

TP1: Rappel des bases de programmation Python:

1. Les variables et types de données:

Nous allons commencer par créer une variable x et lui affecter une valeur:

```
x=1
print(x)

1
x=2
print('x=',x)

x= 2
```

```
x+=1
print(x)

3
```

- Le type **int** (entier)

Ce type est utilisé pour stocker un **entier**, en anglais *integer*. Pour cette raison, on appelle ce type **int**.

X = 128

- Le type **float** (flottant)

Ce type est utilisé pour stocker des **nombres à virgule flottante**, désignés en anglais par l'expression *floating point numbers*. Pour cette raison, on appelle ce type : **float**.

X = 125.5

- Le type **str** (chaîne de caractères)

Sous Python, une donnée de type **str** est une suite quelconque de caractères délimitée soit par des apostrophes (simple quotes), soit par des guillemets (double quotes). str est l'abréviation de *string*, qui veut dire *chaîne* en français.

x= "Hello World!"

- **Le type bool (booléen)**

Le type bool est utilisé pour les *booléens*. Un booléen peut prendre les valeurs True ou False.

2. Les opérateurs de bases:

En plus d'attribuer une valeur initiale à une variable, nous pouvons effectuer les opérations mathématiques habituelles sur les variables

Les opérateurs de base en python incluent +, -, *, /, //, % et ** qui représentent respectivement l'addition, la soustraction, la multiplication, la division, la division entière, le module et l'exposant.

```
# opérations arithmétiques
x = 5
y = 2
print('x+y = ', x+y)
print('x-y = ', x-y)
print('x/y = ', x/y)
print('x//y = ', x//y) #division entière
print('x*y = ', x*y)
print('x^y = ', x**y) # x puissance y
print('x+y = ', x+y)
print('x%y = ', x%y) # Le reste de la division
```

```
x/y = 2.5
```

```
x//y = 2
```

```
x*y = 10
```

```
x^y = 25
```

```
x+y = 7
```

```
x%y = 7
```

```
# opérations de comparaison
```

```
x = 5
```

```
y = 2
```

```
print('égalité = ', x == y)
```

```
print('inégalité ', x != y)
```

```
print('inférieur ou égal ', x <= y)
```

```
print('supérieur ou égal', x >= y)
```

```
égalité = False
```

```
inégalité True
inférieur ou égal False
supérieur ou égal True
```

3. Les types de données construits

Sous python, il est possible d'élaborer de nouveaux types à partir des types de base (*int*, *float*, etc.) appelés types construits.

Parmi ces types:

- **Les listes et les tuples :**

Une liste ou un tuple peuvent contenir tout types de valeurs(int, float, bool, string). On dit que ce sont des structures hétérogènes.

La différence entre les deux est qu'une liste est mutable alors qu'un tuple ne l'est pas (On ne peut pas la changer après qu'il soit créé).

Les éléments d'une liste ou d'un tuple sont ordonnés et on y accède à un élément grâce à sa position en utilisant un numéro qu'on appelle l'**indice** de l'élément.

```
liste_1 = [1, 4, 2, 7, 35, 84]
villes = ['Paris', 'Berlin', 'Londres']
nested_list = [liste_1, villes] # une liste peut contenir des listes, on
appelle cela nested list
print(nested_list)

[[1, 4, 2, 7, 35, 84], ['Paris', 'Berlin', 'Londres']]
```

- **Les dictionnaires**

Un **dictionnaire** en Python va aussi permettre de rassembler des éléments mais ceux-ci seront identifiés par une **clé**. On peut faire l'analogie avec un dictionnaire de français où on accède à une définition avec un mot.

Contrairement aux listes qui sont délimitées par des crochets, on utilise des **accolades** pour les dictionnaires.

Je peux aussi accéder aux valeurs avec:

```
Print(my_dict['c'])
```

```
Print(my_dict.get('c'))
```

4. Les fonctions:

Les fonctions sous python sont définies comme suit:

```
def nom_fonction(liste de paramètres):  
    bloc d'instructions
```

5. Les structures de contrôle

Il existe 3 principales structures de contrôle pour créer des algorithmes:

- Les alternatives if/else
- Les boucles For
- Les boucles While

Alternatives if/ Else:

Cette structure permet de tester une séquence d'alternatives. Si une condition est respectée, alors les instructions qui la suivent sont exécutées et la structure de contrôle est stoppée. Sinon la condition suivante est testée.

```
z = 5  
if z % 2 == 0:  
    print('z est divisible par 2')  
elif z % 3 == 0:  
    print('z est divisible par 3')  
else: print('z n\'est divisible par 2 ni par 3')
```

Les boucles For:

Une boucle for permet de créer des algorithmes itératifs (qui effectuent une certaine tâche plusieurs fois de suite). Pour cela, la boucle parcourt tous les éléments d'un objet dit itérable. Il peut s'agir d'une liste, d'un dictionnaire, d'un tableau...

```
for i in range (0, 10):  
    print(i)
```

Les boucles while

Une While permet d'effectuer en boucle une action tant que la condition d'exécution est validée tant que la condition est vrai).

```
x = 0  
while x < 10:  
    print(x)  
    x += 1
```

Exercices:

Quizz:

```
1. my_tuple = (1, 2, 3)  
   my_tuple[0] = 4  
   print(my_tuple)
```

- (4, 2, 3)
- (1, 2, 3)
- Erreur
- (4, 2, 3, 1)

```
2. my_dict = {"a": 1, "b": 2, "c": 3}  
   result = my_dict.values()  
   print(result)
```

- {1, 2, 3}
- [1, 2, 3]
- {"a": 1, "b":2, "c": 3}
- Error

```
3. def add_numbers(a,b):  
    return a+b
```

```
result = add_numbers(3, 4) + add_numbers(1, 2)  
print(result)
```

- 5
- 8
- 10

➤ 12

```
4. def uppercase_text(text):  
    return text.upper()  
  
result = uppercase_text("Hello, World!")  
print(result)
```

- "Hello, world"
- "HELLO WORLD!"
- "Hello, world!"
- Error

Exercices instructions de bases:

1. Ecrire un programme qui demande l'année de naissance d'une personne puis il calcule et affiche l'âge de la personne.
2. Ecrire un programme qui demande à l'utilisateur de taper la largeur et la longueur d'un rectangle et qui affiche le périmètre et la surface.

Exercice: les structures conditionnelles

1. Ecrire un programme qui demande à l'utilisateur de saisir un nombre et puis en fonction de ce nombre saisi:
 - 6: affiche " Le personnage va à droite"
 - 4: affiche " Le personnage va à gauche"
 - 8: affiche " Le personnage va à droite"
 - 2: affiche " Le personnage va en bas"
 - Dans le cas d'un autre caractère, affiche: " Erreur de saisie, le personnage ne bouge pas"

Exercice: boucle For:

1. A votre naissance, votre grand père vous offre un compte bancaire. Ensuite, à chaque anniversaire, votre grand père vous verse sur votre compte 500DT auxquels il ajoute le triple de votre âge.

Par exemple, lorsque vous avez 4 ans, il vous verse 512DT.

Ecrire un programme qui permet de déterminer quelle somme vous aurez lors de votre nième anniversaire.