

# **Chapitre 0 : Introduction**



# Table des matières

<b>I - Objectifs</b>	<b>3</b>
<b>II - Introduction</b>	<b>4</b>
<b>III - Qu'est-ce que SNT ?</b>	<b>5</b>
1. SNT : présentation.....	5
2. Science, numérique ou technologie ? .....	5
3. SNT : Plusieurs choix possibles !.....	6
<b>IV - Les 5 concepts fondamentaux du numérique</b>	<b>8</b>
1. Les données.....	8
2. Les algorithmes .....	9
3. Les langages .....	10
4. Les machines .....	13
4.1. L'ordinateur .....	13
4.2. Le système d'exploitation .....	14
5. Les interfaces.....	15
<b>V - Exercices</b>	<b>17</b>
1. Exercice : Nombre de disquettes.....	17
2. Exercice : Le premier disque dur.....	17
3. Exercice : Technologies de stockage de masse .....	17
4. Exercice : Exécution de programmes.....	18
5. Exercice : Boucles for et while.....	19
6. Exercice : Choix d'un ordinateur .....	19
7. Exercice : Utiliser un environnement de travail informatisé .....	20
<b>VI - Travaux pratiques</b>	<b>21</b>
1. Partie 1 : prise en main de l'environnement de travail .....	21
2. Partie 2 : programmation en Python .....	21

# Objectifs

- Découvrir la discipline SNT
- Appréhender les 5 piliers sur lesquels repose le numérique
- Utiliser efficacement un environnement de travail informatisé
- S'initier à l'algorithmique et à la programmation en Python

# Introduction

Depuis l'école primaire, vous avez abordé des sujets liés au numérique et à l'informatique dans de nombreux enseignements, principalement en mathématiques et technologie.

L'année de SNT sera l'occasion d'approfondir votre compréhension du numérique et de ses enjeux, de vous perfectionner dans l'utilisation des outils numériques et de faire vos premiers pas en programmation avec Python.

Les élèves souhaitant poursuivre l'apprentissage de l'informatique pourront poursuivre en spécialité NSI (numérique et sciences informatiques) en Première.

Au-delà d'une spécialisation dans les métiers du numérique, tout citoyen doit posséder des compétences et une culture numérique de base. La plateforme PIX permet de se tester à l'aide de défis à réaliser pour mieux maîtriser les outils et développer sa culture numérique. Un niveau entre 1 et 8 est attribué. Tous les élèves devront passer la certification PIX pour valider le Baccalauréat. Certaines formations supérieures demandent un certain niveau PIX. L'enseignement de SNT est bâti autour des compétences testées par cette certification.

# I Qu'est-ce que SNT ?

- SNT : présentation
- Science, numérique ou technologie ?
- SNT : Plusieurs choix possibles !

## 1. SNT : présentation

### **Définition** : SNT

---

SNT est l'acronyme de **Sciences Numériques et Technologie**

### **Remarque** :

---

Nous allons donc étudier le numérique sous trois aspects complémentaires :

- L'aspect **scientifique** : mettre en lumière quelques concepts et démarches de la science informatique sur lesquels repose le numérique
- L'aspect **numérique** : s'interroger sur les impacts du numérique dans notre société
- L'aspect **technologique** : développer une culture technologique et se perfectionner dans l'utilisation des outils numériques

## 2. Science, numérique ou technologie ?

Pour chaque activité proposée, dire si elle permet d'étudier l'aspect Science, l'aspect Numérique ou l'aspect Technologie (un seul choix possible).

### **Exercice 1**

Etudier la structure générale d'un algorithme permettant de traiter une image pixel par pixel

- ☐ Science
- ☐ Numérique
- ☐ Technologie

### **Exercice 2**

Découvrir comment traiter des données avec un tableur

- ☐ Science
- ☐ Numérique
- ☐ Technologie

### **Exercice 3**

Se questionner sur les impacts des réseaux sociaux dans la vie quotidienne

- ☐ Science
- ☐ Numérique
- ☐ Technologie

#### Exercice 4

Découvrir quelles sont les principales opérations sur les données

- ☐ Science
- ☐ Numérique
- ☐ Technologie

#### Exercice 5

Découvrir quels sont les principaux protocoles utilisés sur Internet

- ☐ Science
- ☐ Numérique
- ☐ Technologie

#### Exercice 6

S'interroger sur le poids du numérique dans la consommation énergétiques mondiales

- ☐ Science
- ☐ Numérique
- ☐ Technologie

#### Exercice 7

Découvrir la structure de graphe et comment elle peut être utilisée pour modéliser un réseau social

- ☐ Science
- ☐ Numérique
- ☐ Technologie

### 3. SNT : Plusieurs choix possibles !

Certaines activités permettent d'étudier plusieurs aspects simultanément parmi Science, Numérique et Technologie.

Pour chaque activité, sélectionner les aspects qu'elle permet d'étudier (plusieurs réponses attendues)

#### Exercice 1

Créer un site Web présentant le thème du logiciel libre. Pour créer les pages, utiliser un éditeur graphique à l'interface simple.

- ☐ Science
- ☐ Numérique
- ☐ Technologie

## Exercice 2

Ecrire un programme en Python permettant de répondre à un problème donné à l'aide d'un algorithme.

- ☐ Science
- ☐ Numérique
- ☐ Technologie

## Exercice 3

Utiliser un logiciel de traitement d'image pour réaliser un trucage d'image en superposant deux images existantes

- ☐ Science
- ☐ Numérique
- ☐ Technologie

## Exercice 4

Créer une application pour smartphone permettant de signaler les déchets polluants rencontrés lors d'une promenade. Pour programmer l'application, on utilise un programme similaire à Scratch.

- ☐ Science
- ☐ Numérique
- ☐ Technologie

## Exercice 5

Programmer un robot pour qu'il sorte d'un labyrinthe en utilisant un algorithme exploitant ses capteurs.

- ☐ Science
- ☐ Numérique
- ☐ Technologie

## Exercice 6

Créer une vidéo tutoriel pour expliquer ce qu'est le ciblage publicitaire, comment l'éviter et donner un exemple lors d'une session de navigation.

- ☐ Science
- ☐ Numérique
- ☐ Technologie

## Exercice 7

Créer un diaporama présentant la notion d'algorithme et quelques questions sociétales qu'elle peut soulever.

- ☐ Science
- ☐ Numérique
- ☐ Technologie

# II Les 5 concepts fondamentaux du numérique

- Les données
- Les algorithmes
- Les langages
- Les machines
  - L'ordinateur
  - Le système d'exploitation
- Les interfaces

## 1. Les données

### 💡 **Fondamental** :

Les **données** représentent sous forme numérique unifiée les informations : texte, image, son, vidéo, programme informatique...

Tout élément que l'on veut traiter avec un ordinateur doit être représenté (codé) en binaire : par une succession de 0 et de 1.

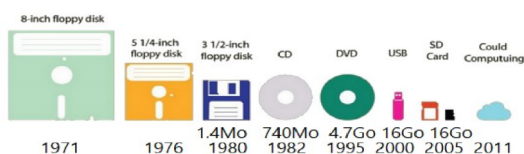
Les données sont organisées en **fichiers** et stockées en **mémoire**.

### 📎 **Remarque** :

Le mot **informatique** a été défini en 1962 par Dreyfuss. Il désigne le traitement automatique de l'information.

### ⚠️ **Attention** :

Un ordinateur ne comprend pas nativement les informations que représentent les données.



Au cours de l'histoire, différents **supports de stockage** ont été utilisés pour stocker les données.

### 💡 **Fondamental** :

L'unité de base pour mesurer la capacité de stockage est l'**octet** (symbole o), qui correspond à 8 bits (chaque bit vaut 0 ou 1).

On utilise ses multiples : le **kilo-octet** (Ko), le **méga-octet** (Mo) et le **giga-octet** (Go)

On a les relations suivantes entre unités :

$$1 \text{ Ko} = 1000 \text{ o}$$

$$1 \text{ Mo} = 1000 \text{ Ko}$$

$$1 \text{ Go} = 1000 \text{ Mo}$$



**Exemple :**

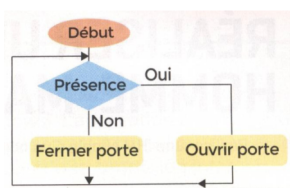
Répondre à chaque question à l'aide de calculs :

1. Combien de photos de 3 Mo peut-on stocker sur une carte mémoire de 16 Go ?
2. Un film HD occupe la totalité de l'espace mémoire d'un DVD. Combien faudrait-il de disquettes de 3,5" pour le stocker ?

## 2. Les algorithmes

**Fondamental :**

Les algorithmes est une succession précise de traitements à effectuer sur des données pour obtenir un résultat.

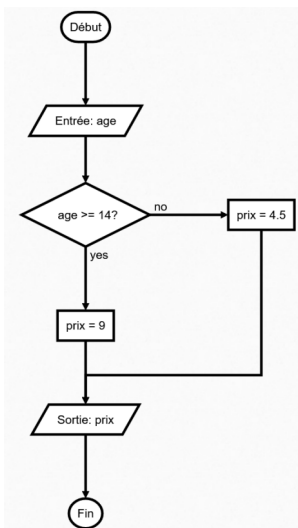
**Remarque :**

**Début**  
**Si** Présence **Alors**  
 ouvrir Porte  
**Sinon**  
 fermer porte  
**Fin Si**  
 Pause 1 seconde  
**Retour** au début

On peut définir un algorithme à l'aide d'un schéma illustrant le traitement (organigramme) ou à l'aide de texte écrit en français ou en langage informatique.

Les structures de base utilisées dans les algorithmes sont :

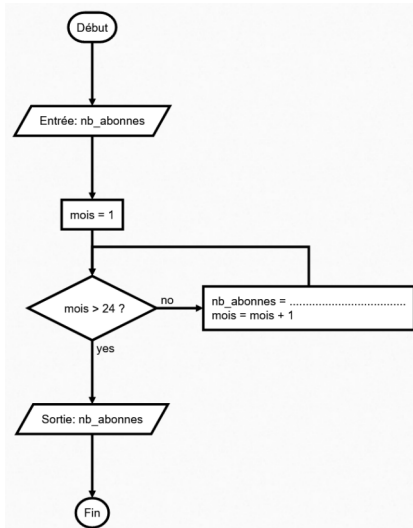
- L'affectation : donner une valeur à une variable.
- La structure conditionnelle : si un test est vérifié, alors faire des instructions. Sinon, faire d'autres instructions.
- Les structures répétitives : répéter des instructions tant qu'une condition est vérifiée.

**Exemple : Instruction conditionnelle**

Dans un cinéma, le prix change selon l'âge. L'organigramme suivant permet de connaître le prix en fonction de l'âge.

1. Combien paie un client de 15 ans ?
2. Faire une phrase traduisant cet organigramme sous la forme "Si... alors... sinon..."

## Exemple : Boucle



Un YouTubeur a 100 abonnés. Le premier mois, il gagne 10 nouveaux abonnés puis chaque mois, le nombre de nouveaux abonnés est le nombre de nouveaux abonnés du mois précédent augmenté de 10.

1. Calculer combien il aura d'abonnés au bout de 3 mois.
2. Compléter l'organigramme pour pouvoir calculer le nombre d'abonnés au bout de 2 ans.
3. Expliquer le procédé de calcul avec une phrase du type : Tant que ..., on ... puis on ...

Pour mettre en œuvre des algorithmes effectuant beaucoup de calculs, on peut utiliser un ordinateur. Il faut traduire l'algorithme dans un langage de programmation.

## 3. Les langages

### Fondamental :

Pour mettre en œuvre un algorithme sur un ordinateur, on le traduit en utilisant un langage de programmation adapté.

Chaque langage possède un vocabulaire (ensemble de mots reconnus) particuliers ainsi qu'une grammaire (ensemble de règles concernant l'enchaînement des mots pour former des structures) particulière

### Attention :

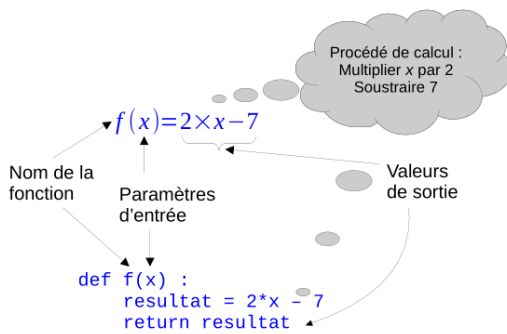
Lorsqu'on écrit un programme, il faut respecter rigoureusement la syntaxe définie par le langage. L'ordinateur ne devine pas ce qu'on souhaite lui faire faire en cas d'erreur ou de manque de précision !

Le langage machine est le seul compris par le processeur. Chaque instruction élémentaire (addition, lecture de la valeur d'une variable, test conditionnel...) est codée par une valeur binaire.

On a développé des langages plus proches de l'anglais et proposant plus d'abstraction :

- Le langage C en 1972, utilisé pour interagir avec le matériel
- Le langage Java en 1995, proposant de programmer en définissant des objets (éléments du programme interagissant entre eux)
- Le langage Python en 1991, facile d'emploi et utilisé dans de nombreux domaines. C'est celui que vous utiliserez au lycée.
- Le langage PHP en 1995, utilisé pour générer des pages Web dynamiquement (par exemple, générer le panier pour chaque utilisateur sur une boutique en ligne)

Tous ces langages sont "traduits" en langage machine lorsqu'on exécute un programme. C'est l'étape de compilation ou d'interprétation (selon les langages)



Les algorithmes peuvent être donnés sous formes de diagramme ou en français.

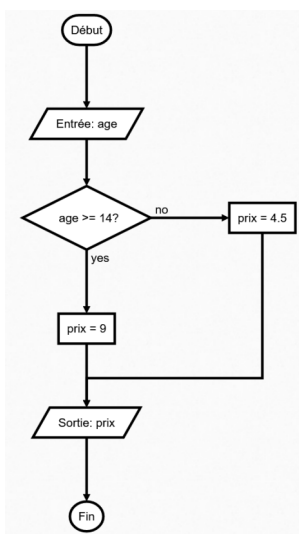
Un algorithme peut être vu comme une fonction qui prend des valeurs en entrée, effectue des calculs et renvoie des valeurs en sortie, un peu comme une fonction mathématique.

Cette analogie est illustrée ci-contre.

### Remarque :

Un algorithme, programmé avec une fonction, peut ne pas prendre de valeur en entrée. Dans ce cas, on n'écrit rien entre les parenthèses après le nom de la fonction.

### Exemple :



Le diagramme ci-contre (vu à la partie précédente) correspond à une fonction qui prend l'âge d'une personne en entrée et retourne le prix à payer.

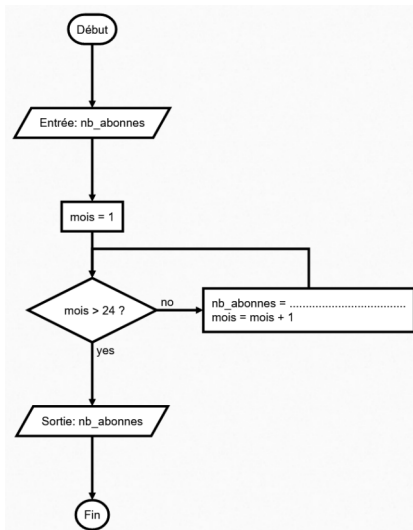
Nom de la fonction : `calcule_prix`

Paramètres d'entrée : `age` (nombre entier)

Valeurs de sortie : `prix` (nombre décimal)

```

1 Définir calcule_prix(age):
2   Si age >= 14:
3     prix = 9
4   Sinon:
5     prix = 4.5
6   retourner prix
  
```

**Exemple :**

On reprend l'algorithme schématisé ci-contre. On le représente par une fonction appelée `nombre_abonnes_apres_2_ans`.

1. Quelle est le paramètre d'entrée ?
2. Quelle est la valeur de sortie ?
3. Traduire cet algorithme en français en utilisant une instruction Tant que.

Une fois que nous avons écrit l'algorithme en français, nous pouvons les traduire dans un langage de programmation particulier.

Le tableau ci-contre présente la "traduction" entre les instructions en français et leur équivalent en Python.

Français	Python
<i>a</i> est-il égal à 5 ?	<code>a == 5</code>
<i>a</i> est-il différent de 5 ?	<code>a != 5</code>
si ... alors :	<code>if ... :</code>
...	...
si ... alors :	<code>if ... :</code>
...	...
Sinon :	<code>else :</code>
...	...
tant que ... faire :	<code>while ... :</code>
...	...
pour <i>i</i> allant de 1 à 5 faire :	<code>for i in range(1, 6) :</code>
...	...
définir <i>f(x)</i> :	<code>def f(x) :</code>
...	...
retourner ...	<code>return ...</code>

**Exemple :**

On reprend l'exemple ci-dessus. Voici son code en français.

```

1 Définir calcule_prix(age):
2   Si age >= 14:
3     prix = 9
4   Sinon:
5     prix = 4.5
6   retourner prix
  
```

A l'aide du tableau, compléter le code en Python correspondant :

```

1 def calcule_prix(age):
2   if age >= 14:
3     prix = 9
4   else:
5     .....
6   return ....
  
```

**Exemple :**

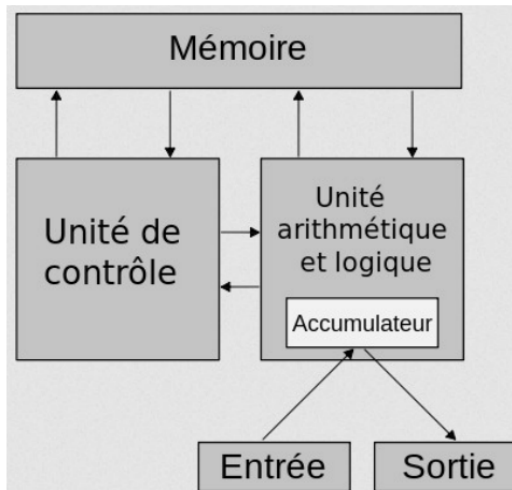
On reprend le code en français correspondant au calcul du nombre d'abonnés (ci-dessus). Traduire cet algorithme en Python.

## 4. Les machines

### 💡 **Fondamental** :

La machine permet l'exécution des programmes. Elles enchaînent les instructions simples, définies par le programme écrit en langage assembleur. Elles stockent des données (utiles au fonctionnement du programme ou sur demande de l'utilisateur). Enfin, elles gèrent les communications avec l'utilisateur et l'extérieur.

### ⊕ **Complément** :



Le modèle de Von Neumann (1956) définit l'organisation d'une machine dans laquelle les instructions du programme et les données (demandées ou produites) sont stockées au même endroit.

### 4.1. L'ordinateur

#### 💡 **Fondamental** :

L'ordinateur est un ensemble de circuits électroniques manipulant des données au format binaire. C'est un ensemble modulaire de composants électroniques.

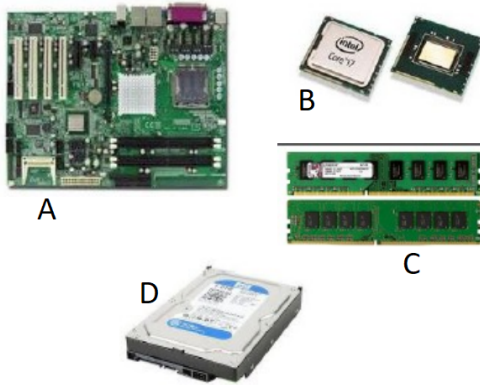
#### 📎 **Remarque** :

De nombreux objets de la vie courante sont aujourd'hui des ordinateurs : smartphone, calculatrice graphique, objets connectés...

Les principaux composants constituant un ordinateur sont les suivants

- Micro-processeur
- Mémoire vive
- Disque dur
- Carte mère
- Alimentation
- Ports de connexion
- Cartes d'extension (son, vidéo, réseau)

### 🔍 Exemple :



Associer chaque composant de l'image ci-contre à son nom dans la liste ci-dessus.

### 💡 Fondamental :

Les périphériques sont des appareils connectés à un ordinateur et permettant des interactions.

Les périphériques d'entrée permettent d'obtenir et de numériser de l'information.

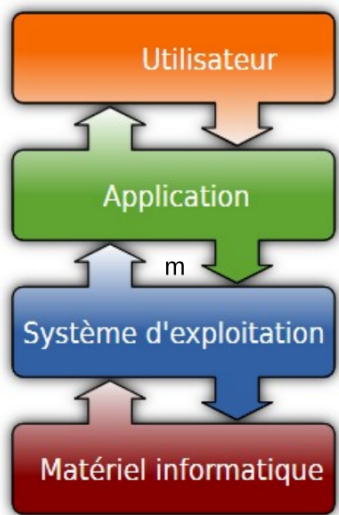
Les périphériques de sortie permettent de représenter et communiquer de l'information à l'utilisateur.

### 🔍 Exemple :

1. Citer cinq périphériques d'entrée.
2. Citer cinq périphériques de sortie.
3. Certains périphériques sont-ils à la fois périphériques d'entrée et de sortie ?

## 4.2. Le système d'exploitation

### 💡 Fondamental :



Le système d'exploitation est un programme informatique qui assure le lien entre utilisateur, matériel et applications.

Il gère l'accès aux ressources (mémoire vive, processeur)

Il gère les droits des utilisateur (administrateur, invités)

Il gère le système de fichiers (écriture sur le support de stockage)

Il fournit de l'abstraction au programmeur

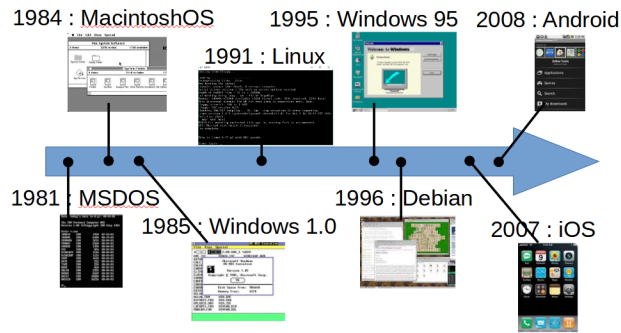
Voici une frise chronologique contenant différents systèmes d'exploitation.

On distingue les systèmes à interface en ligne de commandes (comme MSDOS) par opposition à ceux disposant d'une interface graphique (comme Windows 95).

Certains sont édités par une société (comme Windows de Microsoft) et d'autres sont libres (comme Linux)

Certains ciblent les dispositifs mobiles (smartphones, tablettes...), comme Android.

## Système d'exploitation : exemples



### Remarque :

Généralement, l'utilisateur installe et utilise des applications pour accomplir des tâches évoluées. Celles-ci utilisent les services fournis par le système d'exploitation pour fonctionner.

Quelques catégories d'applications courantes sont :

- Le navigateur Web
- Le lecteur multi-média
- L'éditeur de texte
- La suite bureautique
  - Le traitement de texte
  - Le tableur
  - L'éditeur de présentation...
- Le logiciel de retouche d'images
- Le logiciel de montage vidéo...

Pour chaque catégorie ci-dessus, il existe plusieurs applications différentes. Certaines sont éditées par des sociétés, d'autres sont libres.

### Exemple :

Pour chaque catégorie ci-dessous, citer deux noms d'applications correspondantes :

- Navigateurs Web
- Traitements de texte
- Tableurs
- Editeurs de présentations
- Logiciels de retouche d'images

## 5. Les interfaces

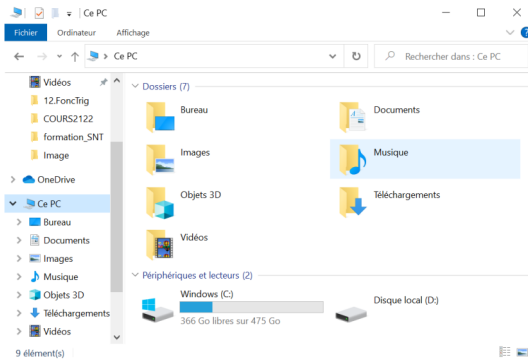
### Fondamental :

L'interface d'un programme permet la communication avec les humains, la collecte de données, la commande de systèmes.

On distingue deux types d'interfaces :

- Les interfaces en ligne de commande : l'utilisateur saisit des commandes au clavier qui sont exécutées par le système.
- Les interfaces graphiques, reprenant la métaphore du bureau (popularisé par Xerox en 1973)

## Exemple :



La commande suivante permet, sous Linux, de compter le nombre de fichiers MP3 présents dans le dossier *Musique* de l'utilisateur.

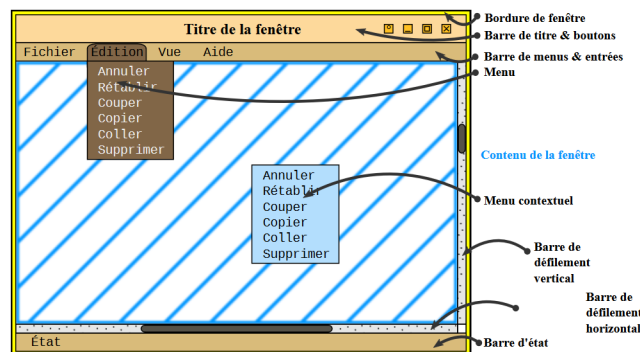
```
ls -l ~/Musique/*.mp3 | wc -l
```

Expliquer comment obtenir cette information avec l'interface graphique ci-contre.

## Remarque :

La plupart des logiciels offrent des interfaces graphiques bâties sur les mêmes principes.

Voici un peu de vocabulaire pour se repérer plus facilement.



## + Complément :

Dans la barre de menus, on trouve le plus souvent les menus suivants :

- Fichier : permet de gérer le fichier en cours d'édition
  - Nouveau : créer un nouveau fichier vierge
  - Ouvrir : rappeler un fichier précédemment créé
  - Enregistrer (ou Sauvegarder) : écrire le fichier sur le support mémoire à l'endroit sélectionné pour le conserver
- Edition
  - Défaire : annuler la dernière opération
  - Refaire : rétablir la dernière opération
  - Copier : stocker la représentation de l'objet sélectionné dans le presse-papiers (zone de mémoire temporaire)
  - Coller : créer un nouvel objet à partir du contenu du presse-papier
  - Couper : copier puis supprimer l'objet sélectionné
- Aide : pour obtenir de l'aide sur l'utilisation du programme



# III Exercices

- Exercice : Nombre de disquettes
- Exercice : Le premier disque dur
- Exercice : Technologies de stockage de masse
- Exercice : Exécution de programmes
- Exercice : Boucles for et while
- Exercice : Choix d'un ordinateur
- Exercice : Utiliser un environnement de travail informatisé

## 1. Exercice : Nombre de disquettes

Au début des années 90, les jeux et logiciels étaient vendus sous forme de disquettes 3,5 pouces. Chacune contenait 1,44 Mo de données. Le jeu Fortnite pèse 20 Go.

### Question

Combien de disquettes auraient été nécessaires pour le stocker ?

## 2. Exercice : Le premier disque dur

Le disque dur IBM « Ramac » était vu comme une révolution avec sa capacité de stockage de 3,75 Mo.

### Question 1

Combien faut-il de disques durs IBM « Ramac » pour avoir la même capacité de stockage qu'une carte MicroSD de 32 Go ?

Les dimensions du disque dur IBM « Ramac » étaient de 1,5 m × 1,7 m × 0,74 m.

Les dimensions d'une carte MicroSD de 32 Go sont de 1,1 cm × 1,5 cm × 1 mm.

### Question 2

Si on remplissait le volume occupé par un IBM « Ramac » avec des cartes MicroSD, quelle capacité de stockage pourrait-on obtenir ?

## 3. Exercice : Technologies de stockage de masse

Pour répondre, vous devrez effectuer des recherches sur le Web.

Actuellement, on utilise couramment deux types de périphériques de stockage internes dans les ordinateurs personnels : les SSD et les HDD. Rechercher les informations suivantes :

### Question

Vous répondrez sous la forme d'un tableau synthétique

- Que signifient les acronymes HDD et SSD ?
- Qu'est-ce qui différencie ces deux types de périphériques ?
- Quels sont leurs avantages et inconvénients respectifs ?
- Un cas d'utilisation typique ?

## 4. Exercice : Exécution de programmes

Dans cet exercice, on va s'entraîner à exécuter des programmes Python très simples.

### Question 1

On considère la fonction Python ci-dessous. Quelle est la valeur retournée par l'appel `fonc1(3)` ?

```
1 def fonc1(k):
2     r = 5*k + 8
3     return r
```

### Question 2

On considère la fonction Python ci-dessous. Quelle est la valeur retournée par l'appel `fonc2(3, 4)` ?

```
1 def fonc2(a,b):
2     c = a-b
3     d = a*b
4     return c*d
```

### Question 3

On considère la fonction Python ci-dessous. Quelle est la valeur retournée par l'appel `fonc3(-2)` ?

```
1 def fonc3(x):
2     if x > 0:
3         return 5*x+3
4     else:
5         return 3*x+5
```

### Question 4

On considère le script Python ci-dessous. Quelle est la valeur de la variable `d` à la fin de l'exécution ?

```
1 a = 3
2 b = 4
3 c = a*b
4 c = c + 1
5 d = a + c
```

### Question 5

On considère le script Python ci-dessous. Quelle est la valeur de la variable `a` à la fin de l'exécution ?

```
1 a = 4
2 b = 5
3 c = a + b
4 b = c + 1
5 a = a + b
```

### Question 6

On considère le script Python ci-dessous. Combien de tours sont effectués par la boucle ?

```
1 a = 4
2 while a < 20:
3     a = a + 5
```

On pourra compléter le tableau suivant :

Valeur de a	4	9			
a < 20 ? (vrai ou faux)	vrai				

## Question 7

On considère la fonction Python ci-dessous. Quelle est la valeur retournée par l'appel `func()` ?

```
1 def func():
2     x = 0
3     for i in range(1,6):
4         x = x + 3*i
5     return x
```

On pourra compléter le tableau suivant :

Valeur de i	1	2	3	4	5
Valeur de x	3	9			

## 5. Exercice : Boucles for et while

On souhaite écrire en Python une fonction `somme` qui prend en entrée un entier naturel  $n$  et fait la somme des entiers entre 1 (inclus) et  $n$  (inclus).

Par exemple, `somme(4)` vaudra  $1 + 2 + 3 + 4 = 10$ .

### Question 1

1. Compléter la fonction `somme` suivante.

```
1 def somme(n):
2     total = 0
3     for ajout in range(1, n+1):
4         total = .....
5     return .....
```

### Question 2


2. Dans cet exemple, on sait qu'on doit ajouter chaque nombre entre 1 et  $n$  au total. On a donc utilisé une boucle `for`.

Réécrire la fonction `somme` en utilisant une boucle `while`.


### Question 3

Bilan : peut-on toujours utiliser une boucle `for` plutôt qu'une boucle `while` ?

## 6. Exercice : Choix d'un ordinateur



PC n°1 - réf. ST44-198r  
Intel Celeron N4120 8 Go SSD 120 Go 15.6" LED Full HD Wi-Fi AC/Bluetooth Webcam (sans OS)  
★★★★★ 5 avis



PC n°2 - réf. SU38-145r  
Intel Core i7-10875H 32 Go SSD 480 Go + SSD 960 Go 15.6" LED Full HD 144 Hz NVIDIA GeForce RTX 2070 SUPER 8 Go Wi-Fi AX/Bluetooth Webcam Windows 10 Professionnel 64 bits

Voici deux ordinateurs portables ainsi que leurs principales caractéristiques.

### Question

1. Quel PC permet de jouer à des jeux vidéo récents ? Justifier.
2. Combien y a-t-il de supports de stockage de masse dans le PC n°2 ?
3. Mélanie pense acheter le PC n°1 ci-dessus. Elle est une utilisatrice novice. Lui conseillerez-vous ce choix ? Justifier.

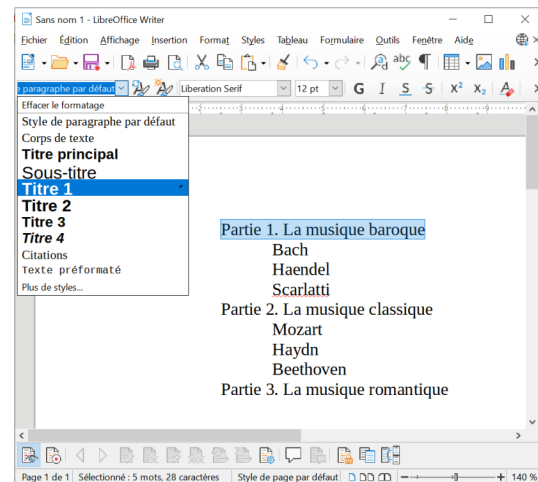
## 7. Exercice : Utiliser un environnement de travail informatisé

Marie doit rédiger un exposé sur différents courants musicaux ; baroque, classique et romantique.

Elle a commencé à taper son plan sur un traitement de texte.

Elle souhaite utiliser les styles afin que son document soit structuré.

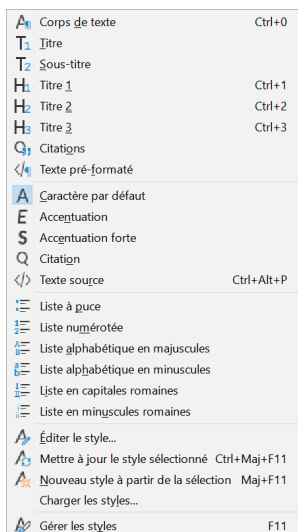
### Question 1



Elle commence par mettre ses trois titres de partie au format Titre 1 en suivant la méthode visible ci-contre.

Décrire en français cette méthode, en utilisant un vocabulaire adapté.

### Question 2



Dans chaque partie, elle souhaite écrire les noms des compositeurs sous la forme de liste non numérotées.

Le menu Styles est visible ci-contre. Après avoir sélectionné les paragraphes correspondants, quelle option doit-elle choisir ?

### Question 3

Au vu des deux captures d'écran ci-dessous, quelle étape importante Marie a-t-elle oublié de faire afin de garantir que son travail soit conservé ?

# IV Travaux pratiques

- Partie 1 : prise en main de l'environnement de travail
- Partie 2 : programmation en Python

Objectifs du TP :

- Se familiariser avec l'environnement de travail du lycée
- Editer et exécuter ses premiers scripts Python.

## 1. Partie 1 : prise en main de l'environnement de travail

### Exercice 1

1. Se connecter à la session

2. Ouvrir l'Explorateur de fichiers



3. Afficher les différents périphériques de stockage (locaux et distants) en cliquant sur  Ce PC

### Méthode :

Le lecteur réseau MA CLASSE vous permet d'aller récupérer les fichiers déposés par votre professeur (Documents en consultation).

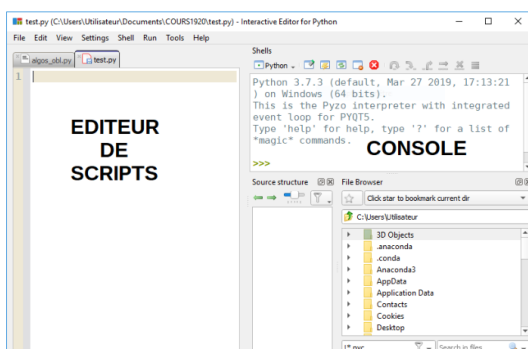
Le lecteur réseau ESPACE PERSONNEL permet d'accéder à votre espace personnel dans lequel vous travaillez.

Il faut copier-coller les fichiers à utiliser depuis MA CLASSE vers votre ESPACE PERSONNEL pour les modifier !

### Exercice 2

1. Dans votre ESPACE PERSONNEL, rendez-vous dans Devoirs.
2. Créer un dossier SNT
3. Dans ce dossier, créez un sous-dossier Ch0
4. Copiez les fichiers mis à disposition par votre professeur de SNT, dans MA CLASSE
5. Collez ces fichiers dans votre dossier Ch0.
6. Quel est l'espace mémoire occupé par ces fichiers ?

## 2. Partie 2 : programmation en Python



Pour programmer en Python, nous utilisons l'éditeur Pyzo.

L'interface est composée de deux parties principales, visibles ci-contre.

On écrit le code dans la partie de gauche, puis on lance le programme et on obtient les résultats dans la partie de droite.

**Conseil :**

Si Pyzo est en anglais, dans le menu Settings, ouvrir le sous-menu Select language puis choisir French. Cliquer sur OK et redémarrer Pyzo.

**Méthode :**

Pour exécuter un script, on procède comme suit :


1. On écrit le code (les instructions) dans l'éditeur de scripts (à gauche)
2. On exécute : dans le menu Exécuter, choisir Démarrer le script.
3. On appelle les fonctions avec les valeurs voulues pour obtenir les résultat, dans la partie de droite.

**Attention :**

Pour exécuter un script, le fichier correspondant doit toujours être enregistré !

Pour enregistrer : Fichier puis Enregistrer (ou File puis Save).

**Exercice 1**

1. Cliquer sur l'icône Pyzo  présent sur le bureau.
2. Si un message s'affiche à gauche (vous demandant de choisir un environnement), cliquer sur **use this environment**.
3. Sauvegarder le fichier sous le nom exo1.py dans votre dossier TP0.
4. Ecrire le code ci-dessous dans l'éditeur (à gauche)

```
1 def f(x):
2     resultat = 2*x - 7
3     return resultat
```

5. Exécuter : dans le menu Exécuter, sélectionner Démarrer le script.
6. Dans la console (à droite), écrire  $f(5)$  et valider (avec la touche Entrée). On doit obtenir 3.

**Exercice 2**

On reprend la fonction ci-dessus, correspondant au prix payé en fonction de l'âge dans un cinéma.

```
1 def calcule_prix(age):
2     if age >= 14:
3         prix = 9
4     else:
5         prix = 4.5
6     return prix
```

1. Dans Pyzo, créer un nouveau fichier ex2.py dans le dossier TP0.
2. Saisir le code ci-dessus.
3. Exécuter le script. Appeler la fonction avec différentes valeurs de age pour la tester.

**Exercice 3**

On reprend la fonction ci-dessus, correspondant au nombre d'abonnés après 24 mois pour le YouTubeur.

```
1 def nombre_abonnes_apres_2_ans(abonnes):
2     mois = 1
3     while mois <= 24:
4         abonnes = abonnes + 10 * mois
5         mois = mois + 1
6     return abonnes
```

1. Dans Pyzo, créer un nouveau fichier ex3.py dans le dossier TP0.
2. Saisir le code ci-dessus.

3. Exécuter le script. Combien d'abonnés aura-t-il au bout de 2 ans s'il avait 100 abonnés au départ ?

#### Exercice 4

Le YouTubeur (exercice précédent) se demande combien de mois il devra attendre pour dépasser un certain nombre d'abonnés (par exemple 1500).

On va écrire une fonction nommée `premier_depassement` et qui a pour paramètre le nombre d'abonnés à dépasser, appelé `cible`.

On considère qu'il démarre avec 100 abonnés au mois n°1.

1. Ecrire la fonction `premier_depassement` dans le fichier `ex3.py` en s'inspirant de la fonction `nombre_abonnes_apres_2_ans` ci-dessus.
2. Expliquer pourquoi il faut enlever 1 à la variable `mois` après la fin de la boucle.
3. Combien de mois doit-il attendre pour avoir 1500 abonnés ?
4. On peut demander la monétisation de ses vidéos à partir de 10000 abonnés. Combien devra-t-il attendre pour être monétisé ?

#### Exercice 5

l'indice de masse corporelle (IMC) est donné par la formule  $IMC = \frac{masse}{taille^2}$  avec la masse en kg et la taille en m. On souhaite écrire une fonction qui calcule et renvoie l'IMC à partir de la masse et de la taille.

1. (a) Quels sont les paramètres de la fonction IMC ?  
(b) Ecrire cette fonction IMC en langage Python.
2. (a) Créer et sauvegarder un nouveau fichier `ex5.py` et y écrire la fonction IMC.  
(b) Exécuter le script puis appeler la fonction dans la console afin de calculer l'IMC d'une personne de 60 kg et 1,65 m.  
(c) De même, calculer l'IMC d'une personne de 70 kg et 1,75 m.
3. Un IMC inférieur à 18 (exclus) correspond à la maigreur. Un IMC compris entre 18 et 30 (inclus) correspond à une corpulence normale. Un IMC supérieur à 30 correspond à du surpoids.

Ecrire une fonction `resultat_IMC` qui :

- prend en paramètre la masse et la taille d'un individu
- calcule l'IMC correspondant par un appel à la fonction IMC écrite plus haut
- renvoie maigreur ou normal ou surpoids en fonction du résultat

Les deux fonctions doivent être écrites dans le même fichier.

#### + Complément :

On peut écrire la fonction `resultat_IMC` avec deux `If... else...` imbriqués mais on peut utiliser la structure présentée ci-dessous.

	Algorithme	En Python
Ces instructions sont exécutées uniquement si la condition 1 est vraie.	Si Condition 1 → Instructions 1	if condition 1: Instructions 1
Ces instructions sont exécutées si la condition 1 était fausse et si la condition 2 est vraie	Sinon Si Condition 2 → Instructions 2	elif condition 2: Instructions 2
Ces instructions sont exécutées uniquement si aucune des conditions n'était vraie	... Sinon → Instructions Fin Si	... else: Instructions

#### Exercice 6

Le prix de vente unitaire d'un article décroît avec la quantité achetée. Ainsi,

- Si on achète 10 articles ou moins, le prix d'un article est de 10 euros
- Si on achète entre 11 et 50 articles (50 inclus), le prix d'un article est de 9,80 euros
- Si on achète plus de 50 articles, le prix d'un article est de 9,50 euros.

Ecrire une fonction `prix_total` qui prend en entrée le nombre d'articles achetés et renvoie le prix total à payer.

### Exercice 7

Le langage python possède de nombreuses bibliothèques pour une grande variété d'utilisations.

Par exemple, le module `math` offre des fonctions mathématiques, comme la racine carrée `sqrt` (abréviation de square root en anglais).

Pour pouvoir calculer des racines carrées dans un programme, il faut importer la fonction avec la ligne suivante :

```
from math import sqrt
```

On peut également importer l'ensemble des fonctions du module `math` en une seule fois (racine carrée, sinus, cosinus...)

```
from math import *
```

On fait cet import avant de définir les fonctions.

Ecrire une fonction `pyth` qui prend en argument les longueurs des deux côtés de l'angle droit d'un triangle rectangle et renvoie la longueur (approchée) de l'hypoténuse.

### Exercice 8

Pour calculer  $a^b$  en python, on écrit `a**b`.

1. Calculer  $3^2 + 4^2$  puis  $5^2$ . A-t-on égalité ?
2. Par un calcul à la main, justifier que  $\left(\frac{3}{7}\right)^2 + \left(\frac{4}{7}\right)^2 = \left(\frac{5}{7}\right)^2$ .
3. Tester cette égalité avec python. Que remarque-t-on ?

C'est un comportement normal et pas un bug. Vous étudierez les raisons l'an prochain en NSI.

### Exercice 9

La fonction `randint` du module `random` permet d'obtenir des nombres entiers aléatoires.

Ainsi, pour donner à `x` une valeur entière aléatoire comprise entre 1 (inclus) et 10 (inclus), on utilise la commande

```
x = randint(1,10)
```

1. Importer la fonction `randint` du module `random` puis écrire une fonction `somme` qui ne prend pas de paramètre en entrée, simule le lancer de deux dés et renvoie la somme des deux nombres obtenus.
2. Dans certains jeux, comme les petits chevaux, il faut faire 6 avec un dé pour commencer à jouer. Ecrire une fonction `simule_6` qui lance un dé jusqu'à obtenir 6 et retourne le nombre de lancers qui ont été nécessaires.
3. En moyenne, combien doit-on lancer un dé de fois avant de faire 6 ? Ecrire une fonction pour estimer ce résultat.

### Exercice 10

Une entreprise produit des boules en verre décoratives. Elle peut produire des boules à partir d'un rayon de 1cm avec une précision de 1mm. Par exemple, elle pourra produire une boule de 2,3cm de rayon, une boule de 2,4cm de rayon mais pas une boule de 2,35cm de rayon.

On se demande quel doit être le rayon minimal de la boule pour que son volume dépasse  $2000\text{cm}^3$  (2 litres).

Le volume  $V$  d'une boule de rayon  $r$  en cm est donné par  $V = \frac{4}{3}\pi r^3$ .



1. Ecrire une fonction `vol_boule` qui prend en entrée un rayon `r` et retourne le volume de la boule de rayon `r`.
2. Ecrire une fonction `rayon_min` répondant à la question initiale.
3. Modifier la fonction `rayon_min` pour qu'elle trouve le rayon minimal tel que le volume de la boule soit supérieur à  $n \text{ cm}^3$  (`n` est un paramètre de la fonction)