



## Fiche d'exercices : Types en python

Donner les types des valeurs suivantes : `13` , `14.5` , `'Hello World!'` , `True` , `'15.5'`

À l'aide de python, écrire un programme qui affiche dans le terminal la table de vérité de la fonction booléenne `xor` .

Écrire un programme qui permet d'afficher la somme de deux nombres entiers de la forme 'La somme est x+y' avec x et y défini précédemment.

Améliorer le programme précédent pour qu'il affiche 'La somme de x et y est x+y'.

Écrire un programme qui étant donné deux valeurs `cout_de_production` et `prix_de_vente`, affiche dans le terminal `profit` si le cout est inférieur au prix de vente, `perte` sinon.

Écrire un programme qui lit deux chaînes de caractères, les concatène et affiche le résultat sous la forme `'La chaîne résultante est : [résultat]'` .

Écrire un programme qui prend une note sur 20 et affiche dans le terminal si l'étudiant a obtenu une mention :

- `'Très bien'` pour une note supérieure ou égale à 16.
- `'Bien'` pour une note entre 14 et 15.
- `'Assez bien'` pour une note entre 12 et 13.
- `'Passable'` pour une note entre 10 et 11.
- `'Échec'` pour une note inférieure à 10.

Écrire un programme qui affiche dans le terminal si un nombre est pair ou impair.

Écrire un programme qui affiche dans le terminal tous les nombres entre 1 et 100. (à l'aide d'une boucle `for` puis d'une boucle `tant que`).

Écrire un programme qui affiche dans le terminal tous les nombres pairs entre 1 et 100.

Écrire un programme qui prend un nombre entier et affiche sa table de multiplication (jusqu'à 10).

Écrire un programme qui lit une chaîne de caractères et affiche le nombre de voyelles présentes dans cette chaîne.

Écrire un programme qui calcule la somme des chiffres d'un nombre donné par l'utilisateur (ex : pour 123, il affichera 6).

À l'aide de python, évaluer les résultats de ces comparaisons avec  $x = 5$ ,  $x = 10$  et  $x = -20$

1.  $x < 20$  and  $x > -20$
2.  $x > 5$  or  $x < 3$
3. `not (x == 10)`
4. `x >= 0 and x <= 10`
5. `x % 2 == 0 or x % 3 == 0`
6. `x < 0 or (x > 0 and x % 2 != 0)`

Écrire un programme qui prend deux variables `longueur` et `largeur` et qui calcule l'aire d'un rectangle en affichant le résultat sous la forme : `'L'aire du rectangle est : [aire]'`.

Évaluer les valeurs de result à la fin de chaque programme.

```
x = 8
y = 6
if x > y:
    result = x - y
else:
    result = y - x
```

```
a = 4
b = 9
if a < b:
    result = a + b
else:
    result = a * b
```

```
m = 7
n = 3
if m % 2 == 0:
    result = m * n
else:
    result = m + n
```

Écrire une fonction python `factorielle` qui prend en paramètre un nombre entier  $n$  et renvoie la factorielle de ce nombre. On rappelle que la factorielle de 5 vaut  $1 \times 2 \times 3 \times 4 \times 5$ .

Tester cette fonction avec les valeurs 5, 6, 8 et 10.

Écrire une fonction python `aire_rectangle` qui prend en paramètres deux entiers correspondant à la largeur et la longueur d'un rectangle et renvoie l'aire de ce rectangle.

Écrire une fonction `moyenne` qui prend deux nombres en paramètre et renvoie la moyenne de ces deux nombres.