

## Activité : Premiers pas en Python

### Coder qu'est-ce que c'est ?

**Coder c'est avant tout communiquer avec un ordinateur pour lui donner des ordres.**

Malheureusement l'ordinateur ne parle pas notre langue : il ne comprends que le binaire, langage basé sur une succession de 0 et de 1, qui n'est pas agréable à utiliser pour un humain...

Les informaticiens ont donc développé des langages, simples pour l'utilisateur, qui seront ensuite traduits par des interpréteurs ou des compilateurs en langage machine. Chaque langage possède ses propres spécificités : certains langages vont être optimisés pour le calcul numérique alors que d'autres visent plutôt le traitement et l'analyse des données. Cela dit, certains langage comme Python sont multifonctions. Python est de plus en plus utilisé car il s'interface avec un très grand nombre d'autres langages. Ainsi, il peut être utilisé aussi bien pour commander des machines, faire des simulations numériques analyser des données, tracer des courbes, ou créer des logiciels comme ... Libreoffice, Google, Gimp!

**Question 1 :** Faites une petite recherche et expliquer en quelques lignes ce qu'est le binaire.

## 1 La philosophie du code

### ★ À retenir ♥

Coder c'est donner une suite logique d'**instructions** à la machine.

Un groupe d'instruction visant à réaliser une action simple peut être regroupé sous la forme d'une **fonction**.

Les fonctions sont généralistes et peuvent effectuer la même action dans différents contextes. Par exemple, la fonction *print* de python permet d'afficher ce que vous voulez à l'écran : *print('Hello world')* va afficher *Hello world* à l'écran. Cette phrase est l'**argument** de la fonction : c'est la variable sur laquelle doit agir la fonction. Une fonction peut avoir plusieurs arguments.

Vous pouvez coder vous-même toutes les fonctions dont vous avez besoin, mais c'est un travail laborieux que d'autres utilisateurs ont déjà fait avant vous. Les fonctions existantes sont regroupées dans des **bibliothèques** qui dépendent de l'utilité de la fonction. Par exemple la bibliothèque **numpy** contient tout ce qui permet de faire des calculs mathématiques et de gérer des tableaux, alors que la bibliothèque **matplotlib** contient tous les outils pour tracer des courbes.

Pour utiliser une fonction d'une bibliothèque il faut charger ou **importer** cette bibliothèque à l'aide de la commande **import** suivi du nom de la bibliothèque.

**Question 2 :** Citer des exemples de fonctions mathématiques que vous connaissez.

**Question 3 :** Soit la fonction diviser par 10 noté `div10` qui permet de diviser n'importe quel nombre par 10. Comment feriez vous pour faire l'opération  $3/10$ ? Quel est dans ce cas l'argument de la fonction ?

**⚠ Attention :** Une suite d'instruction en Python doit être logique et ordonnée car l'ordinateur lit et exécute les lignes une par une. Vous ne pouvez pas ajouter du sucre dans votre café, si vous n'avez pas fait le café!

De même, à l'inverse du langage courant où certaines données sont implicites (exemple : si vous être en cours de physique et que le professeur vous demande de sortir votre livre, vous sortez automatiquement votre manuel de physique et non Oui-Oui à la plage!), l'ordinateur ne connaît pas l'implicite, vous devez donc tout préciser!

**Question 4 :** Imaginez une suite d'instructions, permettant de lire des données situées dans un fichier texte appelé *test.txt* et d'afficher la 5ème ligne. Vous pouvez imaginer des noms de fonctions si vous le souhaitez.

Indice : parmi les fonctions à utiliser il y a *Open* et *readline* qui appartient à la bibliothèque *csv*.

⚠ la numérotation en python commence à 0 la première ligne est ainsi la ligne 0.

**Question 5 :** Voici un extrait de programme permettant de lire des données de marée mesurées en Martinique en avril 2019, en utilisant la fonction `help` de python commentez le script suivant :

```
#!/usr/bin/env python

import numpy as np
import matplotlib.pyplot as plt# pour les courbes
import csv
from datetime import datetime
plt.ion()

name = 'Tide_prediction_Martinique_avril2019.txt'

data = []
with open(name) as f:
    c = csv.reader(f, delimiter=' ', skipinitialspace=True)

    for line in f:
        data_line = line.rstrip().split('\t')
        data.append(data_line)

date = []
height = []
temps = []

for i in range(20, len(data)):
    height.append(np.float(data[i][3]))
    x = data[i][0] + '/' + data[i][2]
    date.append(x)

height = np.asarray(height)
for d in date:
    t = datetime.strptime(d, '%Y/%m/%d/%H:%M')
    temps.append(t)

plt.figure()
plt.plot(temps, height, 'b')
plt.xlabel('Temps')
plt.ylabel('Hauteur ')
plt.grid()
plt.show()
```

**Question 6 :** Quels sont les différents objets utilisés : repérez les boucles, les listes et les chaînes de caractères.

**Question 7 :** Que signifie le point séparant deux mots ? Trouver deux utilités différentes.

**Question 8 :** Que remarquez-vous au niveau du format du texte (indentation, retour à la ligne) ? Trouvez la règle et complétez la définition suivante :

### ★ L'indentation ♥

L'organisation des lignes est très importante puisque Python est un langage dit **indenté**.

L'indentation c'est l'espace entre la *marge* et le début du texte écrit.

Un morceau de texte A indenté de 4 espaces par rapport à une première ligne signifie que l'action A est ..... à la première ligne.

Des instructions avec ..... sont exécutées les unes à la suite des autres sans priorité.

Le retour à la ligne signifie .....

**Question 9 :** Choisissez une couleur par bibliothèque et surlignez de la même couleur les fonctions qui appartiennent à la même bibliothèque.

## 2 Les listes et les tableaux

Comme vous avez pu le voir dans l'exemple de code proposé précédemment, l'usage de `[]` définit une liste que vous pouvez ensuite remplir avec la fonction *append*. Il est aussi possible de définir directement une liste en utilisant des virgules pour en séparer les éléments : `[1, 2, 3, 4, 5]`.

**Question 10 :** Dans votre terminal python, tapez `a = [1, 2, 3, 4, 5]` puis appuyez sur entrer pour valider le fait que la variable `a` est une liste contenant les 5 premiers entiers. La commande `a[1]` permet d'afficher l'élément 1 de la liste. Tapez cette commande et commentez. Comment obtenir le nombre 1 issue de la liste ?

### ★ Listes et tableaux ♥

Il existe sous Python deux méthodes (objets) permettant de structurer les données : les listes (list, `[]`) et les tableaux (array appartenant à la bibliothèque numpy, ex : `np.array([1, 2, 3])`)

- **Les listes** sont des suites de variables (nombres, textes, ou listes) regroupées et ordonnées ensemble entre crochets et séparées par les virgules. Le nombre d'élément d'une liste peut augmenter : pour ajouter le nom d'un élève à la liste *classe*, il suffit d'écrire `classe.append('nom')`. Puisqu'une liste n'est pas forcément composée d'objets mathématiques on ne peut pas utiliser sur elle de fonctions mathématiques.
- **Le tableau** est un objet mathématiques qui regroupe des nombres séparés par des virgules. Par exemple le tableau `t = np.array([1, 2, 3])` regroupe les trois premiers entiers. Les tableaux sont soumis aux opérations mathématiques : si je tape `t * 2` je multiplie tous les éléments de `t` par 2 et j'obtiens `[2, 4, 6]` qui sera aussi un tableau. À l'inverse des listes, les tableaux ont une taille qui ne peut pas changer.

**Question 11 :** Lisez l'encadrer précédent et répondez aux questions suivantes :

1. Que va t'il se passer si j'essaie de multiplier par 3 : `[1, 2, 3, 4, 5]` ?
2. Que puis-je faire pour réussir à multiplier par 3 d'un seul coup les 5 premiers entiers ?
3. Quel est l'avantage des listes ?
4. Quel est l'avantage des tableaux ?

## 3 Les boucles

Les boucles font partie des outils les plus utilisés en programmation et permettent entre autre d'automatiser la répétition d'une opération simple. En python, l'indentation des boucles est primordiale !

Une boucle commence toujours par une condition d'entrée (de départ), continue par une action simple et doit se finir par une condition de sortie, sinon la boucle est infinie et le code ne s'arrête jamais de tourner.

En Python, les boucles commencent souvent par :

```
for i in range(5) :
```

**Question 12 :** Taper `range(5)` dans votre terminal python et décrivez ce que vous obtenez. Complétez ensuite la phrase suivante :

L'instruction `for` donne à la fois la ..... de ..... et de..... de la boucle : ici `i` va prendre toutes les ..... comprises dans l'objet `range(5)`.

De façon plus générale on peut écrire :

```
for i in objet :
```

L'*objet* doit être une liste de valeur ou de caractères que va prendre successivement `i`. Par exemple si `objet = ['oui', 'non']` `i` prendra d'abord la valeur 'oui' puis 'non'.

Il faut noter que le démarrage d'une boucle, tout comme la définition d'une fonction est suivi de :  
. Une boucle est initiée par l'instruction de début : `i = 0` et la condition d'arrêt `i = 10`.

### ★ Les types de boucles ♡

Les boucles peuvent être beaucoup plus complexes que celle présentée, et peuvent utiliser d'autres opérateurs. Les opérateurs logiques les plus courants sont :

- `for`
- `if` : exprime la condition.
- `else` : permet de mettre une autre action si la condition `if` n'est pas validée
- `elif` : contraction de `else + if`
- `while` : permet de continuer une action jusqu'à la vérification d'une condition

Exemples d'instruction `if` :

```
if n > 0 :  
if n == 5 : # n egale une valeur fixe  
if n != 5 : # n different de 5  
if n >= 5 : # n superieur ou egal 5  
if n <= 5 : # n inferieur ou egal 5
```

**Question 13 :** À l'aide d'une boucle et de la fonction `range` créer un code qui calcule la somme des 10 premiers entiers.

**Question 14 :** À l'aide d'une boucle et de la fonction `range(15)`, créer un code qui trie en deux listes de nombres pairs et impairs les 15 premiers entiers.

*Indice : `np.float(i)/2` fait intervenir la division classique et donne un nombre réel alors que `i/2` fait intervenir la division euclidienne (entière).*