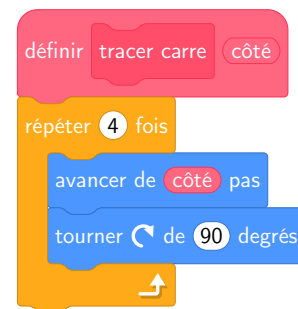
```
1 FONCTION carre(x):
2   RENVOYER x*x
3 FIN FONCTION
4
5 carre(2)
```

Python

```
1 def carre(x):
2     return x*x
3
4 carre(2)
```

Remarque 2

On peut créer des fonctions (appelés *blocs*) avec Scratch, mais celles-ci ne peuvent pas renvoyer de valeur.



Exemple 3

Une fonction peut prendre plusieurs argument. Une fonction peut également renvoyer plusieurs valeurs. Elles apparaîtront alors entre parenthèses. On peut récupérer séparément ces valeurs dans plusieurs variables, comme dans l'exemple suivant :

```
>>> def vitesse(distance, temps):
...     return distance/temps
...
>>> vitesse(60,2)
30.0
>>> vitesse(100,8)
12.5
```

```
>>> def rectangle(a, b):
...     perimetre = 2*a + 2*b
...     aire = a*b
...     return perimetre, aire
...
>>> rectangle(4, 5)
(18, 20)
>>> peri, aire = rectangle(4, 5)
>>> peri
18
>>> aire
20
```

```
def fonction():
    for i in range(5):
        print(i)
        if i == 2:
            return "Sorti !"
```

Une chose importante à noter : le mot-clé `return` arrête la fonction ! Par exemple, le code à gauche contient une fonction qui lance une boucle censée faire cinq itération, afficher le compteur à chaque itération, et renvoyer "Sorti !" à l'itération `i = 2`.

Cependant, si vous appelez cette fonction, vous constaterez que les itérations `i = 3` et `i = 4` sont ignorées.

```
>>> f()
0
1
2
'Sorti !'
```

Une fonction s'arrête au premier `return` qu'elle rencontre. Tout code qui suivrait un `return` serait ignoré.

Remarque 4

Observez le code suivant : selon-vous, que va-t-il produire ?

```

1  a = 4
2
3  def f():
4      b = 9
5      print(a)
6
7  print(b)
```

Remarquez que `a` est définie à l'extérieur de `f`, et `b` est définie à l'intérieur de `f`. En exécutant ce code, `f` va réussir à afficher la valeur de `a` même s'il elle est définie à l'extérieur, mais `print(b)` va renvoyer une erreur, selon laquelle la variable `b` n'est pas définie. En effet, c'est ce qu'on appelle la *localité des variables*. Les variables peuvent passer des environnement les plus grands aux plus petits, mais pas faire le chemin inverse.

Remarque 5 (Pour les experts)

Lorsqu'on précise des arguments à une fonction comme on l'a fait jusque là, on parle d'*arguments positionnels*, car leur position dans les parenthèses détermine leur interprétation. Par exemple, dans l'exemple 3, `dimensions(18,20)` et `dimensions(20, 18)` ne renvoient pas la même chose.

On peut donner une valeur par défaut à certains arguments : on parle alors d'*arguments nommés*. Lors de l'appel de la fonction, on peut alors ne pas préciser la valeur de ces arguments si on n'en a pas besoin.

```

>>> def interets(capital, annee_fin, annee_debut=2000, taux=1.1):
...     temps = annee_fin - annee_debut
...     for _ in range(temps):
...         capital *= taux
...     return capital
...
>>> interets(1000, 2010)
2593.7424601000025
>>> interets(1000, 2010, annee_debut=2009)
1100.0
>>> interets(1000, 2010, taux=2)
1024000
>>> interets(1000, 2010, annee_debut=2009, taux=2)
2000
```

Attention, les arguments nommés doivent être placés après les arguments positionnels.

Remarque 6 (Pour les experts)

Lorsque vous tapez, par exemple, `help(sqrt)`, s'affiche à l'écran une aide, une description de ce que fait la fonction `sqrt`, ce qu'elle prend en entrée et renvoie en sortie.

Ce que s'affiche, c'est ce qu'on appelle la *documentation* de la fonction. Dans les gros projets, il est souvent vital de rédiger la documentation de vos propres fonctions : en effet, si vous travaillez à plusieurs, ou sur une longue période de temps, vous serez amenés à relire des fonctions qui vous sembleront inconnues. Plutôt que de devoir réfléchir et prendre le temps de comprendre le code que vous avez sous les yeux, il sera bien plus efficace de lire la documentation directement.

Pour insérer une documentation, faites comme dans l'exemple suivant :

```

1  def dimensions(perimetre, aire):
2      """
3      À partir de son périmètre et de son aire, retrouver la largeur et la
4      longueur d'un rectangle.
5      Entrée:
6          - perimetre (float): le périmètre du rectangle
7          - aire (float): l'aire du rectangle, dans l'unité au carré du
8                      périmètre !
9      Sortie:
10         - Largeur du rectangle (float)
11         - Longueur du rectangle (float)
12     """
13
14     d = sqrt(perimetre**2 - 16*aire)
15     return (perimetre - d)/4, (perimetre + d)/4

```

(Remarque : ici, la documentation n'était pas vraiment nécessaire, car les noms de la fonction et des variables étaient déjà transparents.)

Remarque 7 (Pour les experts)

Le *type* (ou *type hints*) a été introduit dans la version 3.5 de Python. C'est un complément de la documentation, qui la rend encore plus transparente en permettant de préciser la *signature* de la fonction, c'est-à-dire les types qu'elle reçoit en entrée et ceux en sortie.

```

1  def ecart_longueur(chaine1: str, chaine2: str) -> int:
2      """ Calcule l'écart de longueur entre deux chaînes de caractères. """
3      ecart = len(chaine1) - len(chaine2)
4      if ecart < 0:
5          ecart = -ecart
6      return ecart

```

(Remarque : si la fonction renvoie plusieurs valeurs, alors il faut écrire -> *tuple*.)

2 – Les modules Python

Bien que Python vient initialement avec une très large gamme de fonctions prédéfinies, certaines fonctions plus spécifiques (par exemple, les fonction mathématiques, les fonctions aléatoires, ...) ne sont pas disponibles de base : il faut les *importer* à partir d'un *module*.

A – Les modules mathématiques

Le module `math` contient toute une variété de fonctions mathématiques : `cos`, `sin`, `tan`, `sqrt` (la racine carrée, *square root* en anglais), ... mais aussi des variables spéciales : par exemple, le module `math` contient la variable `pi` qui, comme son nom le suggère, contient la valeur de π .

Pour importer un module, la syntaxe est la suivante : `from math import *`.

Remarque 8 (Pour les experts)

Cette syntaxe pour l'importation d'un module est à éviter au possible. En effet, prenons un exemple : imaginez que vous avez défini une fonction `cos` dans votre programme, puis que vous importez le module `math`. Comme

`math` contient lui-même une fonction `cos`, cette-dernière va simplement supprimer la vôtre et la remplacer.

Une meilleure syntaxe est la suivante : `from math import fonction_1, fonction_2`

On précise les fonctions qu'on souhaite importer, et on importe uniquement celles-ci.

La meilleure syntaxe est la suivante : `import math`

Grâce à cette syntaxe, vous pourrez avoir, par exemple, plusieurs fonctions qui s'appellent `cos`. En effet, pour appeler une fonction du module `math`, la syntaxe est de la forme `math.fonction`. Par exemple, on écrirait `math.cos(math.pi / 2)`.

Pour connaître la liste des fonctions des modules `math` et `random`, exécutez `help(math)` et `help(random)`. Voici tout de même quelques fonction incontournables :

Module	Fonction(s)	Description
<code>random</code>	<code>randint</code>	Nombre aléatoire entre deux bornes (incluses) <code>randint(1,6)</code> renvoie un nombre aléatoire entre 1 et 6
<code>math</code>	<code>sqrt</code>	Racine carrée
<code>math</code>	<code>cos</code> , <code>sin</code> , <code>tan(x)</code>	Fonctions trigonométriques (angles en degrés)
<code>math</code>	<code>floor</code>	Partie entière
/	<code>abs</code>	Valeur absolue
/	<code>round</code>	Arrondi à 10^{-n} <code>round(x,n)</code> arrondit x avec n décimales
/	<code>min</code> , <code>max</code>	Minimum / maximum de deux valeurs

Le module `math` fournit également `pi`.

B – Des modules graphiques – exemple de `turtle`

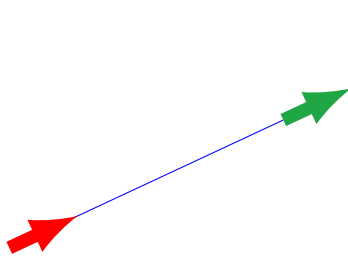
Il existe de nombreux modules permettant l'affichage d'éléments graphiques, parmi eux on peut noter `matplotlib` qui permet la génération de graphiques mathématiques (courbes de fonctions, histogrammes, nuages de points, ...). Dans ce paragraphe, nous allons plutôt nous concentrer sur `turtle`, un module qui permet de faire du dessin de manière assez similaire à Scratch.

Dans ce paragraphe, on liste une poignée de fonctions utiles, avec une petite explication. Il est conseillé de ne pas simplement lire ce morceau de cours, mais plutôt choisir un projet de dessin et consulter la liste des fonctions afin de compléter le projet. L'utilisation du module `turtle` n'est pas un objectif du programme, elle est une application ludique de Python.

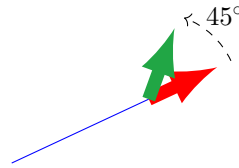
Toute utilisation du module `turtle` débutera par son import, et *dans Basthon*, tout dessin doit se terminer par `done()` pour lancer le tracé.

```
from turtle import *
...
done()
```

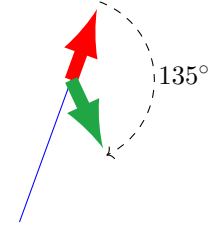
Déplacement et orientation



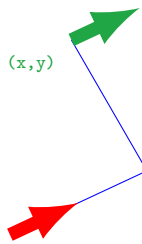
`forward(n) / backward(n)`
Avancer/reculer de n pixels



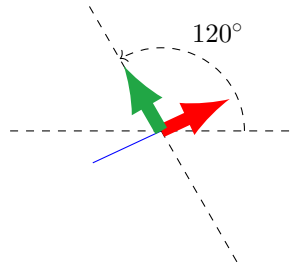
`left(45)`
Tourner de 45° vers la gauche *par rapport à là où la tortue fait face*



`right(135)`
Tourner de 135° vers la droite *par rapport à là où la tortue fait face*



`goto((x,y))`
Déplacer la tortue aux coordonnées (x,y) *sans la tourner*



`setheading(120)`
Tourner la tortue face à la direction 120° *par rapport à l'horizontale* (à droite = 0°)

`clear()`
Efface tous les tracés

Modification du style

`up() / down()`
Relève le stylo (la tortue ne trace plus) / Abaisse le stylo (la tortue trace)

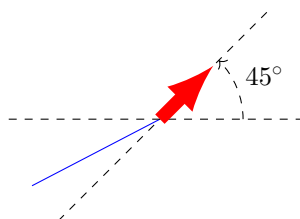
`color("red")`
Change la couleur du tracé

`width(n)`
Change l'épaisseur du trait à n pixels

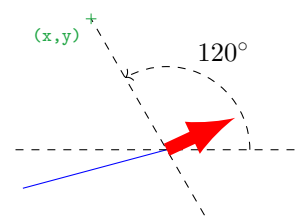
Fonctions de mesure

`position()`
Renvoie les coordonnées (x,y) actuelles de la tortue

`distance(x,y)`
Renvoie la distance en pixels entre la position actuelle de la tortue et le point de coordonnées (x,y)



`heading()`
Renvoie une mesure de l'angle auquel fait face la tortue par rapport à l'horizontale



`towards(x,y)`
Renvoie une mesure de l'angle de la droite horizontale passant par la tortue à la droite passant par la tortue et le point de coordonnées x,y

Remplir une région

Pour remplir une région, écrivez `begin_fill()` puis tracez le contour de cette région. Pour arrêter le mode remplissage, écrivez `end_fill()`. Pour changer la couleur de remplissage, écrivez `fillcolor("red")` pour, par exemple, remplir en rouge.

3 – (Bonus) Apprendre à lire les erreurs

Voici un exemple d'erreur :

```
>>> a = 3
>>> b = "Bonjour"
>>> a + b
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

Une erreur se décompose en trois parties :

- ▶ « Traceback (most recent call last) : » indique qu'une erreur est survenue. Nous y reviendrons dans l'exemple suivant.
- ▶ Ensuite, Python indique le lieu de l'erreur, en particulier la ligne dans le code. C'est le plus utile quand on stocke le code dans un fichier, plutôt que l'écrire dans la console. (Remarque : « File : » indique le nom du fichier, et ici « <stdin> » désigne la console.)
- ▶ Le type d'erreur et une description de la raison. Ici, on a essayé d'ajouter un nombre et une chaîne de caractères.

Voici un exemple plus complexe :

```
>>> def f_division(a,b):
...     return a/b
...
>>> f_division(5,0)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "<stdin>", line 2, in f_division
ZeroDivisionError: division by zero
```

Ici, l'erreur a lieu à l'intérieur d'une fonction. Python va retracer le fil de son exécution : en effet, il nous indique une division par zéro dans la fonction. Mais le problème ne vient pas de la fonction, il vient des arguments qu'on lui a donnés ! Pour nous aider à les retrouver, Python nous indique aussi à quelle ligne on a appelé la fonction avec les mauvais arguments. La phrase « Traceback (most recent call last) : » signifie donc qu'on remonte nos pas, avec en dernière ligne, l'opération de la fonction qui a échoué, en avant-dernière ligne, l'endroit où on a appelé cette fonction, éventuellement, avant encore, l'endroit où on appelé la fonction qui a appelé la fonction, etc. Python refait l'historique des événements qui ont conduit à l'erreur.

Exercices

Exercice 9

Traduire les programmes suivants dans la langage Python :

```
1  FONCTION f(x):
2      RENDRE x**2 - 3*x + 1
3  FIN FONCTION
```

```
1  FONCTION moyenne(a,b,c):
2      S ← a + b + c
3      moy ← S/3
4      RENDRE moy
5  FIN FONCTION
```

a et b en fonction de x puis démontrer la conjecture.

Instruction	a	b
$a = x$		
$b = x$		
$a = a - 5$		
$a = a*a$		
$a = a - 13$		
$b = b*(b-10)$		
$b = b + 12$		

Exercice 10

On considère la fonction :

```
1  def joliot():
2      a = 1
3      b = 5
4      a = b - 3
5      b = 2*a
6      a = b
7      b = a
8      return a, b
```

Déterminer les valeurs renvoyées par cette fonction.

Exercice 12

On considère la fonction :

```
def produits(a, b, c):
    return (a*c, b*c)
```

- 1► Quel est le nom de cette fonction ?
- 2► Combien d'arguments possède cette fonction ?
- 3► Que renvoie l'instruction `produits(4,3,2)` ?
- 4► Écrire une instruction qui appelle la fonction `produits` avec des arguments bien choisis de façon à ce que les valeurs renvoyées soient (15,20).

Exercice 11

On considère la fonction :

```
def curie(x):
    a = x
    b = x
    a = a - 5
    a = a*a
    a = a - 13
    b = b*(b-10)
    b = b + 12
    return a, b
```

- 1► Déterminer les valeurs renvoyées par l'appel `curie(x)` pour :
 - a► $x = 3$
 - b► $x = 12.5$
 - c► $x = 5$
- 2► Que peut-on conjecturer ?
- 3► Compléter le tableau ci-dessous en exprimant

Exercice 13

- 1► Écrire une fonction `peri_carre` qui prend en argument la longueur du côté d'un carré et renvoie son périmètre.
- 2► Faire un appel de cette fonction afin d'afficher le périmètre d'un carré de côté 10.

Exercice 14

- 1► Écrire une fonction `calcul` qui traduit le programme de calcul suivant : on choisit un nombre, on lui ajoute 5, et on double le résultat.
- 2► Faire un appel de cette fonction afin d'afficher le résultat pour la valeur 10 au départ.

Exercice 15

Écrire une fonction `pythagore` qui prend comme arguments les longueurs des deux côtés perpendiculaires d'un triangle rectangle, et renvoie la longueur de son hypoténuse.

On utilisera la fonction `sqrt` du module `math`.

Exercice 16

Écrire une fonction `disque` qui prend comme argument le rayon d'un disque et renvoie son aire.

On utilisera la constante `pi` fournie par le module `math`.

Exercice 17

Écrire une fonction `des` qui ne prend aucune argument et renvoie la somme du lancer de deux dés à 6 faces.

On utilisera la fonction `randint` du module `random`.

Exercice 18

Réalisez les figures des exercices 47, 48 et 49 du chapitre d'algorithmique à l'aide du module `turtle`.

5 Python : boucles et conditionnement

1 – Les structures itératives

A – La boucle inconditionnelle (ou bornée) : la boucle *for*

En Python, la boucle inconditionnelle (ou bornée) se construit à partir des mots-clés *for* et *in* :

Exemple 1

Langage naturel

```
1 POUR n ALLANT DE 1 A 10:  
2   AFFICHER 2 * n  
3 FIN POUR
```

Python

```
1 for n in range(1, 11):  
2     print(2*n)
```

Scratch



Remarque 2

- Python n'a pas d'équivalent de FIN POUR : la boucle s'arrête là où l'indentation s'arrête. C'est pourquoi vous devez prêter le plus grand soin à vos indentations !
- Remarquez que pour dire que le compteur *n* doit varier entre 1 et 10, on utilise la syntaxe `range(1,11)` : en particulier, la première borne est incluse, et la seconde est exclue !
- Constatez qu'avec Scratch, le compteur n'est pas automatiquement créé par la boucle ! Il faut le définir et l'incrémenter à la main.

Remarque 3

Quand on se fiche du compteur et qu'on souhaite juste répéter un bloc d'instructions un certain nombre de fois (comme Scratch le fait par défaut), on peut écrire `for i in range(20)` : pour répéter les instructions 20 fois. (On peut même écrire `for _ in range(20)` : pour signifier que l'on n'a pas besoin d'utiliser le compteur.)

Remarque 4 (Pour les experts)

- Lorsque la première borne n'est pas précisée, elle est implicitement définie à 0. Autrement dit, `range(15)` est un raccourci pour `range(0,15)`.
- La fonction `range` prend un troisième paramètre qui correspond au pas (de deux en deux, trois en trois, voire -1 pour indiquer un parcours décroissant). Exemples : `range(1,10,2)` donne 1, 3, 5, 7, 9 ; `range(5,0,-1)` donne 5, 4, 3, 2, 1.

B – La boucle conditionnelle (ou non bornée) : la boucle *while*

En Python, la boucle conditionnelle (ou non bornée) se construit à partir du mot-clé *while* :

Exemple 5

Langage naturel	Python	Scratch
<pre> 1 x ← 0 2 TANT QUE 2*x ≤ 10: 3 x ← x + 2 4 FIN TANT QUE 5 AFFICHER x </pre>	<pre> 1 x = 0 2 while 2*x <= 10: 3 x = x + 2 4 print(x) </pre>	

Remarque 6

Notez la différence entre la boucle conditionnelle de Scratch et elle de Python. En Scratch, on boucle *jusqu'à ce que* la condition soit vraie, autrement dit *tant que* la condition est fausse. En Python, on boucle *tant que* la condition est vraie. Ainsi, la condition de la boucle non bornée est inversée entre Python et Scratch !

2 – Les structures conditionnelles

A – Les opérateurs logiques

Python vient avec son lot d'opérateurs de comparaison. Ce sont des opérateurs *mathématiques* : ainsi, Python considère que 2 et 2.0 ont la même valeur (mathématique), bien qu'ils soient de types différents.

Opération	Notation	Exemple 1	Exemple 2
Égal à	==	>>> 2 == 2.0 True	>>> 3 == 15 False
Différent de	!=	>>> 2 != 2.0 False	>>> 3 != 15 True
Inférieur (strict) à	<	>>> 2 < 2.0 False	>>> 3 < 15 True
Inférieur ou égal à	<=	>>> 2 <= 2.0 True	>>> 3 <= 15 False
Supérieur (strict) à	>	>>> 2 > 2.0 False	>>> 3 > 15 False
Supérieur ou égal à	>=	>>> 2 >= 2.0 True	>>> 3 >= 15 False

On peut également tester l'égalité ou la différence de deux chaînes de caractères :

Opération	Notation	Exemple 1	Exemple 2
Égal à	==	>>> "Python" == "Python" True	>>> "Fourchette" == "Couteau" False
Différent de	!=	>>> "Python" != "Python" False	>>> "Fourchette" != "Couteau" True

Remarque 7

Les opérateurs `<`, `<=`, `>`, `>=` fonctionnent aussi avec les chaînes de caractères : ils utilisent l'ordre *lexicographique* (ce qui signifie l'ordre alphabétique, étendu aux chiffres et aux symboles, et en distinguant les majuscules des minuscules).

B – Les mots-clés `if`, `elif` et `else`

La structure conditionnelle la plus simple se construit avec le mot clé `if` :

Exemple 8

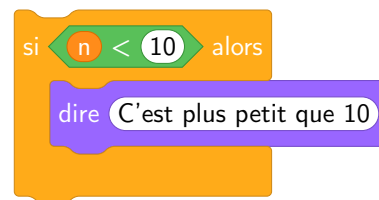
Langage naturel

```
1  SI n < 10:
2    AFFICHER "C'est plus petit que 10"
3  FIN SI
```

Python

```
1  if n < 10:
2    print("C'est plus petit que 10")
```

Scratch

**Remarque 9**

En Python, l'indentation seule définit un bloc de commandes ; celui-ci se termine là où l'indentation s'arrête. Il n'y a pas d'équivalent à `Fin Si` en Python. N'oubliez pas les `:` à la fin de la condition !

La structure conditionnelle avec alternative se construit avec le mot-clé `else` :

Exemple 10

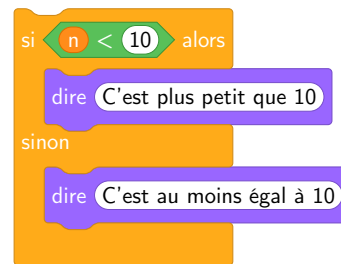
Langage naturel

```
1  SI n < 10:
2    AFFICHER "C'est plus petit que 10"
3  SINON:
4    AFFICHER "C'est au moins égal à 10"
5  FIN SI
```

Python

```
1  if n < 10:
2    print("C'est plus petit que 10")
3  else:
4    print("C'est au moins égal à 10")
```

Scratch



La structure conditionnelle avec plusieurs tests se construit avec le mot-clé `elif` :

Exemple 11

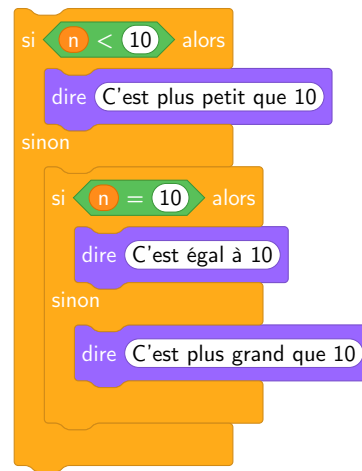
Langage naturel

```
1 SI n < 10:  
2   AFFICHER "C'est plus petit que 10"  
3 SINON SI n = 10:  
4   AFFICHER "C'est égal à 10"  
5 SINON:  
6   AFFICHER "C'est plus grand que 10"  
7 FIN SI
```

Python

```
1 if n < 10:  
2     print("C'est plus petit que 10")  
3 elif: n == 10:  
4     print("C'est égal à 10")  
5 else:  
6     print("C'est plus grand que 10")
```

Scratch



Exercices

Boucle bornée

Exercice 12

Les instructions `avancer()`, `droite()`, ... s'écrivent de la même manière en Python et en langage naturel. Traduire les programmes suivants en langage Python :

```
1 droite()
2 POUR i ALLANT DE 1 A 5:
3     avancer()
4 FIN POUR
```

```
1 gauche()
2 POUR k ALLANT DE 1 A 12:
3     sauter()
4     avancer()
5     coup()
6 FIN POUR
7 ouvrir()
```

```
1 POUR i ALLANT DE 1 A 5:
2     POUR j ALLANT DE 1 A 6:
3         avancer()
4     FIN POUR
5     tourner()
6 FIN POUR
```

Exercice 13

Pour chaque question, choisir la ou les bonnes réponses.

- 1► Dans la boucle `for k in range(2,7)`, la variable `k` prend les valeurs :
 - a► 2 et 7
 - b► 2, 3, 4, 5 et 6
 - c► 2, 3, 4, 5, 6 et 7
 - d► 2, 3, 4, 5, 6, 7 et 8
- 2► Dans la boucle `for k in range(5)`, la variable `k` prend les valeurs :
 - a► 0, 1, 2, 3 et 4
 - b► 0, 1, 2, 3, 4 et 5
 - c► 1, 2, 3 et 4
 - d► 1, 2, 3, 4 et 5
- 3► Lorsqu'on veut qu'une variable `k` prenne successivement toutes les valeurs entières de 0 à 33, on peut utiliser l'instruction `for k in ...` :

- a► `range(0,33)`
- b► `range(0,34)`
- c► `range(33)`
- d► `range(34)`

- 4► Dans la boucle `for k in range(2,29)`, combien `k` prend-elle de valeurs ?

- a► 29
- b► 28
- c► 27
- d► 26

Exercice 14

Déterminer la valeur de la variable `d` après l'exécution du programme dans chaque cas ci-dessous.

```
1 a, b, c, d = 1, 1, 1, 0
2 for k in range(2):
3     a = a + b
4     b = b + a
5     c = c + b
6 d = d + c
```

```
1 a, b, c, d = 1, 1, 1, 0
2 for k in range(2):
3     a = a + b
4     b = b + a
5 c = c + b
6 d = d + c
```

```
1 a, b, c, d = 1, 1, 1, 0
2 for k in range(2):
3     a = a + b
4 b = b + a
5 c = c + b
6 d = d + c
```

Exercice 15

Alice verse 20 000€ sur son livret d'épargne. Celui-ci lui rapporte 100€ par an. Écrire une fonction `capital` qui prend comme argument une variable `n`, et renvoie le capital d'Alice au bout de `n` années. On pourra s'appuyer sur le squelette d'algorithme suivant :

```
1 C ← ...
2 POUR i ALLANT DE 1 A ...:
3     C ←
4 FIN POUR
```

Exercice 16

Bob s'entraîne pour une course de vélo. Il établit un programme sur 12 semaines : la première il parcourt 25km, et augmente la distance de 11km chaque semaine.

- 1▶ Déterminer la distance D parcourue la deuxième semaine, et la distance totale cumulée T parcourue au bout de 2 semaines.
 - 2▶ Écrire une fonction distance qui ne prend aucun argument et renvoie la distance totale cumulée parcourue au bout de 12 semaines.
- On s'appuiera sur le squelette d'algorithme suivant :

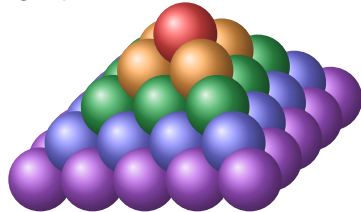
```

1  D ← ...
2  T ← ...
3  POUR i ALLANT DE 2 A 12:
4      D ← ...
5      T ← ...
6  FIN POUR

```

Problème 17

On empile des boules, formant une pyramide à base carrée comme illustré ci-dessous. On a coloré les différents étages pour une meilleure lisibilité.



- 1▶ Combien de boules faut-il pour faire une pyramide à 2 étages ?
- 2▶ Combien de boules faut-il pour faire une pyramide à 5 étages ?
- 3▶ Sachant qu'il faut 140 boules pour faire une pyramide à 7 étages, combien de boules faut-il pour faire une pyramide à 8 étages ?
- 4▶ Compléter la fonction ci-dessous qui permet de calculer le nombre de boules nécessaires pour construire une pyramide à n étages.

```

1  def nb_boules(...):
2      ...
3      for i in range(1,...):
4          ...
5      return ...

```

Boucle non bornée**Exercice 18**

Les instructions avancer(), droite(), ... s'écrivent de la même manière en Python et en langage naturel. Traduire les programmes suivants en langage Python :

```

1  TANT QUE peut_avancer():
2      avancer()
3  FIN TANT QUE

```

```

1  TANT QUE obstacle_devant():
2      coup()
3      sauter()
4      avancer()
5  FIN TANT QUE
6  avancer()
7  ouvrir()

```

```

1  TANT QUE cible_devant():
2      POUR j ALLANT DE 1 A 6:
3          tirer()
4      FIN POUR
5      tourner()
6  FIN TANT QUE

```

Exercice 19

On donne trois version d'une fonction calc :

```

1  def calc(a,b):
2      n = 0
3      while n < 20:
4          n = a + b
5          a = 2*a
6          b = b*3
7          n = n*b
8      return n

```

```

1  def calc(a,b):
2      n = 0
3      while n < 20:
4          n = a + b
5          a = 2*a
6          b = b*3
7      n = n*b
8      return n

```

```

1 def calc(a,b):
2     n = 0
3     while n < 20:
4         n = a + b
5         a = 2*a
6         b = b*3
7         n = n*b
8     return n

```

Pour chaque version de la fonction, déterminer le renvoi de `calc(5,1)`. On dressera un tableau d'avancement dans chaque cas.

Exercice 22

Dans une reprographie, le coût de l'impression en fonction du nombre de pages est donné par l'algorithme suivant :

```

1 if nbPages <= 50:
2     prix = 0.15*nbPages
3 else:
4     prix = 0.10*nbPages

```

Combien coûte l'impression d'un document de 25 pages ? De 75 pages ? De 50 pages ?

Structure conditionnelle

Exercice 20

Les commandes `avancer()`, `droite()`, ... s'écrivent de la même manière en Python et en langage naturel. Traduire les programmes suivants en langage Python :

```

1 SI obstacle_devant():
2     coup()
3 SINON:
4     avancer()
5 FIN SI

```

```

1 SI hauteur > 2:
2     tourner()
3 SINON SI hauteur = 2:
4     sauter_haut()
5 SINON:
6     sauter()
7 FIN SI

```

Exercice 21 (Déclic 2022)

On considère la fonction `mystere` ci-contre.

- 1► Quels sont les paramètres de cette fonction ?
- 2► Quels sont renvois des appels suivants ?

a► `mystere(3,4)` b► `mystere(12,5)`

```

1 def mystere(x, y):
2     if x < 10:
3         return 3*x - y
4     else:
5         return x + y

```

Exercice 23

Déterminer le renvoi le fonction `abs` ci-dessous pour $x = -2$, $x = 3$, $x = -5.8$ et $x = 0$.

```

1 def abs(x):
2     if x >= 0:
3         return x
4     else:
5         return -x

```

Exercice 24

Écrire une fonction `mini` qui renvoie le plus petit de ses deux arguments.

Exercice 25

Un manège fait payer son entrée en fonction de la taille et l'âge du client, selon l'algorithme suivant :

```

1 def prix(taille, age):
2     if taille <= 90:
3         entree = 0
4     else:
5         if 90 <= taille < 130:
6             entree = age*0.5
7         else:
8             entree = age*1.1
9     return entree

```

Combien doit payer un client de 18 ans mesurant 1,82m ? Et un enfant de 8 ans mesurant 1,20m ?

Exercice 26

```

1  def comp(a,b):
2      if a < b:
3          c = 0
4      else:
5          if b < a:
6              c = 1
7          else:
8              c = 2
9      return c

```

- 1► Déterminer le renvoi de comp dans les cas suivants :
 - a► a = 6 et b = -3
 - b► a = 4 et b = 7
 - c► a = -7.12 et b = -7.1
 - d► a = 29/6 et b = 82/17
- 2► Dans quel(s) cas la fonction comp renvoie-t-elle 2 ?

Exercice 27

Jouez au jeu PyRates disponible en ligne à l'adresse <https://py-rates.fr>.

Problème 28 (Suite de Syracuse)

- 1► Réaliser l'exercice 40 du chapitre d'algorithmique en langage Python, dans une fonction syracuse.
- 2► Modifier cette fonction pour qu'elle renvoie le nombre de tours de boucle avant que la suite n'arrive à 1.
- 3► Que fait l'algorithme suivant ?

```

1  n = 1
2  while syracuse(n) <= 100:
3      n + n + 1

```

Problème 29

Un camping propose deux formules de location :

- Réservation à 250€ + location à 10€ par jour
- Réservation à 300€ + location à 5€ par jour

Parmi les fonctions suivantes, laquelle ou lesquelles permet(tent) de déterminer la formule la plus rentable en fonction du nombre de jours ? Justifier en expliquant chaque algorithme en détail (ligne par ligne).

```

1  def algo1(n):
2      A = 250 + 10*n
3      B = 300 + 5*n
4      if A < B:
5          return "B"
6      else:
7          return "A"

```

```

1  def algo2(n):
2      A = 250 + 10*n
3      B = 300 + 5*n
4      if A < B:
5          return "A"
6      else:
7          return "B"

```

```

1  def algo3(n):
2      D = 5*n - 50
3      if D > 0:
4          return "B"
5      else:
6          return "A"

```

6 Python : les listes

7 Les langages du web

Sciences Numériques & Technologies

Seconde GT

Cet ouvrage regroupe tous les cours et exercices dispensés en SNT (sciences numériques et technologies) lors de l'année de seconde générale et technologique. Les chapitres sont ordonnés selon une progression donnée, qui n'est pas la seule envisageable.

Il existe deux versions de ce manuel : une version dite « étendue » (la présente) et une version « raccourcie à compléter ».

- ▶ La version « étendue » est enrichie de remarques et de détails supplémentaires et d'approfondissements pour les esprits curieux et avides de connaissances pointues, et qui a également vocation à servir de support pour les enseignants qui préparent leurs cours de SNT ;
- ▶ La version « raccourcie à compléter » est, comme son nom l'indique, une extraction de la version « étendue », qui ne contient que les essentiels du cours, avec des cadres à compléter, et les exercices. Cet élagage a deux objectifs : le premier étant de créer un support de cours optimisé pour la prise de notes en classe pour laisser plus de temps à la compréhension et aux exercices ; le second est d'économiser du papier, car celui-ci ne pousse pas sur les arbres ! Enfin... façon de parler.

Je remercie chaleureusement toutes celles et ceux qui me font part de fautes de frappes, d'erreurs, ou même de leurs avis sur certains passages, et qui me permettent de proposer un cours de la meilleure qualité possible.

Cours & illustrations par E. Suong

Les ressources sont issues de sources multiples. Pour tout ajout de référence, adaptation ou retrait, merci de me contacter.

Des outils d'intelligence artificielle ont assisté la conception de ce manuel.

