

## Variables, opérations, affichage, saisie

Pour **afficher** quelque chose, on utilise la fonction `print(chose_a_afficher)`.

On peut afficher plusieurs choses en les séparant par une virgule `print(chose1, chose2, ...)`.

Ex: `print("Hello World")`

Ex: `print("La somme de 3 et 6 vaut", 3 + 6)`

Une **variable** est un emplacement mémoire contenant une valeur, que l'on peut lire ou modifier. Pour créer et initialiser une variable, on utilise l'opérateur d'affectation `=`. On fait `nom_de_la_variable = valeur`.

Ex: `age_eleve = 16`  
`print(age_eleve) # Affiche 16`

Pour modifier une variable, on fait utiliser le même opérateur :

Ex: `age_eleve = 16 # Initialisation`  
`age_eleve = 17 # Modification`  
`age_eleve = age_eleve + 2 # Modification en utilisant la valeur précédente`  
`print(age_eleve) # affiche 19`

(Les **commentaires** en python commencent par une astérisque `#`. Ceux ne sont pas des instructions et donc n'ont pas à être écrits en python. On peut les écrire sur une ligne seul ou à la suite d'une instruction.)

Pour les variables contenant des nombres, deux **types** sont possibles : les nombres **entiers** `int` (pour integer) ou les nombres **décimaux** `float` (pour floating point, virgule flottante).

On peut connaître le type d'une variable avec la fonction `type(variable)`.

```
age_eleve = 16
print(type(age_eleve)) # affiche int
age_eleve = 16.5 # Le point sépare les parties entière/décimale
print(type(age_eleve)) # affiche float
```

Attention, le type `float` ne donne très souvent qu'une approximation d'un nombre décimal en base 10.

```
print(0.2 + 0.1) # affiche 0.30000000000000004
print(0.2 + 10.1) # affiche 10.299999999999999
```

Pour demander une **saisie clavier** à l'utilisateur, on utilise la fonction `input("question")`.

Le résultat est toujours une variable de type texte (`str` pour string), appelé aussi **chaîne de caractères**.

Une chaîne de caractères est entouré de simples `'...'` ou de doubles guillemets `"..."`.

Ex: `age = input("Quel age as tu ?")`  
`print(age) # affiche ce que l'utilisateur a saisi`  
`print(type(age)) # affiche str`

On va **convertir** le texte en entier avec `int(n)` ou en décimal avec `float(n)` pour pouvoir calculer avec.

```
age = int(age) # convertit si possible, erreur sinon
print(type(age)) # affiche int
```

### Opérations sur les variables numériques :

| Opération                             | Opérateur | Exemple                 |                    |
|---------------------------------------|-----------|-------------------------|--------------------|
| Addition                              | +         | <code>a = 8 + 5</code>  | <code># 13</code>  |
| Soustraction                          | -         | <code>a = 8 - 5</code>  | <code># 3</code>   |
| Multiplication                        | *         | <code>a = 8 * 5</code>  | <code># 40</code>  |
| Division décimale                     | /         | <code>a = 8 / 5</code>  | <code># 1.6</code> |
| Quotient de la division entière       | //        | <code>a = 8 // 5</code> | <code># 1</code>   |
| Reste de la division entière (modulo) | %         | <code>A = 8 % 5</code>  | <code># 3</code>   |
| Puissance                             | **        | <code>A = 10**2</code>  | <code>#100</code>  |

### Exercice 1 : Prédit, run, investigate

Sans exécuter, qu'affiche les codes suivants ? Vérifier en exécutant le code.

|                                 |  |  |   |
|---------------------------------|--|--|---|
| <pre>a = 4 a = 1 print(a)</pre> | <pre>a = 4 b = 7 a = b print(a, b)</pre> | <pre>a = 4 b = 7 b = a print(a, b)</pre> | <pre>a = 2 b = 4**a a = a**(b//4) print(a, b)</pre> |
|---------------------------------|--|--|---|

### Exercice 2 : Prédit, run, investigate

Sans exécuter, qu'affiche les codes suivants ? Vérifier en exécutant le code.

|  |  |
|--|--|
| <pre>n1 = 10 n2 = 20 print(n1 + n2) print(type(n1 + n2))</pre> | <pre>s1 = "encore un " s2 = "test" print(s1 + s2) print(type(s1 + s2))</pre> |
| <pre>n1 = 2 n2 = 2.0 print(n1 + n2) print(type(n1 + n2))</pre> | <pre>s = '9' print(s, type(s)) n = int(s) print(n, type(n))</pre>            |

### Exercice 3 : Prédit, run, investigate

Sans exécuter, qu'affiche les codes suivants ? Vérifier en exécutant le code. Que calculent ces programmes ?

|   |   |
|---|---|
| <pre># Programme mystère 1 n1 = 10 n2 = 20 m = (n1 + n2) / 2 print(m)</pre> | <pre># Programme mystère 2 c = 10 P = c * 4 A = c**2 print(c, P, A)</pre> |
|---|---|

### Exercice 4 : Modify

- 1) En modifiant le programme mystère 1 de l'exercice 3, écrire un programme qui calcule et affiche la moyenne de 3 nombres. Modifier le nom des variables pour que le code soit plus explicite.
- 2) En modifiant le programme mystère 1 de l'exercice 3, écrire un programme qui calcule et affiche l'aire et le périmètre d'un rectangle. Modifier le nom des variables pour que le code soit plus explicite.
- 3) Modifier vos deux précédents programmes pour que les valeurs utilisées pour les calculs soient demandées puis saisies par l'utilisateur.

### Exercice 5 : Make

- 1) Écrire un programme qui demande à l'utilisateur un nombre et affiche ce nombre divisé par 3.
- 2) Écrire un programme qui demande à l'utilisateur deux nombres n et d puis affiche le quotient et le reste de la division euclidienne de n par d. Le programme doit afficher : '<n> = <d> \* <quotient> + <reste>'
- 3) Écrire un programme qui demande à l'utilisateur son nom son age et affiche l'année de son départ à la retraite. Il faut avoir 62 ans pour pouvoir partir à la retraite. Le programme doit afficher : 'Bonjour <nom>, vous avez <age> donc vous serez retraité en <age\_retraite>.'
- 4) Écrire un programme qui demande 4 valeurs a1, b1, a2, b2 tel que l'on ait deux fonctions affines sécantes non confondues d'équations  $y = a1 * x + b1$  et  $y = a2 * x + b2$  puis affiche les coordonnées de leur point d'intersection. L'abscisse du point d'intersection est  $x = (b2 - b1)/(a1 - a2)$