

Introduction à Python

Tous les codes sont sur <http://tableauxmaths.fr/spip> dans la classe de première Spé Maths.

I Affichage : print()

```
1 5 + 6
2 #Ceci est un commentaire
3 print(3 + 4)
4 print("3" + "4")
5 print("coucou" + "toto")
6 print("3+4", "donne :", 3+4)
```

```
7
34
coucoutoto
3+4 donne : 7
```

- ☞ La ligne 1 n'est pas affichée mais le calcul est effectué.
- ☞ La ligne 2 est un **commentaire**. Cette ligne n'est pas considérée par l'ordinateur.
- ☞ La ligne 3 est affichée. `print()` permet d'afficher dans la console une valeur ou le résultat d'un calcul.
- ☞ Dans la ligne 4, "3" et "4" ne sont pas des nombres mais des **strings** (*Chaînes de caractères*.)
- ☞ Dans les lignes 4 et 5, l'addition de chaînes de caractères donne une **concaténation**.
- ☞ La ligne 6 montre comment afficher plusieurs valeurs sur une même ligne à l'aide des virgules.



Corriger la ligne de code suivante afin d'afficher une phrase correcte :

```
print("9"+"7", "donne :", 9*7) .....
```

II Variables et opérations

```
1 a = 5
2 b = a + 1
3 produit = a*b
4 puissance2 = b**2
5 print(a, b)
6 print(produit, puissance2)
```

```
5 6
30 36
```

- ☞ a, b, produit, et puissance2 sont des **variables** (*des valeurs stockées en mémoire avec un nom*).
- ☞ On peut utiliser des **variables** déjà existantes pour en créer d'autres.



Écrire un programme avec trois nombres stockés dans des **variables** a, b et c et qui affiche la somme des trois valeurs puis le produit des trois valeurs.

III Les tests if et else

```
1 a = 5
2 b = 4
3 if a < b:
4     print("a est plus petit que b")
5 if a > b:
6     print("a est plus grand que b")
7 if a == b:
8     print("a et b sont identiques")
9 else:
10    print("a et b sont differents")
```

☞ `if a == b:` permet de tester une égalité. Il faut un double `==` et `:` à la fin.

☞ Si le test est vrai, les conséquences du test doivent être **indentées**, décalées vers la droite de 4 espaces.

☞ Il peut y avoir plusieurs lignes indentées. On peut utiliser la touche **tabulation** pour l'**indentation** :



```
a est plus grand que b
a et b sont differents
```

☞ `else:` peut être traduit par « *sinon faire ...* ».



Écrire un programme avec trois nombres stockés dans des **variables** a, b et c et qui affiche le plus grand des trois nombres. Il est possible d'utiliser les mots **and** et **or** dans les tests. Par exemple :

```
if a > b and a > c:...
```

IV La boucle for

```
1 for mon_compteur in range(5):
2     print(mon_compteur, "*", mon_compteur)
3     print(mon_compteur**2)
4 print("Fin de la boucle")
```

- ☞ `for i in range(5):` peut se traduire par :
« Pour *i* allant de 0 à 4 faire : »
- ☞ Le **compteur** de la boucle *i* parcourt les valeurs :
0; 1; 2; 3 puis 4.
- ☞ La fin de l'**indentation** signifie la fin du contenu de la boucle.

```
0 * 0
0
1 * 1
1
2 * 2
4
3 * 3
9
4 * 4
16
Fin de la boucle
```



Écrire un programme qui affiche la table de 7 jusqu'à 700 comme dans le résultat ci-contre :

```
7*0 = 0
7*1 = 7
7*2 = 14
...
7*100 = 700
```

V Les fonctions

- ☞ En programmation, les **fonctions** permettent de réutiliser des portions de codes plusieurs fois.
- ☞ Il suffit de **définir** une seule fois ce que va faire la fonction :
`def ma_fonction():...`
- ☞ Une fonction peut demander des **paramètres**.
Ici il y a 3 paramètres : a, b et c.
- ☞ Le contenu d'une fonction doit être **indenté**.
- ☞ La fin de l'**indentation** signifie la fin du contenu de la fonction.
- ☞ **return** permet de **renvoyer** une valeur et l'exécution de la fonction s'arrête.

```
1 import math
2
3 def racines(a, b, c):
4     delta = b**2 - 4*a*c
5     x1 = 0
6     x2 = math.sqrt(9) #Racine carree
7     if delta == 0:
8         print("Une seule racine")
9         return x1
10    print("Deux racines")
11    return(x1,x2)
12
13 print(racines(1,2,1))
14
15 print(racines(1,1,1))
```

```
Une seule racine
0
Deux racines
(0, 3.0)
```

Compléter la fonction **racines(a, b, c)** afin qu'elle calcule le discriminant d'un polynôme de degré 2 et renvoie :



- ☞ Les deux racines si delta est strictement positif
- ☞ La racine si delta est nulle
- ☞ "Pas de racines réelles" si delta est strictement négatif

Voir le devoir à la maison n°1 sur <http://tableauxmaths.fr/spip>