

Les bases de python

1 Affichage simple avec des print

1. Écrire une fonction appelée `triangle`, sans paramètre, qui affiche le motif suivant :

```
*  
**  
***  
****  
*****
```

2. Écrire une fonction appelée `variables`, sans paramètre, qui définit 3 variables appelées `titre`, `auteur` et `prix` (valeurs de votre choix) et qui affiche leurs valeurs.
3. Écrire une fonction appelée `bonjour`, qui prend 1 paramètre appelé `nom` et affiche "Bonjour" suivi du nom donné en paramètre.

2 Opérateurs et conditions

1. Écrire une fonction `somme`, qui prend 2 paramètres et renvoie leur somme.
2. Écrire une fonction `conversion`, qui prend 1 paramètre `distance` exprimé en kilomètres, et renvoie sa conversion en mètres. Afficher le résultat pour le paramètre `distance=442`.
3. Écrire une fonction `perimetre`, qui prend 2 paramètres `l` et `h` et la valeur du périmètre d'un rectangle de longueur `l` et de hauteur `h`. Utiliser la fonction `somme` écrite précédemment. Afficher le résultat pour les paramètres `l=2` et `h=3`.
4. Écrire une fonction `equilateral`, qui prend 3 paramètres représentant les longueurs des côtés d'un triangle, et qui renvoie `True` si le triangle est équilatéral.
5. Écrire une fonction `plus30`, qui prend 2 paramètres et renvoie `True` si l'un des deux est supérieur ou égal à 30. Afficher le résultat pour les paramètres 10 et 29.
6. Écrire une fonction `proche100`, qui prend 2 paramètres et renvoie celui qui est le proche de 100. Afficher le résultat pour les paramètres 66 et 89.
7. Écrire une fonction `taille_mots`, qui prend 2 paramètres qui sont des mots, et affiche celui qui contient le plus de lettres.

Note La taille d'un mot est donnée par : `len(mot)`

8. Écrire une fonction `mention`, qui prend un paramètre `note` et affiche :
 - (a) "Très Bien" si $note \geq 16$,
 - (b) "Bien" si $16 > note \geq 14$,
 - (c) "Assez Bien" si $14 > note \geq 12$,
 - (d) "Pas de mention" dans les autres cas.
9. Écrire une fonction `top3`, qui prend 3 paramètres et renvoie la valeur du plus grand. Afficher le résultat pour les paramètres 12, 66 et 3.

3 Boucles

1. Écrire une fonction puissance, qui prend 2 paramètres a et b et renvoie la valeur de a^b .
2. Écrire une fonction puissance, qui prend un paramètre n et qui pour tous les nombres i compris entre 1 et n :
 - (a) affiche "Fizz" si i est divisible par 3,
 - (b) affiche "Buzz" si i est divisible par 5,
 - (c) affiche "FizzBuzz" si i est divisible par 3 et 5

Note : pour parcourir les nombres de 1 à n vous pouvez utiliser : `for i in range(1,n):`

3. Reprendre la question 1.1 en utilisant des boucles.
4. Écrire une fonction appelée table_n, qui prend un paramètre n et affiche la table de multiplication de n.
Exemple pour n=3 :

```
Table de 3
-----
0x3 = 0
1x3 = 3
2x3 = 6
3x3 = 9
4x3 = 12
5x3 = 15
6x3 = 18
7x3 = 21
8x3 = 24
9x3 = 27
```

5. Écrire une fonction appelée leet qui prend un mot en paramètre, et affiche ce mot en remplaçant :
 - (a) les "E" par des 3,
 - (b) les "S" par des 5,
 - (c) les "L" par des 1,
 - (d) les "O" par des 0,
 - (e) les "T" par des 0,
 - (f) les "A" par des 4.

Afficher le résultat pour l'entrée "DES CHIFFRES ET DES LETTRES".

Note : pour parcourir les lettres d'un mot vous pouvez utiliser : `for lettre in mot`

Point culture : « Le *leet speak* (en leet speak : 1337 5|*34| <), de l'anglais « elite speak » est un système d'écriture utilisant les caractères alphanumériques ASCII d'une manière peu compréhensible pour le néophyte (appelé noob et déclinaisons) pour s'en démarquer. Le principe est d'utiliser des caractères graphiquement voisins des caractères usuels, par exemple « 5 » au lieu de « S », « 7 » au lieu de « T » et, pour les fanatiques, « |_| » au lieu de « U » ou « | < » au lieu de « K », sans respect de l'orthographe ou des majuscules. »¹

1. https://fr.wikipedia.org/wiki/Leet_speak