

TP1 : PREMIERS PAS EN PYTHON

Pour les trois premiers exercices, essayez de donner la réponse avant de taper le code et de vérifier. Il s'agit de s'assurer ici que l'on a bien compris l'affectation des variables et les différents types de base.

Exercice 1 : Que sera-t-il affiché à l'écran après l'exécution du code suivant :

```
a = 5
b = a + 7
a = 10
print(b)
```

Exercice 2 : Quel sera le type de chacune des expressions suivantes ?

```
print(type(5))
print(type("abc"))
print(type(True))
print(type(5.5))
print(type(12/27))
print(type(2.0/1))
print(type(12 ** 3))
print(type(5 == "5"))
```

```
a = str((-4 + abs(-5) / 2 ** 3) + 321 - ((64 / 16) \% 4) ** 2)
print(type(a))
```

Exercice 3 : Que sera-t-il affiché à l'écran après l'évaluation des chacune des expressions suivantes :

```
print(5 == 5.0)
print(int(1/2))
print(float(1/2))
print(5 == "5")
print("sdf" != "sdf")
print(True and (False or not True))
print(str(53) + str(True))
```

```
a = 20
print(15-(a-15), ",", ",")
a = 10
print(15-(a-15),,)
```

Exercice 4 : Manipulation des chaînes de caractères et exemples de fonctions sur les chaînes de caractères

Exécutez le code suivant et analysez le.

```
print (5+3)

s='un exemple de chaine'

s2="un autre exemple"

print(s[1]) # Acces au caractere d'indice 1 (les indices commencent a zero)

print(s[0],s2[0])

print(s[4],s2[0])

print(s + ' et ' + s2) # Concatenation de chaines

s3=s + ' et ' + s2

s2*2

print('La taille de s est :', len(s))

print(s3[0:3]) # Recuperation des caracteres de position entre les 0 et 3e

print(s3[4:8])

print(s3[:3]) # Recuperation des 3 premiers caracteres

print(s3[3:]) # Recuperation des caracteres a partir de la position 3
```

Exercice 5 : Boucles et conditions

A partir de cet exercice, vous pouvez saisir vos instructions dans un fichier avec l'extension .py et exécuter ensuite ce fichier.

Attention : En Python il n'y a pas, comme dans certains langages, d'accolade ouvrante ou fermante pour délimiter un bloc d'instructions. Les blocs d'instructions en python sont délimités par “:” puis des tabulations : toutes les instructions consécutives à un “:” et débutant par un même nombre de tabulations appartiennent à un même bloc d'instructions.

Tapez les codes suivants et observez le résultat.

1. Boucle for

```
for i in range(10): # Ne pas oublier les deux points!!
    x = 2 # Attention ne pas oublier une tab. en debut de ligne sinon erreur!!!
    print(x*i) # Ne pas oublier la tabulation en debut de ligne!!
# Tapez encore une fois <Enter> si vous etes en ligne de commande
```

2. Boucle while

```

a=0
while(a<12): # Ne pas oublier les deux points!!
    a=a+1 # Ne pas oublier la tabulation en debut de ligne!!
    print(a, a**2,a**3) # Ne pas oublier la tabulation en debut de ligne!!
# Tapez encore une fois <Enter> si vous etes en ligne de commande

```

3. Condition If/Then/Else

```

a=0
if a==0: # Ne pas oublier les deux points!!
    print('0') # Ne pas oublier la tabulation en debut de ligne!!
elif a==1: # Ne pas mettre de tabulation et ne pas oublier les deux points!!
    print('1') # Ne pas oublier la tabulation en debut de ligne!!
else: # Ne pas mettre de tabulation et ne pas oublier les deux points!!
    print('2') # Ne pas oublier la tabulation en debut de ligne!!
# Tapez encore une fois <Enter> si vous etes en ligne de commande

```

Exercice 6 : Listes

Tapez chacune des instructions suivantes et observez le résultat.

```

list=['lundi', 2, 'janvier']
print(list)
print(list[0])
list[-1]
print(list[2])
len(list)
list[2]
del list[0]
list
list.insert(0,'mardi')
print(list)
print(list.index("mardi"))

```

Exercice 7 : Variables et affectations

On considère la liste d'affectations suivante (on considère ici que les variables ont correctement été déclarées avant d'être affectées).

1. $a \leftarrow 8$
2. $2 \leftarrow a$
3. $1 + 2 \leftarrow a$
4. $y \leftarrow 5 + 3$
5. $a \leftarrow 1 + 2$
6. $b \leftarrow a + 5$
7. $a \leftarrow a + 5$

Taper la suite de commandes correspondantes dans l'environnement Python. Vous regarderez attentivement les erreurs qui sont renvoyées par Python.