

TP1

Exercice 1

IDLE

Lancer Idle, version “ Python 3”. Il apparaît une fenêtre « Python Shell » qui est l’interpréteur. C’est dans cette fenêtre qu’on va travailler au tout début.

Taper dans l’interpréteur les expressions suivantes :

`2+5; 2*5; 2**5; 2.1/3.5; 2.2/3.5; 17.0/5.0; 17 // 5; 17 / 5; 17 % 5`

Il faut taper ces 9 expressions sur 9 lignes différentes, et sans point-virgule. La fenêtre Idle aura l’aspect suivant :

```
>>> 2 + 5
7
>>> 2 * 5
10
...
```

Exercice 2

Depuis l’interpréteur, on peut créer une nouvelle fenêtre : soit via “ CTRL-N ” soit via le menu Fichier (ou File). Cette nouvelle fenêtre est une fenêtre d’édition. Ce qu’on tape dedans est appelé à être sauvé (en fait IMMÉDIATEMENT) puis exécuté.

Créer une nouvelle fenêtre, sauvegarder le fichier dans votre répertoire personnel sous le nom **premieres_instruction.py**

Dans ce fichier taper les lignes suivantes :

```
2 * 5
print(3 * 5)
4 * 5
```

Sauvegarder le fichier en tapant control-S puis exécuter le par F5 ou à partir du menu. Que remarque t-on ?

On peut noter que lorsque l’on ferme l’interpréteur (le shell) tout ce qu’il contient est perdu alors que après avoir sauvegardé la fenêtre le fichier peut être réutilisé. Pour charger (ou rechargé) un fichier on utilise soit le menu soit l’instruction Control-O.

On note que le résultat d’une opération s’affiche immédiatement dans l’interpréteur alors qu’il faut utiliser la fonction **print(operation)** pour afficher ce résultat lorsque l’opération est réalisée dans la fenêtre.

Il est à noter que les noms de fichier ne doivent pas contenir d’accent et se terminent par **.py** cette terminaison

signifiant qu'il s'agit d'un fichier exécutable python.

Vérifier le en fermant la fenêtre d'édition puis chargez le fichier créé précédemment, corrigez les deux lignes en rajoutant print correctement, sauvegardez puis exécutez à nouveau le fichier.

Exercice 3

Dans l'éditeur tapez les deux lignes :

```
for i in range(2,5):  
    print(i**2)
```

Sauvegardez, exécutez et essayez d'expliquer ce que réalise cette boucle. Tapez la même boucle dans l'interpréteur.

Tapez dans l'interpréteur :

```
print(list(range(2,5))).
```

 Que remarque t-on ?

Exercice 4

Variables

Une variable est une zone mémoire de l'ordinateur :

Qui porte un nom.

Qui contient de l'information (une valeur).

On dit qu'on réalise une affectation d'une variable quand on lui fournit une valeur ou que l'on modifie la valeur. En Python, si l'on veut affecter la variable a de la valeur 62 on note **a=62**. On peut réaffecter à a une autre valeur, qui écrase la précédente.

```
>>> x = 5 * 3  
>>> y = 6  
>>> x + y  
21  
>>> x = 36  
>>> x + y  
42  
  
>>> x = 10  
>>> y = x  
>>> x = 15
```

Après ces affectations que valent x et y ? Devinez puis vérifiez.

Dans l'éditeur tapez les lignes suivantes :

```
x = 10
y = 15
z = x + y
x = y
y = z
print(x + y + z)
```

Que va t-il se passer. Exécutez le fichier et vérifiez.

Quelles ont les valeurs de x et de y après ces instructions ?

```
x=42
y=10
x=y
y=x
```

Prédire puis vérifier dans l'interpréteur.

Même question avec :

```
x=42 y=10 z=x x=y y=z, tapez ces instructions sur des lignes différentes.
```

Même question avec : x=42 y=10 x,y=y,x

Exercice 5

Exécuter les commandes suivantes dans l'interpréteur :

```
t = [6, 12, 'a'] ; t[1]; t[3]; t[0]; t[-1]; len(t); t[1] = 2; t; t+t
```

en remplaçant les ; par un passage à la ligne.

Même question avec :

```
x = "truc"; y = "bidule"; x[1]; y[2]; x[4]; x[-1]; z = x+y;
z; t = x+" "+y; t; print(t); print(x,y); print(x+y); x[2] = "z"; x
```

Exercice 6

Boucles

Tapez dans l'éditeur puis exécutez :

```
for i in range(10,15):
    print("bonjour ",i)
```

Remplacer `print("bonjour ",i)` par `print("bonjour "+str(i))` et exécuter.

Remplacer `print("bonjour ",i)` par `print("bonjour "+i)`, exécuter.

Que remarque dans le dernier cas.
Même question avec :

```
somme = 0
for i in range(1,5):
    somme = somme + i
```

Après exécution, que valent i et $somme$?

Valeur de $somme$ dans ce qui suit :

```
somme = 0
for i in range(10,18):
    if i%3 == 0:
        somme = somme + i
```

Remarque : if $i\%3 == 0$ signifie que la valeur de i doit être un multiple de 3 pour être prise en compte.

Pour terminer calculer $\sum_{k=10}^{100} k^2$ en écrivant une boucle dans l'éditeur.

Exercice 7

Tapez dans l'interpréteur : $\cos(\pi/3)$ que remarque t-on ?
taper dans l'interpréteur :
from math import cos, pi
puis réessayer le calcul précédent.

Exercice 8

charger le module cmath :
from cmath import *

définir le nombre complexe i par $i = \text{complex}(0, 1)$ puis calculer $z = \left(\frac{1 + i\sqrt{3}}{1 - i}\right)^2$. Calculer le module (abs) et un argument (phase) de z .

Exercice 9

Un nombre binaire s'écrit 0b101101 c'est à dire zéro suivi de 0 et du nombre binaire, l'opération inverse est bin(42) mais dans ce dernier cas l'élément fourni est une chaîne de caractère.

0o4578 nombre 4578 en octal, conversion inverse par oct

0x4A2E58 nombre 4A2E58 en hexadécimal, conversion inverse par hex.