

NOM :
Prénom :
Classe :

Lycée Jeanne d'Arc
Clermont-Ferrand
2025 – 2026

Fascicule de SNT (Seconde)

CHAPITRE I : Les systèmes de numération.....2	TP 6 : Une exploitation forestière.....19
LEÇON 1 : Le système décimal (rappels).....2	TP 7 : La durée de vie des mammifères.....20
<i>Exercice 1 : Décomposer un nombre.....2</i>	CHAPITRE IV : Le langage Python.....21
<i>Exercice 2 : Expliquer sans calculer.....3</i>	LEÇON 8 : Découverte de Python.....21
LEÇON 2 : Découverte du système binaire.....3	<i>Exercice 21 : Utiliser des variables dans la console.....22</i>
<i>Exercice 3 : Calculer la valeur d'un nombre binaire.....3</i>	LEÇON 9 : Utiliser des fonctions en Python.....23
<i>Exercice 4 : Énumérer les nombres binaires.....3</i>	<i>Exercice 22 : Utiliser des fonctions dans la console.....24</i>
LEÇON 3 : Convertir vers le système binaire.....4	<i>Exercice 23 : Jouer avec les codes ASCII.....24</i>
<i>Exercice 5 : Convertir en base 2.....4</i>	<i>Exercice 24 : Mon premier programme.....24</i>
<i>Exercice 6 : Manipuler la base 2.....4</i>	<i>Exercice 25 : Convertisseur de prix.....24</i>
LEÇON 4 : Le système hexadécimal.....5	LEÇON 10 : Définir des conditions en Python.....25
<i>Exercice 7 : Énumérer les nombres hexadécimaux.....6</i>	<i>Exercice 26 : Comparer des valeurs.....25</i>
<i>Exercice 8 : Manipuler le système hexadécimal.....6</i>	<i>Exercice 27 : Convertisseur de température.....26</i>
<i>Exercice 9 : Compter comme un Shadok.....6</i>	<i>Exercice 28 : La divisibilité d'un nombre.....26</i>
<i>Exercice 10 : Convertir en temps limité.....6</i>	LEÇON 11 : Faire des boucles en Python.....26
CHAPITRE II : Stocker et coder des informations.....7	<i>Exercice 29 : Je me répète.....27</i>
LEÇON 5 : L'organisation d'un ordinateur.....7	<i>Exercice 30 : Pair ou impair.....27</i>
<i>Exercice 11 : Étudier une arborescence.....9</i>	<i>Exercice 31 : Une séquence d'ADN.....27</i>
<i>Exercice 12 : Mesurer l'information.....9</i>	<i>Exercice 32 : Jouer à deviner un nombre.....27</i>
<i>Exercice 13 : Connaître les extensions courantes.....9</i>	CHAPITRE V : Internet et le Web.....27
<i>Exercice 14 : Rendre sa clé USB retrouvable.....10</i>	LEÇON 12 : Internet.....27
TP 1 : Explorer le système de fichiers.....10	TP 8 : Simulation du protocole TCP/IP.....29
TP 2 : Les métadonnées des photos.....11	LEÇON 13 : Le Web.....29
LEÇON 6 : Le codage des données.....12	TP 9 : Découverte des langages du Web.....31
<i>Exercice 15 : Un proverbe codé.....13</i>	TP 10 : Projet de fin d'année.....31
<i>Exercice 16 : Déchiffrer un texte binaire.....13</i>	Annexe 1 : Les instructions en Python.....31
<i>Exercice 17 : Blague de robot.....13</i>	<i>Les opérations en Python.....31</i>
<i>Exercice 18 : Coloriage RGB.....13</i>	<i>Les conditions en Python.....32</i>
<i>Exercice 19 : Le code Morse.....14</i>	<i>Les autres mots-clés en Python.....32</i>
<i>Exercice 20 : La taille du génome humain.....14</i>	<i>Les fonctions de base en Python.....33</i>
TP 3 : Le codage des textes.....15	<i>Les fonctions de la bibliothèque « random ».....33</i>
TP 4 : Les couleurs des pixels.....16	<i>Les fonctions de la bibliothèque « math ».....33</i>
CHAPITRE III : Traiter des données structurées.....17	Annexe 2 : Les langages du Web.....34
LEÇON 7 : Le format CSV.....17	<i>Les principales balises en HTML.....34</i>
TP 5 : Traiter les données des secours français.....18	<i>Les principales propriétés en CSS.....34</i>

Site Web : <https://evoluscope.fr/>

CHAPITRE I : Les systèmes de numération

LEÇON 1 : Le système décimal (rappels)

Un système de numération est une méthode mathématique permettant d'écrire les nombres entiers en combinant un ensemble limité de symboles représentant des valeurs. Les symboles d'un système de numération s'appellent des chiffres (*digits* en anglais).

Une notation positionnelle est un système de numération dans lequel les valeurs des chiffres varient en fonction de leurs positions dans l'écriture du nombre. La position d'un chiffre s'appelle aussi son rang.

La base d'une notation positionnelle est le nombre de chiffres différents qu'elle contient. Le système à base dix s'appelle le système décimal, il contient 10 chiffres : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Exemples :

- Dans le système décimal le nombre treize s'écrit « 13 » et dans cette écriture le chiffre « 3 » vaut trois. Le nombre trente-cinq s'écrit « 35 » et ici le chiffre « 3 » vaut trente.
- La numération romaine **n'est pas** une notation positionnelle car les valeurs des chiffres ne dépendent pas de leurs positions. Ainsi, « II » signifie deux et pas onze.

Dans une notation positionnelle la valeur d'un nombre est la somme des valeurs des chiffres multipliés par le poids de leur rang. On compte les rangs de droite à gauche en commençant par 0, puis on calcule le poids avec la formule : **poids = base^{rang}**

Exemple :

Décomposition et calcul de la valeur du nombre qui s'écrit « 253 » dans le système décimal.

Chiffre	2	5	3
Rang	2	1	0
Poids	$10^2 = 100$	$10^1 = 10$	$10^0 = 1$
Valeur	$2 \times 100 = 200$	$5 \times 10 = 50$	$3 \times 1 = 3$

Valeur du nombre « 253 » : $200 + 50 + 3$, soit deux-cent-cinquante-trois.

Donc dans le système décimal le chiffre situé tout à droite est toujours multiplié par 1, c'est l'unité, le chiffre suivant est toujours multiplié par 10, c'est la dizaine, le chiffre suivant est toujours multiplié par 100, c'est la centaine, etc.

Exercice 1 : Décomposer un nombre

Décomposez l'écriture du nombre qui s'écrit « 726 » dans le système décimal et donnez sa valeur en toutes lettres.

Chiffre			
Rang			
Poids			
Valeur			

Exercice 2 : Expliquer sans calculer

Expliquez sans calculer pourquoi dans le système décimal les nombres 621 et 162 n'ont pas la valeur alors qu'ils sont composés des trois mêmes chiffres.

LEÇON 2 : Découverte du système binaire

Dans un ordinateur, tout est représenté par des nombres : les textes, les images, les sons, les vidéos, les programmes, etc. Or, les ordinateurs ne peuvent manipuler que deux symboles, le 0 et le 1. On utilise donc un système à base deux, ou système binaire, pour représenter les nombres dans un ordinateur.

Pour ne pas confondre un nombre écrit dans différents systèmes de numération qui ont des symboles en commun, on écrit la base en bas à droite du nombre.

Exemple :

101_{10} est un nombre écrit en base 10 qui vaut cent-un.
 101_2 est un nombre écrit en base 2, il représente la valeur cinq.

Contrairement au système décimal où les poids sont des puissances de 10, dans le système binaire les poids sont des puissances de 2. Ainsi, au lieu d'avoir affaire à des unités, dizaines, centaines, etc. on a affaire à des unités, deuzaines, quatraines, huitaines, seizaines, etc.

Exemple :

Décomposition et calcul de la valeur du nombre qui s'écrit « 101_2 » dans le système binaire.

Chiffre	1	0	1
Rang	2	1	0
Poids	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
Valeur	$1 \times 4 = 4$	$0 \times 2 = 0$	$1 \times 1 = 1$

Valeur du nombre « 101_2 » : $4 + 0 + 1 = 5$, soit cinq.

Pour convertir un nombre du système binaire vers le système décimal on peut toujours utiliser cette méthode du tableau, mais on peut aussi directement faire le calcul en ligne.

Exemple :

Conversion du nombre $11\ 010_2$ vers le système décimal :
 $11\ 010_2 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 16 + 8 + 2 = 26_{10}$

Exercice 3 : Calculer la valeur d'un nombre binaire

Avec la méthode de votre choix, convertissez les trois nombres binaires suivants vers le système décimal :

101 101_2

10 011_2

111 010_2

Exercice 4 : Énumérer les nombres binaires

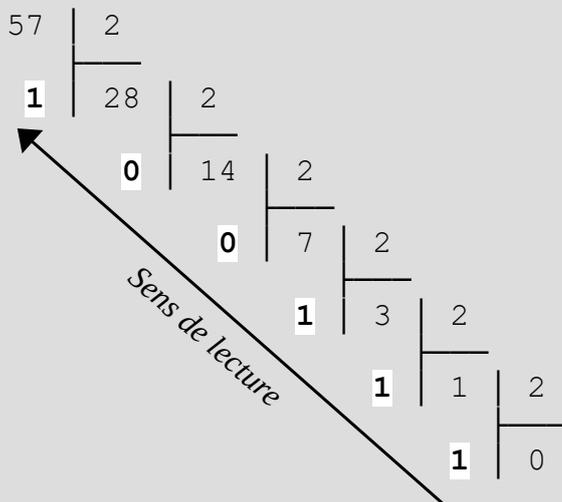
Écrivez dans l'ordre les 17 premiers nombres binaires (dont les valeurs vont de 0 à 16).

LEÇON 3 : Convertir vers le système binaire

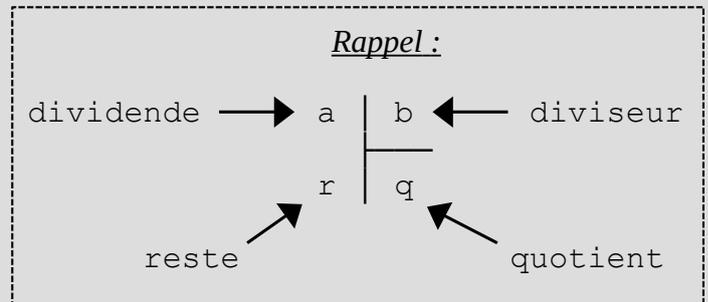
Pour convertir un nombre du système décimal vers le système binaire il suffit de faire des divisions euclidiennes successives par 2 jusqu'à obtenir 0 comme quotient. Puis on lit les restes dans l'ordre inverse pour former le nombre binaire.

Exemple :

Conversion du nombre 57_{10} vers le système binaire :



Donc $57_{10} = 111\ 001_2$



Conseil pratique : Pour faciliter l'écriture et la lecture d'un nombre binaire, et ainsi éviter les erreurs, regroupez les chiffres trois par trois (et n'oubliez pas la base). Écrivez ainsi $111\ 001_2$ plutôt que 111001 .

Exercice 5 : Convertir en base 2

Convertissez les trois nombres en base 10 suivants vers le système binaire avec la méthode des divisions euclidiennes successives :

71_{10}

132_{10}

758_{10}

Exercice 6 : Manipuler la base 2

1) Convertissez les trois nombres binaires suivants vers le système décimal en justifiant par un calcul :

$1\ 101\ 100_2$

$110\ 110_2$

$10\ 111\ 000_2$

2) Convertissez les trois nombres en base 10 suivants vers le système binaire en justifiant par un calcul :

38_{10}

521_{10}

399_{10}

LEÇON 4 : Le système hexadécimal

L'écriture d'un nombre binaire peut être très longue. Pour représenter de grandes valeurs avec moins de symboles on utilise donc le système hexadécimal, qui est une notation positionnelle à base 16.

Les seize chiffres du système hexadécimal sont : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

Les dix premiers chiffres ont les mêmes valeurs que dans le système décimal, puis le A vaut dix, le B vaut onze, le C vaut douze, le D vaut treize, le E vaut quatorze, et le F vaut quinze.

⚠ Attention ! A, B, C, D, E et F ne sont pas des lettres ici, ce sont des chiffres !

Contrairement au système décimal où les poids sont des puissances de 10, dans le système hexadécimal les poids sont des puissances de 16. Ainsi, au lieu d'avoir affaire à des unités, dizaines, centaines, etc. on a affaire à des unités, seizaines, deux-cinq-cinquante-sizaines (puisque $16^2 = 256$), etc.

Pour convertir un nombre du système hexadécimal vers le système décimal on peut utiliser la méthode du tableau ou le calcul en ligne direct.

Exemple :

Conversion du nombre qui s'écrit « $1B5_{16}$ » dans le système décimal.

Chiffre	1	B = 11	5
Rang	2	1	0
Poids	$16^2 = 256$	$16^1 = 16$	$16^0 = 1$
Valeur	$1 \times 256 = 256$	$11 \times 16 = 176$	$5 \times 1 = 5$

Donc $1B5_{16} = 256 + 176 + 5 = 437_{10}$

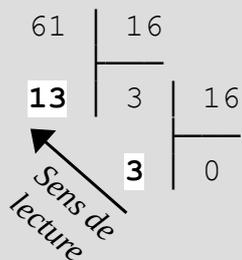
Méthode en ligne directe :

$$1B5_{16} = 1 \times 16^2 + 11 \times 16^1 + 5 \times 16^0 = 256 + 176 + 5 = 437_{10}$$

Pour convertir un nombre du système décimal vers le système hexadécimal on doit utiliser la méthode des divisions euclidiennes successives en utilisant 16 comme diviseur.

Exemple :

Conversion du nombre 61_{10} vers le système hexadécimal :



Donc $61_{10} = 3D_{16}$ (car la valeur 13 est représentée par le chiffre D)

Exercice 7 : Énumérer les nombres hexadécimaux

- 1) Écrivez dans l'ordre les 50 premiers nombres hexadécimaux (dont les valeurs vont de 0 à 49).
- 2) Quel nombre hexadécimal succède à 99_{16} ?
- 3) Quel nombre hexadécimal succède à $9F_{16}$?
- 4) Quel nombre hexadécimal succède à FF_{16} ?

Exercice 8 : Manipuler le système hexadécimal

- 1) Convertissez en base 10 et justifiez vos réponses par un calcul.

$$D3_{16} \quad 7AC_{16} \quad 9B7_{16}$$

- 2) Convertissez en base 16 et justifiez vos réponses par un calcul.

$$184_{10} \quad 546_{10} \quad 359_{10}$$

- 3) Effectuez les additions des nombres hexadécimaux suivants sans les convertir :

$$3A_{16} + 7C_{16} = ?$$

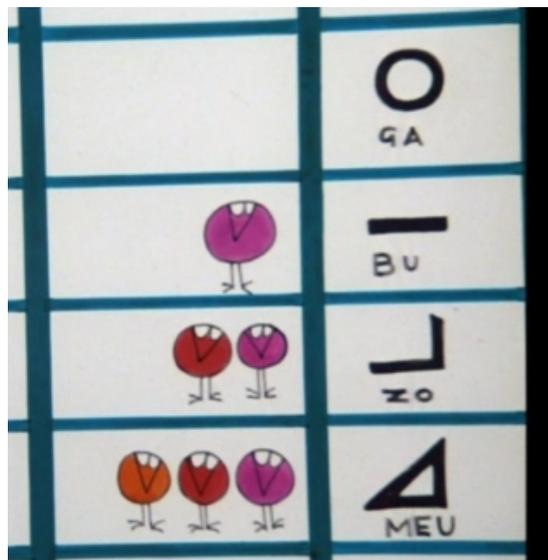
$$B9_{16} + 4D_{16} = ?$$

$$1C7_{16} + 8A_{16} = ?$$

Exercice 9 : Compter comme un Shadok

Regardez l'extrait du dessin animé des Shadoks puis répondez aux questions suivantes.

- 1) Quelle est la base du système de numération des Shadoks ?
- 2) Combien valent les nombres suivants dans le système décimal ?
 $BU BU = ?$
 $ZO ZO ZO = ?$
 $MEU BU ZO ZO GA = ?$
 $BU GA ZO GA MEU = ?$
- 3) Comment prononce-t-on en Shadok les nombres suivants ?
 $25 = ?$
 $73 = ?$
 $100 = ?$
 $999 = ?$



Extrait du dessin animé des Shadoks

Exercice 10 : Convertir en temps limité

Allez sur l'application en ligne « Convertisseur de Bases » à l'adresse : <https://evoluscope.fr/convertisseur/>

Cliquez sur le lien « [S'entraîner >>](#) » et lisez la consigne. Rassemblez le matériel demandé et quand vous êtes prêt cliquez sur le bouton « Commencer ».

Si vous n'avez pas obtenu un score de 10/10 recommencez pour améliorer votre score.

CHAPITRE II : Stocker et coder des informations

LEÇON 5 : L'organisation d'un ordinateur

Un bit (contraction de *binary digit*) est une unité élémentaire d'information ne pouvant prendre que deux valeurs distinctes : 0 ou 1. Un octet est une suite de 8 bits (symbole « o »).

Dans un octet on peut donc stocker exactement un nombre hexadécimal à deux chiffres. Vérifiez en effet que :
 $11111111_2 = 255_{10} = FF_{16}$

Une quantité d'informations se mesure donc en bits ou en plus couramment en octets.

Préfixes à connaître : kilo- (k) = 10^3 / méga- (M) = 10^6 / giga- (G) = 10^9 / téra- (T) = 10^{12}

Donc 1 Go c'est un milliard d'octets.

Un fichier (*file* en anglais) est un ensemble d'octets stockés dans une mémoire permanente (disque dur par exemple). Un fichier est toujours identifié par un nom de fichier et il possède une extension qui renseigne sur la nature des données qu'il contient. L'extension permet au système d'exploitation de déterminer le programme par défaut pour ouvrir ce fichier. L'extension est séparée du nom du fichier par un point.

⚠ Attention à ne pas confondre l'extension et le format du fichier. Si vous changez manuellement l'extension d'un fichier l'ordinateur ne pourra plus le lire car son format ne correspondra plus à l'extension !

Exemples :

- lisezmoi.txt : « lisezmoi » est le nom du fichier, « txt » est l'extension qui indique qu'il s'agit d'un texte.
- voyage.jpeg : « voyage » est le nom, « jpeg » est l'extension qui dit qu'il s'agit d'une image.
- tableau.jpeg : attention c'est un piège, ce n'est pas un tableau mais une image, on ne peut pas l'ouvrir avec Microsoft Excel ou LibreOffice Calc !

Précautions concernant les noms des fichiers :

Pour faciliter la lecture et éviter les problèmes de compatibilité entre différents systèmes il est recommandé de n'utiliser que les caractères suivants pour les noms des fichiers : les 26 lettres de l'alphabet latin en minuscules et majuscules (a-z, A-Z), les 10 chiffres de 0 à 9, le trait d'union (-) et le tiret du bas (_). Il faut éviter le point, l'espace, les accents, ou tout autre signes de ponctuation ou caractères spéciaux.

Exemples :

Voici plusieurs noms de fichiers incorrects proprement renommés :

- ❌ devoir n°1.docx → ✅ devoir_n1.docx
- ❌ compte rendu final.doc → ✅ compte_rendu_final.doc
- ❌ été_2024.jpg → ✅ ete_2024.jpg
- ❌ résumé+bio.txt → ✅ resume_bio.txt
- ❌ schéma.à.rendre.png → ✅ schema_a_rendre.png

Extensions de fichiers à connaître :

Texte brut : .txt

Tableau brut : .csv

Document texte : .doc, .docx, .odt

Feuille de calcul : .xls, .xlsx, .ods

Présentation : .ppt, .pptx, .odp

Document imprimable (non modifiable) : .pdf

Image : .bmp, .jpg, .jpeg, .png, .gif

Son : .wav, .mp3

Vidéo : .avi, .mp4, .mov

Programme exécutable : .exe

Archive : .zip

Page web : .htm, .html

Script Python : .py

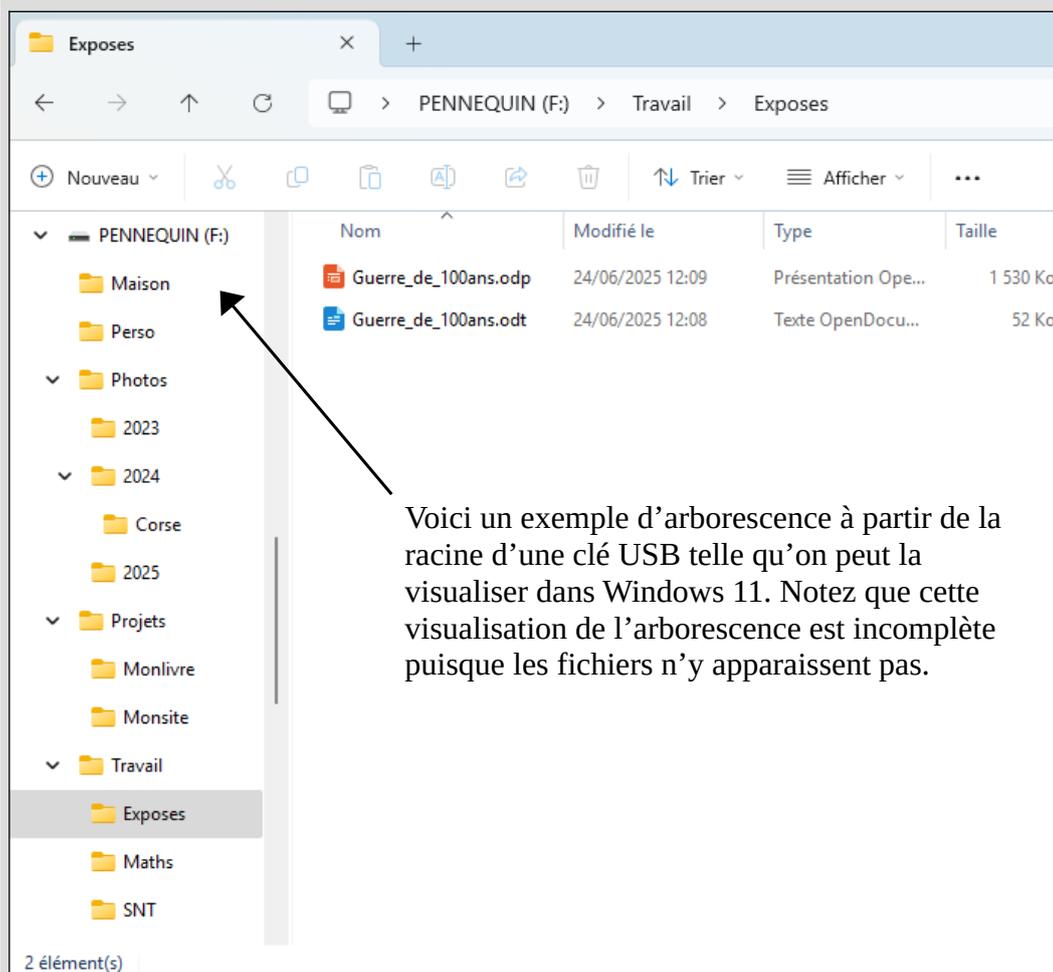
Les données (*data* en anglais) sont les informations contenues dans le fichier.

Les métadonnées (*metadata* en anglais) sont des informations que l'ordinateur a sur le fichier (dates de création, de dernière modification, taille du fichier, auteur, coordonnées géographiques, etc.). Sur Windows on peut accéder aux métadonnées d'un fichier en faisant un clic droit dessus, puis « Propriétés ».

Les fichiers sont disposés dans un système de fichiers qui comprend des emplacements appelés dossiers ou répertoires (*folders* ou *directories* en anglais). Tous les fichiers sont placés dans des dossiers et tous les dossiers sont placés dans d'autres dossiers, sauf la racine qui est le seul dossier non inclus dans un autre.

Cette organisation des dossiers forme une arborescence, c'est-à-dire une structure en forme d'arbre dont les fichiers seraient les feuilles.

Exemple :



Voici un exemple d'arborescence à partir de la racine d'une clé USB telle qu'on peut la visualiser dans Windows 11. Notez que cette visualisation de l'arborescence est incomplète puisque les fichiers n'y apparaissent pas.

**Capture d'écran de
l'Explorateur de fichiers
sous Windows 11**

Un chemin (*path* en anglais) est une succession de dossiers inclus les uns dans les autres, qui permet d'identifier un emplacement particulier dans le système de fichiers. Sur Windows les dossiers d'un chemin sont séparés par des antislashes (\).

Exemple :

C:\Utilisateurs\Jerome\Documents\TP1\compte_rendu.odt

Le fichier compte_rendu.odt est rangé dans le dossier TP1, lui-même dans Documents, etc. et C:\ est la racine du disque C (ce qui correspond souvent au disque dur principal).

Le bureau est un dossier spécial qui s'affiche lors de l'allumage de l'ordinateur et qui sert d'espace de travail temporaire.  **Attention, lorsque vous fermez votre session au lycée tout ce qui est sur le bureau est supprimé.**

Un système d'exploitation (en anglais *Operating System*, abrégé OS) est un ensemble de programmes essentiels qui permettent à un ordinateur de fonctionner en assurant :

- la gestion du matériel (mémoire, processeur, périphérique) ;
- l'organisation des fichiers ;
- l'exécution des autres programmes.

Exemples : Les systèmes d'exploitation courants sont Windows, macOS, Linux pour les ordinateurs fixes. On trouve Android ou iOS pour les téléphones portables.

Exercice 11 : Étudier une arborescence

Répondez aux questions suivantes en vous servant de la capture d'écran de la page précédente.

- 1) Redessinez l'arborescence en traçant des branches entre chaque dossier enfant et son dossier parent.
- 2) Quelles les tailles des deux fichiers situés dans le dossier « Exposes » ?
- 3) Écrivez le chemin complet vers le fichier de *Présentation* ayant pour sujet la guerre de Cent Ans.

Exercice 12 : Mesurer l'information

- 1) Combien y a-t-il de bits dans 7 octets ?
- 2) Combien d'octets contient un fichier de 3 ko ?
- 3) Convertissez 32,5 ko en Mo.
- 4) Un fichier contient 156 220 000 octets. Quelle est sa taille en Go ?

Exercice 13 : Connaître les extensions courantes

Reliez les extensions de fichiers aux types de contenus sans regarder le cours de la page précédente.

XLSX ●	● Document texte
PNG ●	
MP3 ●	● Document imprimable
ODT ●	
JPEG ●	● Vidéo
ODS ●	
MP4 ●	● Feuille de calcul
BMP ●	
DOCX ●	● Son
MOV ●	
PDF ●	● Image

Exercice 14 : Rendre sa clé USB retrouvable

Allumez l'ordinateur et branchez votre clé USB. À la racine de votre clé USB faites un clic droit et créez un fichier texte au format TXT qui s'appelle (en majuscules) « LISEZMOI.txt ».

Dans ce fichier écrivez les informations suivantes :

- votre NOM et votre Prénom
- votre classe
- le nom du lycée,
- l'année scolaire en cours
- un message invitant la personne qui trouverait cette clé à la ramener au bureau de la Vie scolaire du lycée.

TP 1 : Explorer le système de fichiers

Nous allons découvrir le fonctionnement du système de fichiers de Windows en manipulant des dossiers et différents types de fichiers.

Répondez aux questions en gras sur votre cahier d'exercices.

Constituez un binôme, allumez l'ordinateur et ouvrez une session.

Sur le bureau, cliquez sur « Ce PC » et naviguez dans le système de fichiers pour accéder au dossier suivant :

P:\Diffusion\SNT\PENNEQUIN\dossier_en_desordre

✚ Copiez ce dossier sur le bureau.



1) Observez le contenu du dossier et donnez-lui un nom pertinent. Créez des sous-dossiers pour organiser les fichiers par thèmes. 🖋️ **Faites un dessin de votre arborescence.**

Corrigez les noms des fichiers en supprimant ou en remplaçant les caractères interdits (comme les espaces, les accents, les caractères spéciaux, etc.).

2) Repérez le fichier « truc.doc ». Faites un clic droit dessus, puis allez dans ses « Propriétés ».

🗺️ **Quel est le chemin absolu (c'est-à-dire depuis la racine) de ce fichier ?**

3) Repérez le fichier « budget-voyage-japon.xlsx ». Changez l'extension de ce fichier en .ods pour le convertir en feuille de calcul au format LibreOffice. Essayez maintenant d'ouvrir le fichier.

Le fait d'avoir modifié l'extension a-t-il réellement permis de changer le format du fichier ? Justifiez.

4) On souhaite maintenant créer une archive au format 7Z contenant les fichiers réorganisés. Faites un clic droit sur le dossier réorganisé, allez dans le menu « 7-Zip » pour créer une archive au format 7Z. 🗜️

- **Une archive est-elle un fichier ou bien un dossier ? Justifiez.**

- **Comparez la taille du dossier réorganisé et de l'archive que vous venez de créer.**

5) On souhaite finalement plutôt une archive au format ZIP. Modifiez l'extension de votre archive en .zip et vérifiez que le fichier est toujours lisible.

Si ce n'est pas le cas expliquez pourquoi, puis créez une nouvelle archive au format ZIP.

Renommez l'archive avec les initiales en majuscules des deux membres de votre binôme. Connectez-vous à l'ENT et placez l'archive dans les « Dossiers partagés » dans la catégorie « Informatique ».

TP 2 : Les métadonnées des photos

Nous allons nous étudier les métadonnées associées à des photographies et nous en servir pour les géolocaliser.

Répondez aux questions en gras sur votre cahier d'exercices.

Constituez un binôme, allumez l'ordinateur et ouvrez une session. Sur le bureau, cliquez sur « Ce PC » et naviguez dans le système de fichiers pour accéder au dossier suivant :

P:\Diffusion\SNT\PENNEQUIN\collection_photos

✦ Copiez les fichiers qui s'y trouvent sur le bureau.

1) On s'intéresse à « pelican.jpg ». 📷

- **Quel est le nom du fichier ?**
- **Quelle est son extension ?**
- **Quel type de contenu renferme-t-il ?**

2) Faites un clic droit sur le fichier « pelican.jpg » et rendez-vous dans les « Propriétés » pour consulter les métadonnées.

- **Quel est la taille du fichier ?**
- **À quelle date et à quelle heure a été effectuée la prise de vue ?**
- **Avec quel appareil photo (modèle et marque) ?**
- **Le flash a-t-il été utilisé ?**

3) Cherchez les coordonnées géographiques de la prise de vue. Les différents nombres des deux premières coordonnées correspondent dans l'ordre aux degrés (°), minutes d'angles (') et secondes d'angles ("). Arrondissez à un seul chiffre après la virgule pour les secondes d'angles. La troisième coordonnée est la hauteur exprimée en mètres. 🌐

Notez ces coordonnées GPS.

4) À l'aide du navigateur rendez-vous sur <https://www.google.fr/maps> et rentrez les coordonnées GPS, latitude puis longitude séparés par une virgule. Utilisez un point à la place de la virgule dans les nombres.

Exemple :

Les coordonnées GPS du lycée Jeanne d'Arc sont : 45°46'37.3",3°5'42.3"

Où a été prise la photographie ?

5) Répondez aux mêmes questions (2, 3 et 4) à propos de la photographie « lac.jpeg ». 🏞️

Pouvez-vous identifier de quel lac il s'agit ?

6) De même, étudiez la photographie « montagne.jpeg ». 🏔️

Dans quel pays se situe cette montagne ?

Renommez l'archive avec les initiales en majuscules des deux membres de votre binôme. Connectez-vous à l'ENT et placez l'archive dans les « Dossiers partagés » dans la catégorie « Informatique ».



LEÇON 6 : Le codage des données

Dans les fichiers texte les caractères sont codés par des nombres. On utilise une table de correspondance entre les nombres et les caractères qu'on appelle la table ASCII. Elle permet de coder 128 caractères avec les nombres de 0 à 127, ou de 0 à 7F en hexadécimal. Dans la norme ASCII un caractère occupe toujours exactement un octet (même si seulement 7 bits sont utilisés, le premier bit est donc toujours un 0).

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STH	ETH	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	spc	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

La table de caractères ASCII (codage hexadécimal). Ici « spc » est le caractère d'espace. On lit dans l'ordre le numéro de la ligne puis le numéro de la colonne.

Exemples : $41_{16} = \text{« A »}$ $6F_{16} = \text{« o »}$ $20_{16} = \text{espace}$ $35_{16} = \text{« 5 »}$

Des extensions de la table ASCII permettent de coder davantage de caractères (caractères accentués, caractères grecs, caractères chinois, émojis, etc.) en utilisant plusieurs octets par caractères. C'est la table Unicode.

Le format de fichier le plus simple pour consigner du texte est le format TXT. L'éditeur de texte le plus simple est le logiciel Bloc-notes (sur Windows). Aucune mise en forme n'est possible dans le format TXT.

Dans les fichiers image les pixels peuvent être codés avec le système RGB : la couleur est divisée en ses trois composantes primaires rouge (R), vert (G), bleu (B), et chaque composante est codée sur un octet, c'est-à-dire avec un nombre allant de 0 à 255 (de 00 à FF). Chaque pixel est donc codé sur 3 octets.

Le système RGB est un modèle additif où les couleurs se mélangent en ajoutant de la lumière. Cela ne fonctionne pas comme de la peinture !

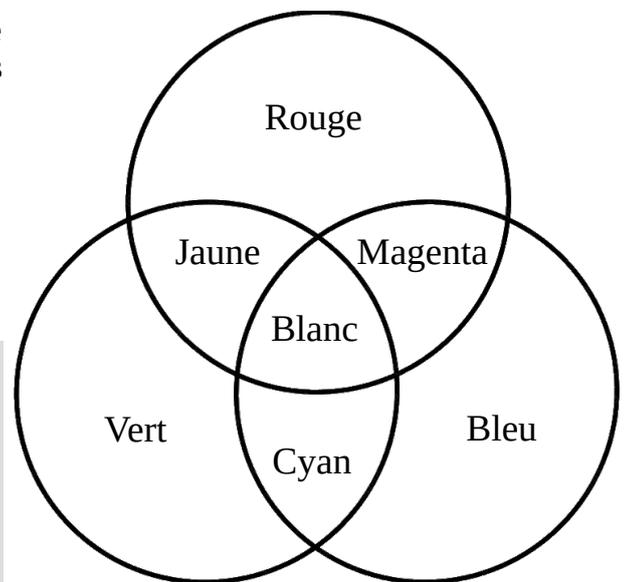
Colorie proprement le diagramme ci-contre.

Rappels sur la synthèse additive des couleurs :

Rouge + Vert = Jaune Bleu + Vert = Cyan
 Rouge + Bleu = Magenta Rouge + Vert + Bleu = Blanc

Exemples :

rgb(255,0,0) = Rouge #FFFF00 = Jaune
 rgb(128,0,0) = Rouge foncé #000000 = Noir
 rgb(255,255,255) = Blanc #0000FF = Bleu
 rgb(255,192,192) = Rose #A08040 = Marron



Les images peuvent être stockées dans les fichiers selon différents types de formats.

- **Format BMP** : format brut (non compressé) où chaque pixel est représenté individuellement.
- **Formats compressés sans perte** : comme PNG, permettent de réduire la taille du fichier sans altérer l'image. Les données sont transformées pour occuper moins d'espace, *mais peuvent être restituées exactement*.
- **Formats compressés avec pertes** : comme JPEG, suppriment certaines informations visuelles jugées secondaires, ce qui permet de réduire fortement la taille du fichier, *mais au prix d'une dégradation de l'image*.

Exercice 15 : Un proverbe codé

1) À l'aide de la table ASCII, déchiffrez le texte suivant :

43 68 6F 73 65 20 70 72 6F 6D 69 73 65 2C 20

2) Complétez la phrase précédente en chiffrant le texte suivant en hexadécimal :
chose due.

Exercice 16 : Déchiffrer un texte binaire

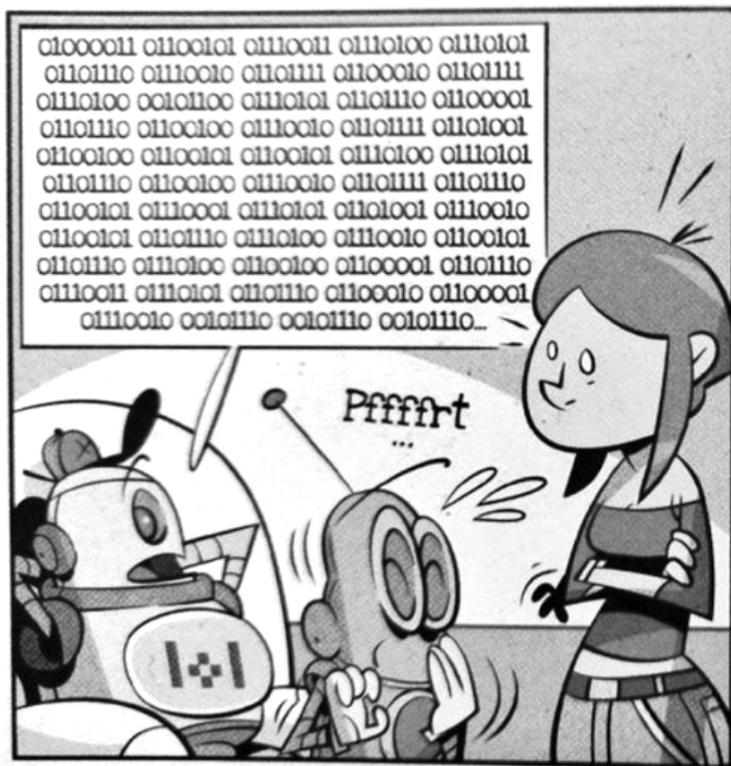
On peut convertir directement un nombre binaire en un nombre hexadécimal en regroupant les chiffres par groupes de 4 puis en convertissant chaque groupe en un seul chiffre hexadécimal.

Déchiffrez le texte ci-dessous en utilisant la table ASCII. Que remarquez-vous de singulier dans l'orthographe de ce message ?

```
01000010 01110010 01100001 01110110
01101111 00101100 00100000 01110100
01110101 00100000 01100001 01110011
00100000 01110000 01110010 01100101
01110011 01110001 01110101 01100101
00100000 01110100 01101111 01110101
01110100 00100000 01110100 01110010
01101111 01110101 01110110 11101001
00101110 00101110 00101110
```

Exercice 17 : Blague de robot

Déchiffrez la blague dans la case de bande dessinée ci-contre (*Rob Niveau 3*, James, Boris Miroir, éd. Delcourt, p. 56).



Exercice 18 : Coloriage RGB

Identifie chacune des couleurs ci-dessous, puis colorie le quadrillage ci-contre en respectant le code de chaque case.

- 0 = rgb(0,0,0)
- 1 = rgb(255,255,255)
- 2 = rgb(255,0,0)
- 3 = rgb(0,255,0)
- 4 = rgb(0,0,255)
- 5 = #FFFF00
- 6 = #FF00FF
- 7 = #00FFFF
- 8 = #808080
- 9 = #FFC0C0

1	1	1	1	1	5	5	5	5	5	5	1	1	1	1	1
1	1	1	5	5	5	5	5	5	5	5	5	5	1	1	1
1	1	5	5	5	5	5	5	5	5	5	5	5	5	1	1
1	5	5	5	5	5	5	5	5	5	5	5	5	5	5	1
1	5	5	4	4	4	4	4	5	5	4	4	4	4	5	5
1	5	5	4	0	1	4	5	5	4	0	1	4	5	5	1
1	5	5	4	0	0	4	5	5	4	0	0	4	5	5	1
1	5	5	4	4	4	4	5	5	4	4	4	4	5	5	1
1	5	5	5	5	5	5	5	5	5	5	5	5	5	5	1
1	5	5	5	5	5	5	5	5	5	5	5	5	5	5	1
1	5	5	5	2	5	5	5	5	5	5	2	5	5	5	1
1	5	5	2	2	2	1	1	1	1	2	2	2	5	5	1
1	5	5	2	2	2	2	2	2	2	2	2	2	5	5	1
1	1	5	5	2	2	2	9	9	2	2	2	5	5	1	1
1	1	1	5	5	2	9	9	9	9	2	5	5	1	1	1
1	1	1	1	5	5	5	5	5	5	5	5	1	1	1	1

Exercice 19 : Le code Morse

Le codage Morse, développé dans les années 1830 par Samuel Morse et Alfred Vail, est une méthode de transmission d'informations utilisant des séquences de signaux courts et longs. Historiquement, il joua un rôle crucial dans la communication à longue distance, notamment en télégraphie, et marqua les prémices de la communication codée. Il s'agit d'un précurseur des codages binaires plus récents comme l'ASCII.

Un tiret (signal long) est trois fois plus long qu'un point (signal court). Le silence entre deux éléments d'une même lettre dure aussi longtemps qu'un point.

Le silence entre deux lettres dure aussi longtemps qu'un tiret. Le silence marquant une espace entre deux mots est 7 fois plus long qu'un point, et il se note avec une barre oblique (/).

A	● —	U	● ● —
B	— ● ● ●	V	● ● ● —
C	— ● — ●	W	● — —
D	— ● ●	X	— ● ● —
E	●	Y	— ● — —
F	● ● — ●	Z	— — ● ●
G	— — ●		
H	● ● ● ●		
I	● ●		
J	● — — —		
K	— ● — —		
L	● ● ● ●		
M	— —		
N	— ●		
O	— — —		
P	● — — ●		
Q	— — ● ●		
R	● — ●		
S	● ● ●		
T	—		
		1	● — — — —
		2	● ● — — —
		3	● ● ● — —
		4	● ● ● ● —
		5	● ● ● ● ●
		6	— ● ● ● ●
		7	— — ● ● ●
		8	— — — ● ●
		9	— — — — ●
		0	— — — — —

1) Chiffrez le message suivant en Morse : SOS

2) Faites de même avec ce message : LE PROGRES

3) Déchiffrez le message en Morse suivant :

— . . . — — — — — . . . — . . . — . . . — — — — . / . . . — — — . . . /

4) Combien de symboles différents le code Morse utilise-t-il ? (Attention ce n'est pas deux !)

Exercice 20 : La taille du génome humain

Le génome humain de référence contient environ 3,1 milliards de nucléotides. Un nucléotide est représenté par une lettre parmi A, T, C ou G.

1) Combien d'octets sont nécessaires pour coder une seule lettre en ASCII dans un fichier de texte brut ?

2) Quel serait la taille approximative d'un fichier .txt contenant ce génome entier ? Choisissez une unité appropriée pour l'exprimer.

3) Avec la police Courier New et une taille de 12 pt une page au format A4 avec 2 cm de marge peut contenir 53 lignes et chaque ligne peut contenir 66 caractères. Combien peut-on mettre de caractères par page ? Combien de pages seraient nécessaires pour imprimer le génome humain de référence ? Utilisez le nombre de chiffres significatifs pertinent pour écrire cette estimation.

4) Une feuille de papier standard mesure environ un dixième de millimètre d'épaisseur. Quel serait la hauteur de la pile de feuilles si on imprimait tout le génome de référence en recto verso ? Utilisez une unité appropriée pour exprimer cette quantité.

5) Une feuille de papier standard pèse 80 g/m² et mesure 21 cm de large par 29,7 cm de haut. Quelle est la masse d'une feuille de papier standard ? Quelle est la masse de la pile de feuilles précédemment imprimées (on négligera la masse de l'encre) ? Choisissez une unité appropriée.



TP 3 : Le codage des textes

Nous allons explorer la manière dont les fichiers texte et les documents texte sont codés dans un ordinateur.

Répondez aux questions en gras sur votre cahier d'exercices.

Connectez-vous à l'ENT, puis dans l'espace des classes téléchargez le fichier « bienvenue.zip » sur le bureau. Ce fichier est une archive. Faites un clic droit dessus puis allez dans le menu « 7-zip » pour extraire les fichiers compressés.

1) Pour chacun des trois fichiers que vous avez extraits faites un clic droit et allez dans « Propriétés ».

Notez l'extension et la taille de chaque fichier.

2) Ouvrez les trois fichiers avec LibreOffice Writer.

Quels points communs et différences constatez-vous ?

Rendez-vous sur le site HexEdit <https://hexed.it/> pour explorer le codage hexadécimal de ces fichiers. Ouvrez les trois fichiers à l'aide de cette application en ligne.

3) Commençons par explorer le fichier « bienvenue.txt ».

D'après ce que vous voyez, à combien de chiffres hexadécimaux correspond chaque caractère du texte ?

4) Comparez la longueur du texte et la taille du fichier texte.

Que constatez-vous ? Comment pouvez-vous l'expliquer ?

5) Modifiez l'un des nombres hexadécimaux. **Que constatez-vous ?**

Faites plusieurs essais pour retrouver les nombres correspondant aux 26 lettres en minuscule de l'alphabet.

Notez ces codes hexadécimaux. Dans quel ordre sont-ils ?

6) Explorez maintenant « bienvenue.rtf ».

Quelles différences constatez-vous avec le fichier précédent ?

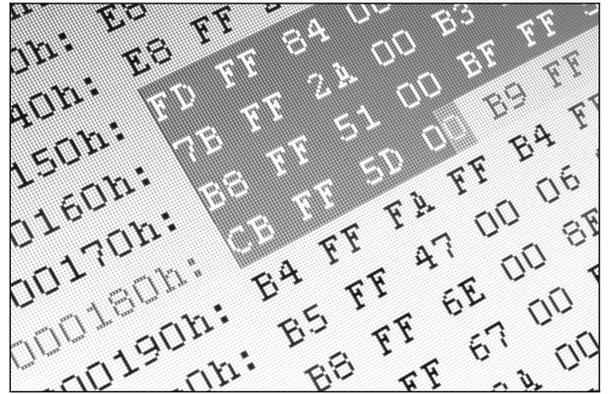
7) Retrouvez dans ce code l'emplacement des mots du texte.

Par quel sorte de code est entouré chaque mot ? Pouvez-vous deviner ce qu'il signifie ?

8) Explorez enfin « bienvenue.odt ».

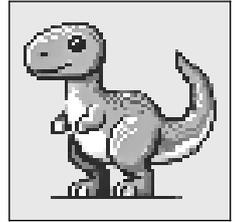
- **Pouvez-vous retrouver les mots du texte ? Pourquoi selon vous ?**

- **Déduisez-en l'utilité d'un logiciel comme LibreOffice Writer.**



TP 4 : Les couleurs des pixels

Constituez un binôme, allumez l'ordinateur et ouvrez une session. Sur le bureau, cliquez sur « Ce PC » et naviguez pour accéder au dossier suivant, puis copiez-le sur le bureau :
P:\Diffusion\SNT\PENNEQUIN\Dinos



1) Ouvrez le fichier « dino.bmp » avec Paint.

- **Quelles sont ses dimensions ?**

- **Sachant que chaque pixel est codé par 3 octets, combien d'octets devrait comporter ce fichier ?**

2) Affichez les propriétés et relevez la taille exacte du fichiers en nombre d'octets.

Combien y a-t-il d'octets de différence par rapport à votre prédiction ?

3) Rendez-vous sur le site HexEdit <https://hexed.it/> pour explorer le codage de ce fichier. Les octets supplémentaires que vous avez calculés dans la question précédente forment une entête au début du fichier avant le code du premier pixel.

Repérez les 3 octets du premier pixel.

Dans le format BMP les 3 octets d'un pixel de couleur sont dans l'ordre inversé Bleu, Vert, Rouge.

Modifiez la composante verte du code du premier pixel (écrivez 00 à la place de l'octet correspondant) et enregistrez le nouveau fichier sous le nom de « dino2.bmp ». Ouvrez ce nouveau fichier avec Paint.

Où se trouve le premier pixel dans l'image ?

4) Modifiez maintenant le code du deuxième pixel avec HexEdit et enregistrez le nouveau fichier sous le nom de « dino3.bmp ». Ouvrez-le ensuite avec Paint.

Où se trouve le deuxième pixel par rapport au premier ? Les pixels sont-ils regroupés colonne par colonne ou bien ligne par ligne ?

5) Cherchez maintenant avec HexEdit le code du premier pixel de la deuxième ligne en partant du bas.

Pour cela calculez le nombre total d'octets occupés par la première ligne et ajoutez le nombre d'octets de l'entête du fichier.

Pour vous rendre dans le fichier à la position que vous venez de calculer utilisez dans le volet de droite le champ « Aller à » et tapez simplement le nombre (en base 10) correspondant à cette position. Enregistrez et ouvrez le nouveau fichier avec Paint pour vérifier que vous avez bien modifié le bon pixel.

Votre calcul vous a-t-il permis de trouver le bon pixel ?

6) On rappelle que les 3 octets sont ici dans l'ordre inversé Bleu, Vert, Rouge.

Quel est dans ce cas le code hexadécimal de la couleur orange (on attend une suite de 6 chiffres) ?

7) L'œil du dinosaure comporte deux points très clairs. Les deux points très clairs sont situés respectivement sur le 19^e pixel de la 52^e ligne en partant du bas de l'image, et sur le 20^e pixel de la 53^e ligne.

Calculez les positions respectives du premier octet de chacun de ces deux pixels dans le code hexadécimal du fichier de l'image.

Avec HexEdit rendez-vous sur ces deux positions, modifiez les codes de ces deux pixels en orange, enregistrez le fichier en « dino4.bmp », puis ouvrez-le avec Paint pour vérifier que vous avez modifier les bons pixels.

8) Comparez l'image de « dino.bmp » sur laquelle vous avez travaillé avec « dino_compresse.jpg ».

La taille de ce fichier est-elle différente ? Qu'en est-il de la qualité de l'image ? Déduisez-en à quel type de compression correspond le format JPG.

9) Remplacez au hasard un octet au milieu du fichier JPG par 00 et enregistrez le nouveau fichier. Comparez attentivement l'ancienne image compressée et la nouvelle.

Quel effet cela a-t-il eu ? Avez-vous modifié un seul pixel ? Cela confirme-t-il votre réponse précédente ?

CHAPITRE III : Traiter des données structurées

LEÇON 7 : Le format CSV

Les données structurées sont des informations organisées sous la forme de tableaux, c'est-à-dire selon plusieurs dimensions (au moins deux dimensions : les lignes et les colonnes). Un tableau de ce genre est couramment appelé une table de données. Ces tables de données permettent de représenter des informations plus complexes qu'une simple liste. Des données structurées peuvent comporter plusieurs tables de données reliées entre elles, on les appelle alors des bases de données.

Le format CSV est un format très courant et très simple pour représenter des tables de données. Cette simplicité rend ce format facilement lisible par les hommes même sans traitement particulier. Les fichiers CSV contiennent en effet du texte dont l'organisation mime celle d'un tableau : chaque ligne du texte représente une ligne du tableau et chaque colonne est séparée par une virgule.

Exemple :

Descripteur = titre d'une colonne

Objet = ligne entière
(sauf la première)

```
eval.csv
Fichier  Modifier  Affichage
Prénom,Note,Appréciation
Marie,16,"Bien"
Julien,18.5,"Très bien"
Antoinette,4,"Insuffisant, il faut travailler davantage"
Ln 1, Col 25 | 121 caractères | 100% | Windows (CRLF) | UTF-8
```

Valeur = donnée située à l'intersection d'une colonne et d'une ligne
(c'est-à-dire une case du tableau)

⚠ Attention aux difficultés suivantes :

- Pour les **nombres décimaux** la virgule « , » doit être remplacée par un point « . ».
- **Si une valeur contient une virgule** alors il faut encadrer cette valeur avec des guillemets doubles « " » (sinon la virgule sera considérée comme un séparateur entre deux colonnes, voir l'exemple ci-dessus).
- Dans les fichiers CSV en langue française le point-virgule est parfois utilisé comme séparateur à la place de la virgule, *ce qui est pourtant déconseillé*.

Un fichier CSV peut être créé et modifié avec un simple éditeur de texte (comme Bloc-notes), mais il faut utiliser un tableur-grapheur comme LibreOffice Calc ou Microsoft Excel pour traiter les données, c'est-à-dire pour les trier, les filtrer, faire des calculs, etc. LibreOffice Calc permet d'enregistrer les tableaux comme des feuilles de calcul au format ODS, qui est un format bien plus riche (formules, mises en forme, graphiques) que le format CSV.

Voici les formules utiles pour traiter des données avec LibreOffice Calc au lycée (à connaître) :

=SOMME (plage)	Calcule la somme d'une plage	=NB (plage)	Indique le nombre de cellules non vides d'une plage
=MOYENNE (plage)	Calcule la moyenne d'une plage	=ALEA ()	Produit un nombre aléatoire entre 0 et 1
=MEDIANE (plage)	Calcule la médiane d'une plage	=LN (cellule)	Calcule le logarithme népérien d'un nombre
=MIN (plage)	Trouve le minimum d'une plage	=EXP (cellule)	Calcule l'exponentiel d'un nombre
=MAX (plage)	Trouve la maximum d'une plage	=ARRONDI (a;b)	Arrondit un nombre a avec b chiffres après la virgule

Remarque : Une *plage* est un ensemble de cellules, par exemple « B2:B10 » qui signifie « de B2 à B10 ».

TP 5 : Traiter les données des secours français

Nous allons nous entraîner à explorer et à traiter des données publiées sur le Web en accès libre.

Répondez aux questions en gras sur votre cahier d'exercices.

L'État français rend publique les données qu'il a développées dans le cadre de ses missions et qu'il juge non sensibles à l'adresse suivante : <https://www.data.gouv.fr>

Allez sur le site des données publiques de l'État français. Dans le moteur de recherche de ce site tapez le mot « secours » afin d'obtenir la table de données sur les « Interventions réalisées par les services d'incendie et de secours ».



📌 Téléchargez le fichier le plus récent (au format CSV) et placez-le sur le bureau.

1) Ouvrez le fichier avec le logiciel Bloc-notes.

- **Le tableau est-il facilement lisible ?** 🗨️

- **Quel caractère est utilisé pour séparer les colonnes ? S'agit-il du séparateur standard ?**

- **Est-il possible de traiter les données avec ce logiciel ? Justifiez.**

Ouvrez le fichier avec LibreOffice Calc. Au moment de l'ouverture, choisissez le jeu de caractères « Europe occidentale (ISO-8859-1) » et le séparateur approprié (décochez tous les autres).

2) On souhaite savoir quels sont les 7 départements où il y a eu le plus d'interventions au total. Pour cela commencez par repérer le descripteur « Total interventions », sélectionnez la colonne correspondant à ce descripteur, choisissez dans le menu « Données » l'option « Trier... », une fenêtre apparaît, faites « Étendre la sélection » pour que l'intégralité des objets soient triées et pas seulement la colonne sélectionnée.

Listez ensuite les noms des 7 départements (sans doublon) où il y a eu le plus d'interventions dans l'ordre décroissant.

3) Vous devez maintenant effectuer un tri selon deux critères pour répondre à la prochaine question. Lorsque la fenêtre apparaît lors du tri, utilisez deux clés de tris en les choisissant de manière pertinente.

Quels sont les 7 départements de catégorie C ayant connu le plus d'interventions pour des accidents sur les lieux de travail ?

4) Réalisez un nouveau tri selon deux critères pertinents. **Parmi les départements où il y a eu des secours en mer quels sont ceux où il n'y a pas eu de fausse alerte ?** 🚚

5) Écrivez des formules dans des cellules pour répondre aux dernières questions.

- **Quel est le nombre total d'interventions pour « Secours à victime » sur toute la France ?**

- **Combien cela fait-il d'interventions par jour (arrondissez à l'unité) ?**

- **Quelle la moyenne du nombre d'interventions par département pour les accidents de circulation (arrondissez à l'unité) ?** 🚗

TP 6 : Une exploitation forestière

Traiter des données peut permettre de trouver des solutions à des problèmes précis et concrets qui peuvent avoir des conséquences économiques.

Répondez aux questions en gras sur votre cahier d'exercices.

Dans une exploitation forestière de cerisiers noirs (*Prunus serotina*), on veut estimer le volume de bois que ces arbres pourront produire si on les abattait (donc avant de le faire). Nous utilisons pour cela un jeu de données contenant des mesures effectuées sur 31 arbres abattus de différentes tailles, et dont le volume de bois est connu. 🌲



Constituez un binôme, allumez l'ordinateur et ouvrez une session. Sur le bureau, cliquez sur « Ce PC », puis copiez le fichier suivant sur le bureau :
P:\Diffusion\SNT\PENNEQUIN\arbres_cerisier_noir.csv

1) Ouvrez ce fichier avec LibreOffice Calc.

- **Quelles données contient-ils exactement ?**

- **Parmi ces données lesquelles pourraient nous permettre d'estimer le volume de bois avant l'abattage ?**

2) Pour tester vos hypothèses précédentes nous allons construire deux graphiques. Allez dans le menu « Insertion » puis « Diagramme » pour créer un premier graphique. Cliquez sur « XY (dispersion) » puis « Terminer ». Déplacez le graphique pour qu'il ne gêne pas la lecture du tableau. Double-cliquez sur votre graphique pour retourner dans ses menus de conception.

3) Cliquez sur le bouton « Plage de données » . Supprimez toutes les séries de données créées automatiquement et créez une nouvelle série en cliquant sur « Ajouter ». Cliquez sur cette nouvelle série puis sélectionnez « Valeurs X » pour choisir les données à mettre en abscisse. Cliquez sur le petit bouton  à droite de « Plage pour Valeurs X » puis sélectionnez avec la souris les valeurs (sans le titre de la colonne) de la colonne « Hauteur ». De même, pour les ordonnées vous devez sélectionner la colonne « Volume ». Cliquez ensuite sur « OK » pour fermer la fenêtre.

4) Cliquez enfin sur le bouton « Type de diagramme »  tout en haut et sélectionnez « Points seuls » pour supprimer les segments reliant les points sur le graphique. Cliquez sur le bouton « Titres »  pour ajouter un titre au graphique ainsi qu'aux deux axes.

5) Créez un deuxième graphique en recommençant toute la procédure, mais cette fois-ci vous devez choisir la colonne « Circonférence » comme données à placer en abscisse.

6) Maintenant que les deux graphiques ont été créés nous voulons tracer deux droites qui passent au milieu de ces deux nuages de points pour estimer dans chaque cas la corrélation entre le paramètre mesuré et le volume de bois. Double-cliquez sur un graphique pour l'éditer, puis faites un clic-droit sur l'un des points pour ouvrir le menu et choisissez « Insérer une courbe de tendance... ». Choisissez un type linéaire, ne forcez pas l'ordonnée à l'origine, mais cochez bien « Afficher l'équation » et « Afficher le coefficient de détermination (R^2) », puis fermez la fenêtre en cliquant sur « OK ». Faites de même avec le deuxième graphique.

7) Le coefficient de détermination R^2 estime combien les points sont dispersés et éloignés de la courbe de tendance (0 = points aléatoires ; 1 = points parfaitement alignés).

- **Dans lequel des deux graphiques R^2 est-il le plus élevé ?**

- **Vaut-il mieux se servir de la hauteur ou bien de la circonférence des arbres pour estimer le volume de bois que l'on obtiendra après les avoir abattus ?**

- **Calculez le volume qu'on obtiendrait si on abattait 100 arbres de 40 cm de circonférence en moyenne.**

TP 7 : La durée de vie des mammifères

Traiter des données permet parfois de mieux comprendre la nature et de répondre à des questions scientifiques.

Répondez aux questions en gras sur votre cahier d'exercices.

Dans cet exercice on cherche à mettre en évidence une relation (si elle existe) la masse moyenne des différentes espèces de mammifères et leur durée de vie maximale. Nous allons pour cela traiter des données qui ont été collectées par des chercheurs.

Allez sur le serveur du lycée et copiez sur le bureau le fichier suivant :

P:\Diffusion\SNT\PENNEQUIN\vertebrata_data.csv

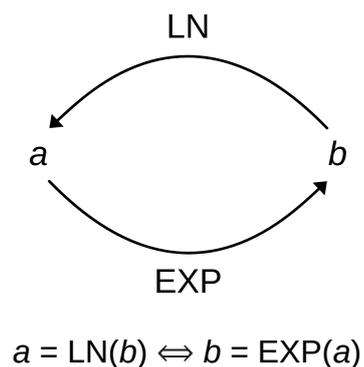
1) Ouvrez le fichier avec le Bloc-notes.

Quel est le séparateur de colonnes utilisé ? S'agit-il du séparateur de colonnes standard ?

Ouvrez le fichier avec LibreOffice Calc en sélectionnant seulement le séparateur de colonnes approprié dans le menu d'ouverture et en choisissant la langue « Anglais (États-Unis) » afin que les valeurs numériques soient lues correctement par le logiciel.

Lisez les explications suivantes pour bien comprendre à qui correspondent les données dans chaque colonne :

Pour chaque espèce les descripteurs indiquent la classification de cette espèce (Phylum, Class, Order, Family, Genus, Species), le nom commun en anglais, la masse moyenne des adultes en grammes (AdultMass) et la durée de vie maximale en années (MaxLife). Les deux derniers descripteurs (LMaxLife et LAdultMass) indiquent le résultat d'une transformation mathématique des deux données précédentes avec la fonction logarithme népérien (ce qui correspond à la formule « =LN(cellule) » dans LibreOffice Calc). La fonction exponentielle (c'est la formule « =EXP(cellule) ») est l'inverse du logarithme népérien.



2) Cherchez la section où se trouve la classe des Mammifères (Mammalia). **Quels sont les numéros des lignes où cette section commence et se termine ? Combien cela fait-il d'espèces de Mammifères ?**

3) Créez un graphique de type « XY (dispersion) ». Placez en abscisses les LAdultMass des mammifères, et en ordonnée leurs LMaxLife. Ajoutez des titres au graphique et aux axes. Faites un clic-droit sur l'un des points pour construire une courbe de tendance. Choisissez le type linéaire sans forcer l'ordonnée à l'origine, et affichez l'équation de la droite et le coefficient de détermination R^2 .

Quelle est la relation mise en évidence (arrondissez les valeurs à 3 chiffres après la virgule) ?

4) Le coefficient R^2 indique si les points suivent bien la courbe de tendance (0 = points aléatoires ; 1 = points parfaitement alignés). **Les points suivent-ils bien cette relation ?**

5) Repérez la ligne sur laquelle se trouve l'homme sachant que son nom scientifique est *Homo sapiens* et qu'il appartient à l'ordre des Primates. Relevez la masse moyenne de l'homme et sa durée de vie maximale. Repérez le point représentant l'homme sur votre graphique (grâce aux valeurs LAdultMass et LMaxLife) et changez sa couleur pour le mettre en évidence.

L'homme suit-il bien la relation mise en évidence ou bien est-il très éloigné de la courbe de tendance ?

Calculez la durée de vie maximale que devrait avoir l'homme par rapport à sa masse moyenne s'il suivait strictement la courbe de tendance.

CHAPITRE IV : Le langage Python

LEÇON 8 : Découverte de Python

Un algorithme est une suite d'instructions permettant de résoudre un problème ou d'accomplir une tâche (comme une recette de cuisine par exemple). Un programme est un algorithme destiné à être exécuté par un ordinateur. Les programmes peuvent servir à faire fonctionner des appareils (voitures, électroménager), produire des documents (logiciels de traitement de texte), traiter des données, calculer, jouer (jeux vidéos), etc.

Un langage de programmation est un système de notation rigoureux permettant de décrire un algorithme de façon compréhensible pour un ordinateur. On distingue principalement deux catégories de langages de programmation selon la manière dont les programmes sont exécutés :

- Un langage compilé permet de transformer le programme en un fichier exécutable directement par le système d'exploitation de l'ordinateur. Ce fichier exécutable a souvent l'extension .exe sous Windows.
- Un langage interprété nécessite qu'un autre programme, appelé un interpréteur, se charge de lire et d'exécuter les instructions, ligne par ligne. L'interpréteur est un programme compilé directement exécuté par le système d'exploitation. Un programme ayant besoin d'un interpréteur est souvent appelé un script.



Python et son environnement

Python est un langage de programmation libre, interprété, et multi-plateforme, dont le développement a commencé en 1991. Il est largement utilisé pour l'apprentissage de la programmation en raison de sa simplicité et de sa lisibilité. Python peut être téléchargé gratuitement à l'adresse suivante : <https://www.python.org/>

Une fois Python installé, il est possible de lancer le logiciel en cherchant « Python IDLE » dans le menu Démarrer. À l'ouverture, la console de Python s'affiche, elle permet de dialoguer de manière interactive avec l'interpréteur. Cette console permet d'écrire et d'exécuter des instructions une par une, elle ne permet pas d'écrire un script complet.

Pour écrire un véritable programme, il est possible d'utiliser un éditeur de texte simple comme le Bloc-notes, mais Python intègre également son propre éditeur, accessible via le menu File > New File. Une fois le programme enregistré dans un fichier avec l'extension .py il est possible de le faire exécuter par l'interpréteur via le menu Run > Run Module, ou avec la touche de raccourci F5.

Manipuler des variables

Une variable est une association entre un nom et une valeur modifiable. On peut la représenter comme une boîte étiquetée, dans laquelle on place une information. Le nom d'une variable doit respecter certaines règles : il ne doit contenir que des lettres non accentuées (a à z), éventuellement des chiffres (sauf en première position), et il ne doit contenir ni espace, ni trait d'union (-), ni caractère spécial. En revanche le tiret du bas (_) est autorisé.

Python distingue différents types de variables. Voici les 3 types fondamentaux à connaître (d'autres types existent mais ils sont abordés dans des cours plus avancés) :

- **Les entiers** (*integers* en anglais) : nombres sans décimales (par exemple 15) ;
- **Les nombres à virgule** (*floating-point numbers* en anglais) : ils s'écrivent avec un point comme séparateur décimal (par exemple 3.14 et non 3,14) ;
- **Les chaînes de caractères** (*strings* en anglais) : elles représentent du texte et doivent être encadrées par des guillemets (par exemple "bonjour").

Quelques opérateurs fondamentaux (à bien connaître) sont utilisés pour manipuler les variables :

=	permet d' <u>affecter</u> une valeur à une variable, ce n'est pas la même signification qu'en maths ;
+, -, *, /	servent à faire des opérations arithmétiques ;
(et)	permettent de prioriser les opérations ;
**	permet de calculer une puissance (par exemple 2**3 vaut 8) ;
+	permet aussi de <u>concaténer</u> deux chaînes de caractères (c'est-à-dire de les coller ensemble).

Voir l'**Annexe 1** pour une liste plus complètes des opérateurs disponibles en Python.

Exemples :

La fonction `print()` permet d'afficher dans la console le contenu de ce qui a été mis entre parenthèses.

```
prix_article = 12.5
quantite = 4
total = prix_article * quantite
print(total)
```

Le programme ci-dessus va afficher « 50.0 » dans la console lorsqu'on l'exécutera.

```
prenom = "Lina"
message = "Bonjour " + prenom + ", bienvenue !"
print(message)
```

Le programme ci-dessus va afficher « Bonjour Lina, bienvenue ! » dans la console lorsqu'on l'exécutera.

```
age = 15
texte = "J'ai " + age + " ans"
print(texte)
```

Le programme ci-dessus va afficher un message d'erreur car « age » est une variable de type entier, ce n'est pas une chaîne de caractères, on ne peut donc pas la concaténer avec les deux autres chaînes de caractères. Pour que le programme soit correct il aurait fallu écrire sur la première ligne : `age = "15"`

Il faut bien comprendre la différence entre « 15 » et « "15" » dans ce dernier exemple. Le premier est un nombre, c'est la quantité 15 qui est stockée en mémoire. Le deuxième est un texte, c'est le caractère "1" (numéro ASCII 31) suivi du caractère "5" (numéro ASCII 35) qui sont stockés en mémoire.

Exercice 21 : Utiliser des variables dans la console

Notez dans votre cahier les instructions que vous avez tapées correctement sur l'ordinateur ainsi que les résultats affichés dans la console.

1) Dans la console de Python définissez trois variables pour stocker les prix des articles suivants.
Pommes : 2,50 €/kg Tomates : 3,15 €/kg Carottes : 1,80 €/kg

2) Créez trois nouvelles variables pour stocker les quantités achetées : 900 g de tomates, 1 kg de carottes et 1,5 kg de pommes.

3) Créez une nouvelle variable pour calculer le prix du panier en vous servant uniquement des variables précédentes et pas des nombres. Affichez le résultat.

LEÇON 9 : Utiliser des fonctions en Python

Une fonction est un bloc d'instructions qu'on peut appeler pour exécuter une tâche précise. Une fonction peut prendre en entrée une ou plusieurs valeurs, appelées arguments, et elle renvoie parfois un résultat. Les arguments s'écrivent entre parenthèses, après le nom de la fonction.

qui prend en entrée zéro, un ou plusieurs paramètres et renvoie un résultat. Ces paramètres sont les arguments de la fonction, ils sont placés entre parenthèses. Par exemple

Exemple :

La fonction `len()` permet de connaître la longueur d'une chaîne de caractères.

```
longueur = len("français")
```

Ce code enregistre 8 dans la variable « longueur ». Notez que la variable permettant d'enregistrer le résultat se trouve toujours à gauche du signe égal.

Plusieurs fonctions de base sont disponibles directement en Python. Voici quelques-unes des plus utiles à connaître pour débiter :

- `type()` : pour afficher le type de variable, renvoie la valeur `<int>` pour un nombre entier, la valeur `<float>` pour un nombre décimal et la valeur `<str>` pour une chaîne de caractères.

- Il est possible de convertir une variable d'un type vers un autre à l'aide de fonctions spécifiques :

`str()` transforme une valeur en chaîne de caractères,

`int()` transforme une valeur en entier,

`float()` transforme une valeur en nombre décimal.

Cependant, ces fonctions ne modifient pas automatiquement la variable d'origine : si on veut conserver le résultat de la conversion, il faut explicitement l'assigner à une variable avec l'opérateur `=`.

Exemple :

```
a = "33.3"
```

```
float(a)
```

```
type(a)
```

```
a = "33.3"
```

```
a = float(a)
```

```
type(a)
```

❌ Programme incorrect. Ici on verra que la variable `a` est toujours une chaîne de caractères.

✅ Programme correct. Ici on verra que la variable `a` est bien devenue un nombre décimal.

- La fonction `print()` permet d'afficher dans la console le contenu d'une variable.

- La fonction `input()` permet de mettre le programme en pause pour demander à l'utilisateur de saisir une valeur au clavier. Ce que l'utilisateur saisit est **toujours** considéré comme une chaîne de caractères, il faut donc le convertir si on veut faire des calculs avec des nombres.

Python permet également d'utiliser des fonctions supplémentaires en important des modules (ou bibliothèques, ou *libraries* en anglais), c'est-à-dire des ensembles de fonctions spécialisées. Pour importer toutes les fonctions d'un module, on utilise la syntaxe suivante, au début du programme :

```
from nom_du_module import *
```

Le module `math` contient des fonctions mathématiques comme `sin(a)` (qui utilise une valeur en radians pour la variable `a`), ainsi que la constante `pi`. Le module `random` propose la fonction sans argument `random()` qui génère un nombre aléatoire compris entre 0 et 1, et la fonction `randint(a, b)` qui renvoie un entier aléatoire entre `a` et `b`, bornes incluses.

Exercice 22 : Utiliser des fonctions dans la console

Notez dans votre cahier les instructions que vous avez tapées correctement sur l'ordinateur ainsi que les résultats affichés dans la console.

1) Dans la console de Python, affectez l'année en cours à la variable `annee_actuelle` et affectez votre année de naissance à la variable `annee_naissance`. Réalisez une soustraction de ces deux variables (pas une soustraction directe des nombres) et affectez le résultat à une nouvelle variable `age` pour calculer l'âge que vous avez eu ou que vous aurez cette année lors de votre anniversaire. Affichez le résultat.

2) Assignez votre prénom à la variable `prenom`, puis affichez les types des variables `prenom` et `age` à l'aide de la fonction appropriée.

3) Définissez une variable `presentation` dans laquelle vous rédigerez une phrase de présentation incluant par concaténation les deux variables `prenom` et `age`. *Attention à bien convertir la variable `age` en une chaîne de caractères avec la fonction appropriée.*

4) Affichez la longueur de votre phrase de présentation grâce à la fonction adéquate.

Exercice 23 : Jouer avec les codes ASCII

Notez dans votre cahier les instructions que vous avez tapées correctement sur l'ordinateur ainsi que les résultats affichés dans la console.

1) Dans la console de Python, utilisez la fonction `ord()` pour obtenir et afficher le numéro ASCII du caractère « a » (en minuscule). Faites de même pour le caractère « b », puis « c ». Que remarquez-vous ?

2) Dans la console, calculez le numéro ASCII de la 16^e lettre de l'alphabet sans chercher à savoir quelle est cette lettre, seulement à l'aide des résultats précédents.

3) Utilisez la fonction `chr()` pour transformer ce numéro ASCII en la lettre correspondante et affichez le résultat. Vérifiez que la lettre que vous avez obtenue est bien la 16^e lettre de l'alphabet.

4) Comparez le numéro ASCII du caractère « a » (minuscule) et du caractère « A » (majuscule). De même comparez « b » et « B », puis « c » et « C ». Que remarquez-vous ? Déduisez-en le numéro ASCII de la 16^e lettre de l'alphabet en majuscule et vérifiez dans la console que votre déduction est correcte.

Exercice 24 : Mon premier programme

Écrivez un programme `bonjour.py` qui commence par dire `bonjour`. Puis le programme demande à l'utilisateur de saisir son prénom, et ensuite son nom de famille. Enfin le programme utilise ces informations pour faire une phrase de réponse. Testez votre programme en l'exécutant.

Exercice 25 : Convertisseur de prix

Cherchez sur internet le taux de change actuel pour convertir des dollars américains en euros. Écrivez un programme `dollareuro.py` qui demande à l'utilisateur quel prix en dollars il souhaite convertir en euros, puis qui effectue cette conversion et l'affiche avec un arrondi à deux chiffres après la virgule.

LEÇON 10 : Définir des conditions en Python

Lorsqu'on écrit un programme, on souhaite souvent que certaines instructions ne soient exécutées que si une condition est remplie. On peut préciser ces conditions après le mot-clé `if` (qui signifie « si ») suivi d'une condition logique. Cette condition doit être une expression dont la valeur est soit vrai, soit fausse.

On met deux points « : » à la fin de la ligne du `if` et on indente (décale) le bloc d'instructions qui dépend de cette condition. Cette indentation est obligatoire. On utilise pour cela soit exactement 4 espaces, soit la touche de tabulation du clavier (voir ci-contre).



Exemple :

```
choix = input("Saisissez une valeur inférieure à 10 : ")
if float(choix) > 10 :
    print("Trop grand !")
print("Fin du programme")
```

Dans ce code le premier `print` ne sera exécuté que si la condition est vraie, alors que le deuxième `print` ne dépend pas de cette condition, il sera toujours exécuté car il n'est pas indenté (décalé).

Si l'on souhaite que d'autres instructions soient exécutées lorsque la condition n'est pas remplie, on peut utiliser le mot-clé `else`, qui signifie « sinon ». Comme pour `if`, il faut un deux-points : et une indentation.

Cela permet donc de choisir entre deux blocs d'instructions selon que la condition est vraie ou fausse.

Le mot-clé `elif` (contraction de « else if ») permet d'introduire une nouvelle condition, lorsque la première n'est pas vérifiée. On peut utiliser plusieurs `elif` à la suite d'un `if`, pour tester plusieurs cas différents, les uns après les autres.

Exemple :

```
note = input("Saisissez votre note sur 20 : ")
note = float(note)
if note >= 16 :
    print("Très bien")
elif note >= 10 :
    print("Passable")
else :
    print("Insuffisant")
```

Ce code demande à l'utilisateur de saisir une note, la convertit en nombre, puis affiche un commentaire selon sa valeur. Il utilise une structure conditionnelle pour choisir entre trois messages.

Exercice 26 : Comparer des valeurs

1) Écrivez un programme `nombremin.py` qui demande de saisir deux valeurs et qui affiche ensuite la plus petite des deux valeurs.

2) Écrivez un programme `chainemax.py` qui demande de saisir deux chaînes de caractères et qui affiche ensuite la plus grande des deux chaînes (celle qui a le plus de caractères).

Exercice 27 : Convertisseur de température

Cherchez sur Internet la formule permettant de convertir des degrés Celsius en Fahrenheit et inversement. Écrivez un programme `temperature.py` qui demande à l'utilisateur s'il veut convertir une température de Celsius en Fahrenheit ou l'inverse (il faut par exemple demander à l'utilisateur de taper « c » ou « f »).

En fonction du choix, le programme demande ensuite la température à convertir, effectue la conversion et arrondit le résultat à deux chiffres après la virgule. Le résultat est alors affiché avec la bonne unité.

Exercice 28 : La divisibilité d'un nombre

Créez un programme que vous nommerez `multiple.py`. Ce programme doit demander à l'utilisateur de saisir un nombre. Une fois cette valeur récupérée, le programme doit indiquer s'il s'agit d'un entier ou d'un nombre décimal. Pour cela, vous pouvez comparer le nombre avec sa version arrondie avec la fonction `round`.

Si le nombre est un entier, le programme doit ensuite déterminer s'il est pair ou impair. Pour cela, réfléchissez à ce que donne le reste de la division euclidienne par 2 selon le cas. Quelle condition permet de distinguer les deux situations ? Enfin, si le nombre est un entier, le programme doit aussi tester s'il est divisible par 3, c'est-à-dire si son reste lors de la division entière par 3 est nul.

LEÇON 11 : Faire des boucles en Python

Dans un programme il est souvent utile de répéter automatiquement un bloc d'instructions plusieurs fois. Pour cela, on utilise ce qu'on appelle une boucle. Il existe deux manières principales de faire une boucle en Python : avec le mot-clé `while`, ou avec le mot-clé `for`.

Le mot-clé `while`, qui signifie « tant que », permet de répéter des instructions tant qu'une condition est vérifiée. Cette condition est testée avant chaque répétition. Dès qu'elle devient fausse, la boucle s'arrête et le programme continue avec les instructions suivantes.

Pour la syntaxe, on écrit `while` suivi d'une condition logique (comme dans un `if`), la ligne se termine par deux-points « : », et le bloc d'instructions à l'intérieur de la boucle doit être indenté.

Exemple :

```
a = 0
while a < 10 :
    print(a)
    a = a + 1
print("C'est fini !")
```

Ce code affichera les nombres 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, puis lorsque la variable `a` sera égale à 10, la condition de la boucle ne sera plus vraie, et le code affichera donc « C'est fini ! ». **Attention, si on oublie de modifier la variable dans la boucle, la condition reste toujours vraie, donc la boucle est infinie !**

Si vous savez à l'avance combien de fois vous voulez répéter une instruction, il peut être plus simple d'utiliser une boucle `for`. Il est suivi d'une variable qui prend successivement les valeurs d'une séquence, comme une liste ou une suite de nombres. Pour générer une suite de nombres, on utilise souvent la fonction `range()`.

Exemple :

```
for a in range(10) :
    print(a)
print("C'est fini !")
```

Ce code affichera la même chose que le code précédent. L'instruction `range(10)` crée la liste des nombres de 0 à 9 (10 est exclu) qui sera parcourue par la variable `a`.

Exercice 29 : Je me répète

Écrivez un programme nommé `bureau.py` qui affiche 10 fois « Je dois ranger mon bureau ! » à l'aide du mot-clé `for`. Faites de même avec le mot-clé `while`.

Exercice 30 : Pair ou impair

Écrivez un programme `pairimpair.py` qui affiche les nombres pairs de 0 jusqu'à 100 inclus avec une boucle `for`, puis les nombres impairs de 1 jusqu'à 99 inclus avec une boucle `while`.

Exercice 31 : Une séquence d'ADN

Écrivez un programme `sequenceADN.py` qui construit une séquence d'ADN aléatoire. La séquence d'ADN doit avoir une longueur déterminée par l'utilisateur grâce à `input()`. On utilise ensuite une boucle `for` pour ajouter les lettres une par une à une séquence initialement vide. On utilisera la fonction `choice` de la bibliothèque `random` pour choisir la lettre aléatoirement parmi "ATCG".

Exercice 32 : Jouer à deviner un nombre

Écrivez un programme `devinette.py` dans lequel l'ordinateur choisit un nombre entier aléatoire entre 1 et 100 (inclus) à l'aide de la fonction `randint()` de la bibliothèque `random`.

Le joueur doit ensuite deviner ce nombre. On lui demande de proposer une valeur à l'aide de la fonction `input()`, et le programme répond si le nombre proposé est trop grand, trop petit, ou exact.

Le programme doit se poursuivre tant que l'utilisateur n'a pas trouvé le bon nombre. Vous utiliserez une boucle `while` pour cela.

Une fois le bon nombre trouvé, le programme félicite l'utilisateur et affiche le nombre de tentatives effectuées avant de réussir.



CHAPITRE V : Internet et le Web

LEÇON 12 : Internet

Internet est un réseau mondial d'ordinateurs interconnectés. Il repose sur une infrastructure physique (câbles, routeurs, serveurs, antennes, etc.) qui permet aux machines de communiquer entre elles. Ce réseau est décentralisé, ce qui signifie qu'il n'a pas de centre unique de contrôle. En cas de panne ou d'engorgement d'un point du réseau, les données peuvent suivre d'autres chemins pour atteindre leur destination.

Pour que ces échanges fonctionnent tous les appareils doivent parler le même langage. C'est le rôle des protocoles de communication. Un protocole est un ensemble de règles qui définissent comment les données doivent être organisées, envoyées, reçues et interprétées.

Le protocole TCP/IP

Le protocole TCP (*Transmission Control Protocol*) découpe les données en paquets. Chaque paquet contient une partie des données ainsi que des informations de contrôle, comme l'adresse de destination, l'ordre des paquets ou des codes de vérification. Le protocole TCP permet de s'assurer que tous les paquets sont reçus, dans le bon ordre, sans erreur et sans doublons. Si ce n'est pas le cas, ils sont renvoyés.

Ces paquets voyagent sur Internet en utilisant le protocole IP (*Internet Protocol*). Ce protocole attribue une adresse IP à chaque appareil connecté. Une adresse IP est une suite de nombres (par exemple 192.168.1.14) qui identifie de façon unique un appareil sur un réseau. Elle permet aux données de parvenir au bon destinataire.

Serveur et client

Dans un réseau on distingue souvent deux rôles : le client et le serveur. Un serveur est un ordinateur configuré pour fournir un service (site web, fichier, vidéo, etc.) à d'autres machines. Il attend les requêtes (demandes) envoyées par les clients, et y répond. Un client est un appareil (ordinateur, smartphone...) qui envoie une requête à un serveur pour obtenir un service ou des informations, puis reçoit une réponse en retour.

Ne pas confondre Internet et le Web

Internet est l'infrastructure physique du réseau alors que le Web est un réseau de documents accessibles depuis Internet et connectés entre eux par des liens hypertextes. Un lien hypertexte est un élément cliquable qui permet de passer automatiquement d'un document à un autre.

Navigateur, moteur de recherche et DNS

Un navigateur web (comme Mozilla Firefox, Microsoft Edge ou Google Chrome) est un logiciel qui permet d'afficher les pages web. Il interprète le code (HTML et CSS) contenu dans les fichiers des sites.

Un moteur de recherche (comme Google ou Bing) est un service en ligne qui permet de rechercher des informations sur le Web. Il fonctionne en indexant le contenu de millions de pages web et en utilisant des algorithmes pour afficher les résultats les plus pertinents en réponse à des requêtes saisies par les utilisateurs.

Le protocole DNS (*Domain Name System*) est un système qui traduit les noms de domaine lisibles par l'homme (comme www.exemple.com) en adresses IP compréhensibles par les ordinateurs. Cela permet aux utilisateurs d'accéder aux sites web en utilisant des noms faciles à retenir au lieu de suites de chiffres. Le protocole DNS permet donc de trouver l'adresse IP de l'ordinateur sur lequel est stocké le site web.

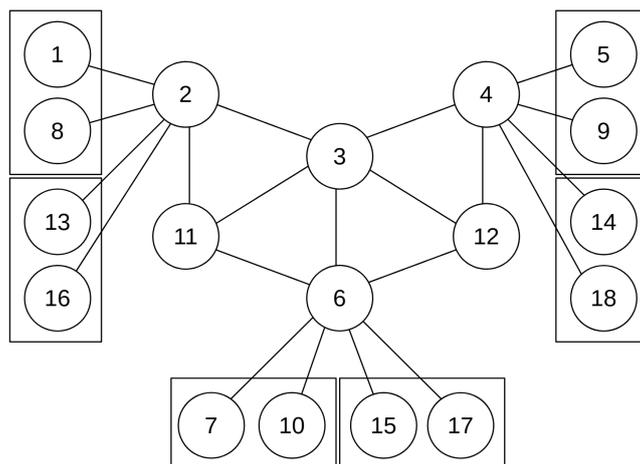
Exemple :

Analyse de l'adresse « <https://www.exemple.com> » :

Protocole	https://	Indique le protocole utilisé pour accéder à la ressource. Les plus courants sont HTTP (non sécurisé) et HTTPS (sécurisé par chiffrement).
Sous-domaine	www.	C'est une partie optionnelle de l'adresse qui peut désigner une section spécifique du site. Par exemple, www est souvent utilisé pour le site principal, mais vous pouvez avoir d'autres sous-domaines comme « shop.exemple.com ».
Nom de domaine	exemple	Le nom de domaine est le nom lisible par l'homme, c'est la partie principale de l'adresse, choisie par le propriétaire du site comme une sorte de titre.
Domaine de premier niveau (TLD)	.com	Le TLD indique le type d'organisation ou le pays de l'adresse web. Les TLD courants incluent .com, .org, .net, et les TLD nationaux comme .fr pour la France, .uk pour le Royaume-Uni, etc.

TP 8 : Simulation du protocole TCP/IP

Le silence doit être total. Vous n'avez pas le droit de parler pendant le jeu. Le but est de transmettre votre message complet à votre destinataire. Chaque élève prend position sur le réseau (voir schéma) en fonction de son numéro dans l'ordre alphabétique de la classe. Chaque élève simule un ordinateur. Il y a deux rôles possibles. Vous êtes soit un client qui envoyer et recevoir des messages, soit un routeur qui transmet les messages à leur destinataire.



Vous ne pouvez transmettre qu'UN SEUL mot à la fois. Si votre message fait 6 mots vous devez donc le découper en 6 paquets. Chaque paquet est un petit morceau de papier. Un ordinateur ne peut transmettre un paquet qu'à un autre ordinateur auquel il est directement relié (voir schéma).

- Si vous recevez un paquet qui vous est destiné vous devez émettre un accusé de réception.
- Si vous ne recevez pas l'accusé de réception d'un paquet vous devez le renvoyer.
- Si vous recevez un paquet en double vous devez à nouveau émettre un accusé de réception.

Le professeur passera dans les rangs pour intercepter des paquets au hasard :

- Vous perdez si votre destinataire ne parvient pas à reconstituer la totalité du message dans le bon ordre.
- Vous perdez aussi si l'expéditeur du message que vous avez reçu n'a pas tous les accusés de réceptions.

Chaque paquet doit contenir les informations suivantes :

- 1) L'expéditeur du paquet
- 2) Le destinataire du paquet
- 3) Le numéro du paquet
- 4) Le nombre total de paquets du message
- 5) Une donnée : un seul et unique mot de votre message

OU BIEN

- 1) L'expéditeur du paquet
- 2) Le destinataire du paquet
- 3) L'accusé de réception « Bien reçu ! »
- 4) Le numéro du paquet dont vous voulez accuser la réception

LEÇON 13 : Le Web

Pour créer une page web, on peut écrire du code dans un éditeur de texte comme Bloc-notes, Notepad++, etc. Ces outils permettent de créer et modifier des fichiers contenant le code HTML, CSS et JavaScript.

Une page web est souvent composée de plusieurs fichiers liés entre eux, comme :

- un fichier HTML pour la structure du contenu,
- un ou plusieurs fichiers CSS pour l'apparence visuelle,
- éventuellement un fichier JavaScript pour rendre la page interactive,
- et d'autres fichiers comme des images, vidéos ou polices de caractères.

Les trois langages du web sont :

- **HTML** (*HyperText Markup Language*) est un langage de balisage. Il sert à organiser le contenu d'une page : titres, paragraphes, images, liens, listes, tableaux, etc.
- **CSS** (*Cascading Style Sheets*) permet de définir le style visuel de la page : couleurs, tailles, marges, polices, disposition des éléments, etc.
- **JavaScript** est un langage de programmation utilisé pour rendre la page dynamique et interactive, par exemple pour gérer les effets des clics, les évènements, des animations, etc.

Le code HTML repose sur des balises qui indiquent le rôle de chaque élément. La plupart des balises doivent encadrer un contenu. Elles sont composées d'une balise ouvrante et d'une balise fermante.

Exemple : `<p>Voici un paragraphe</p>`

Certaines balises n'ont pas de contenu à encadrer, ce sont des balises orphelines (ou autofermantes).

Exemple : ``

En CSS, les styles sont définis à l'aide de blocs d'instructions entourés d'accolades { }. Par exemple :

Exemple :

```
p {
  color: blue;
  font-size: 16px;
}
```

Ce code signifie que tous les paragraphes (<p>) seront affichés en bleu avec une taille de police de 16 pixels.

Quand on crée un lien dans une page HTML (vers une autre page, une image, un fichier, un autre site, etc.), il existe deux manières d'écrire l'adresse cible :

- Un lien absolu utilise l'URL complète, comme : `https://www.exemple.com/images/logo.png`
- Un lien relatif indique le chemin depuis le fichier courant, en respectant l'arborescence des dossiers, par exemple : `images/logo.png`

Le lien relatif est utilisé pour relier des fichiers qui se trouvent dans le même site.

Tous les navigateurs modernes (Chrome, Firefox, Edge...) possèdent un inspecteur d'éléments, accessible par un clic droit sur une page puis « Inspecter ». Cet outil permet de :

- visualiser le code HTML et CSS d'une page web,
- modifier temporairement ce code pour tester des changements,
- déboguer les erreurs d'affichage.

Les modifications faites avec l'inspecteur ne sont pas enregistrées dans les fichiers : elles sont uniquement visibles localement.

Exemple : Voici le code HTML (à gauche) et le code CSS (à droite) pour une page web minimale.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Mon Site Web</title>
  <link rel="stylesheet"
href="style.css">
</head>
<body>
  <h1>Ceci est un titre</h1>
  <p>Ceci est un paragraphe.</p>
  <p>Ceci est un autre paragraphe avec un
<a href="http://google.fr">lien</a> vers un
site.</p>
</body>
</html>
```

```
body {
  background-color: pink;
  font-family: Arial;
  margin: 40px;
}
h1 {
  color: #8B0000;
  text-align: center;
}
p {
  font-size: 12px;
}
a {
  color: white;
}
```

TP 9 : Découverte des langages du Web

1) Allez chercher le dossier suivant sur le serveur, puis copiez-le sur votre bureau :

P:\Diffusion\SNT\PENNEQUIN\TPweb

2) Repérez et corrigez les erreurs de codage et de mise en page dans les fichiers de ce site Web :

- Le bandeau de l'entête de la page d'accueil n'est pas coloré en vert comme il le devrait.
- Les titres des pages ne sont pas centrés.
- L'image de la page d'accueil ne s'affiche pas.
- Le lien vers la page du parc des Écrins dans la page d'accueil est mort.
- La page de contact n'est pas du tout mise en forme.
- Sur la page du parc des Écrins le paragraphe sur les activités n'utilise pas la bonne mise en forme.

3) Changez la police du site, utilisez par exemple « Arial » ou « Times New Roman ».

4) Créez une nouvelle page pour l'un des parcs nationaux et insérez un lien vers celle-ci depuis « index.html ».

TP 10 : Projet de fin d'année

À l'aide de Notepad++ programmez un site web contenant au moins deux pages HTML reliées entre elles par des liens hypertextes et un fichier CSS de mise en forme du contenu. Les fichiers de votre site doivent être correctement ordonnés dans différents dossiers. Votre site doit comporter des médias comme des images. Il est important que vous indiquiez les sources de ces médias. **Votre site doit avoir pour thème une de vos passions ou un de vos hobbies.**

Annexe 1 : Les instructions en Python

Les opérations en Python

Opérateur	Rôle	Exemple
=	Affecte une valeur à une variable	nbeleve = 36
#	Insère un commentaire	nbeleve = 36 # Ceci n'est pas du code
"	Encadre une chaîne de caractères	appreciation = "Très bien !"
+	Additionne deux nombres ou bien concatène deux chaînes de caractères	somme = 13 + 27 presentation = "Je m'appelle " + prenom
-	Soustrait deux nombres	presents = nbeleve - absents
/	Divise un nombre par un autre	vitesse = distance / temps
*	Multiplie deux nombres ou bien concatène une chaîne de caractères avec elle-même le nombre de fois indiqué	airerectangle = longueur * largeur rire = "ha" * 3 # Enregistre "hahaha"
(et)	Priorité des opérations	moyenne = (note1 + note2) / 2
//	Donne le résultat de la division entière	part = 15 // 4 # Donne 3
%	Donne le reste (modulo) de la division entière	Reste = 30 % 7 # Donne 2
**	Élève un nombre à la puissance d'un autre nombre	airecube = 2 ** 3 # Donne 8

Les conditions en Python

Opérateur	Rôle	Exemple
if	Exécute les <i>instructions indentées</i> qui suivent seulement si la condition testée est vraie	<pre>a = 5 b = 5 c = 7 if a == b : # Ceci sera affiché print("a et b sont égaux") if a != c : # Ceci ne sera pas affiché print("a et c sont différents") if age > 17 : print("vous êtes majeur") if age >= 18 : print("vous êtes majeur") if age < 18 : print("vous êtes mineur") if age <= 17 : print("vous êtes mineur") if note >= 14 : print("Bien") elif note >= 10 : print("Passable") else : print("Échec")</pre>
==	Teste si deux valeurs sont égales (ou identiques)	
!=	Teste si deux valeurs sont différentes	
>	Teste si une variable est strictement supérieure à une autre	
>=	Teste si une variable est supérieure ou égale à une autre	
<	Teste si une variable est strictement inférieure à une autre	
<=	Teste si une variable est inférieure ou égale à une autre	
else	Exécute les <i>instructions indentées</i> qui suivent seulement si la condition testée par if est fausse	
elif	Exécute les <i>instructions indentées</i> qui suivent seulement si la condition testée par if est fausse et que la condition testée après elif est vraie	

Les autres mots-clés en Python

Mot-clé	Rôle	Exemple
and	Donne un résultat vrai si et seulement si les deux valeurs sont vraies	<pre>a = 5 b = 5 c = 2 d = 2 if a == b and c == d : print("Ceci s'affiche") if a < c or b > d : print("Ceci s'affiche aussi")</pre>
or	Donne un résultat faux si et seulement si les deux valeurs sont fausses	
in	Teste l' appartenance d'une valeur à une liste ou bien d'un caractère à une chaîne de caractères	<pre>5 in range(10) # Ceci est vrai "e" in "doigt" # Ceci est faux</pre>
for	Exécute en boucle les <i>instructions indentées</i> qui suivent pour chaque valeur de la liste cible	<pre>for i in range(5) : print(i ** 2) # Affiche 0, 1, 4, 9, 16</pre>
while	Exécute en boucle les <i>instructions indentées</i> qui suivent tant que la condition testée reste vraie (testée à chaque début de boucle)	<pre>i = 3 while i < 5 : print(i) # La boucle n'est i = i + 1 # exécutée que 2 fois</pre>
from et import	Permet d'importer les fonctions d'un module	<pre># Importe toutes les fonctions # de la bibliothèque random from random import *</pre>
def	Permet de définir une nouvelle fonction personnalisée	<pre>def cubemoinscarre(mavariabale) : x = mavariabale**3 - mavariabale**2 return x print(cubemoinscarre(3)) # Affiche 18</pre>
return	Termine l'exécution d'une fonction et renvoie une valeur	

Les fonctions de base en Python

Fonction	Rôle	Exemple
<code>type()</code>	Renvoie le type de la variable (nombre entier, nombre décimal, chaîne de caractères, etc.)	<code>nbeleve = 36</code> <code>type(nbeleve) # Renvoie "int"</code>
<code>print()</code>	Affiche une valeur sur la console (nombre ou bien chaîne de caractères)	<code>print("Bonjour") # Affiche "Bonjour"</code>
<code>str()</code>	Convertit un nombre en une chaîne de caractères	<code>k = str(36) # Enregistre la chaîne "36"</code>
<code>int()</code>	Convertit une chaîne de caractères ou un nombre décimal en un nombre entier	<code>nb = int(36.5) # Donne le nombre 36</code>
<code>float()</code>	Convertit un nombre entier ou une chaîne de caractères en un nombre décimal	<code>nb = float(36) # Enregistre 36.0</code> <code>nb = float("36") # Donne aussi 36.0</code>
<code>len()</code>	Donne la longueur d'une chaîne de caractères	<code>longueur = len("chien") # Enregistre 5</code>
<code>chr()</code>	Renvoie le caractère d'une valeur décimale ASCII ou Unicode donnée	<code>car = chr(65) # Enregistre "A"</code>
<code>ord()</code>	Renvoie la valeur décimale ASCII ou Unicode d'un caractère	<code>num = ord("A") # Enregistre 65</code>
<code>round()</code>	Arrondit un nombre décimal à n chiffres après la virgule	<code>resultat = round(8.36,1) # Donne 8.4</code>
<code>input()</code>	Demande à l'utilisateur de saisir un texte au clavier (renvoie toujours une chaîne de caractères)	<code>reponse = input("Quel âge as-tu ? ")</code> <code># Enregistre la chaîne "15"</code> <code># et pas le nombre 15</code>
<code>range()</code>	Renvoie une liste de n nombres entiers de 0 jusqu'à n-1	<code>range(5) # Donne la liste des cinq</code> <code># nombres suivants 0, 1, 2, 3, 4</code>
<code>replace()</code>	Remplace une chaîne de caractère par une autre	<code>txt = "j'aime les bananes"</code> <code>x = txt.replace("bananes", "pommes")</code> <code>print(x) # Affiche "j'aime les pommes"</code>

Les fonctions de la bibliothèque « random »

Fonction	Rôle	Exemple
<code>random()</code>	Renvoie un nombre décimal aléatoire entre 0 et 1	<code>nb = random()</code> <code># Donne par exemple 0,386452114529...</code>
<code>randint()</code>	Renvoie un nombre entier aléatoire entre deux bornes	<code>nb = randint(1,6)</code> <code># Donne par exemple 5</code>
<code>choice()</code>	Renvoie aléatoirement l'un des caractères d'une chaîne	<code>lettre = choice("abcdefghij")</code> <code># Donne par exemple "h"</code>

Les fonctions de la bibliothèque « math »

Fonction	Rôle	Exemple
<code>pi</code>	Constante renvoyant le nombre π	<code>x = pi # Enregistre 3.14159... dans x</code>
<code>cos()</code>	Renvoie le cosinus d'un angle exprimé en radians	<code>x = cos(pi/3) # Enregistre 0.5</code>
<code>sin()</code>	Renvoie le sinus d'un angle exprimé en radians	<code>x = sin(pi/6) # Enregistre 0.5</code>
<code>tan()</code>	Renvoie la tangente d'un angle exprimé en radians	<code>x = tan(pi/4) # Enregistre 1</code>
<code>sqrt()</code>	Renvoie la racine carrée d'un nombre positif	<code>x = sqrt(9) # Enregistre 3</code>

Annexe 2 : Les langages du Web

Les principales balises en HTML

Balise	Rôle
<html>	Racine du document HTML (tout le document doit être inclus dans cette balise)
<head>	Métadonnées de la page, non affichées à l'écran
<title>	Titre à afficher dans l'onglet du navigateur, pas dans la page
<meta />	Métadonnées, par exemple le jeu de caractères à utiliser pour afficher correctement la page <meta charset="utf-8">
<link />	Lien vers des fichiers externes, comme des feuilles de style CSS <link rel="stylesheet" href="styles/monstyle.css" />
<script>	Lien vers du code JavaScript <script type="text/javascript" src="scripts/code.js"></script>
<body>	Contenu de la page qui sera affiché
<h1>	Titre de la page
<h2>	Titre d'une section
<h3>	Sous-titre d'une section
	Liste non ordonnée
	Liste ordonnée
	Élément d'une liste (ordonnée ou non)
<p>	Paragraphe
	Texte important (italique par défaut)
	Texte très important (gras par défaut)
 	Saut de ligne
<a>	Lien hypertexte Texte du lien
	Image
<button>	Bouton cliquable
<table>	Tableau
<tr>	Ligne d'un tableau
<td>	Cellule d'une ligne d'un tableau
<header>	Entête de la page
<footer>	Pied de page

Les principales propriétés en CSS

color	Couleur du texte	text-align	Alignement du texte (center, left...)
background-color	Couleur de l'arrière-plan	margin	Espace extérieur autour de l'élément
font-size	Taille du texte	padding	Espace entre le contenu et les bords
font-family	Police d'écriture du texte	width	Largeur de l'élément
font-weight	Épaisseur du texte (bold)	height	Hauteur de l'élément
font-style	Italique (italic)	border	Bordure de l'élément, par exemple : border: 1px solid black;
text-decoration	Souligné (underline)		