



CALCULS ET VARIABLES AVEC PYTHON

Réaliser des calculs dans la console (shell), les types int et float, connaître des fonctions de calcul intégrées à Python, connaître et utiliser les modules maths et random.

Affecter une valeur à une variable, vérifier son type, connaître et manipuler les variables de type str, les commandes print et input.

1. CALCULS:

✓ ACTIVITE:

Entrez dans la console de Python la série d'instructions suivantes en validant par ENTREE.
Observez les résultats obtenus puis complétez le tableau « aide mémoire ». attention le symbole >>> n'est pas à écrire c'est une entête indiquant qu'on est sur une nouvelle ligne.

Calculs simples, priorités :

```
>>> 98 | >>> 98+2*5 | >>> (98+2)*5 | >>> 5(98+2) | >>>5*-2 | >>>3,5+1
```

```
>>>3.5+1
```

```
>>> 5**2 | >>> 5**3 | >>> 2**5 | >>> 2***5 | >>> 10E2 | >>> 5E3 | >>> 5E-3
```

Quotients:

```
>>> 5/8 | >>> 5/9 | >>> 5/(-8) | >>> 5/-8 | >>> 5/8 | >>> 25/5 | >>> 25//5 | >>> 23/5
```

```
>>> 23//5 | >>> 23%5 | >>> 127//10 | >>> 127%10
```

Type de nombres:

```
>>> type(3) | >>> type(3.2) | >>> type(10) | >>> type(10/3)
```

Fonctions intégrées à Python:

```
>>> abs(5) | >>> abs(-5) | >>> abs(7-8)
```

```
>>> int(3.5674) | >>> int(-54.98765) | >>> int(23/5)
```

```
>>> round(7.99) | >>> round(7.02) | >>> round(9.5) | >>> type(round(9.5))
```

```
>>> round(9.50001) | >>> round(23/7) | >>> round(23/7,5) | >>> round(23/7,2)
```

Accès à des fonctions mathématiques supplémentaires:

```
>>> sqrt(16) | >>> from math import * | >>> sqrt(16) | >>> sqrt(100) | >>> sqrt(-16)
```

```
>>> pi | >>> cos(pi) | >>> cos(60) | >>> pi | >>> cos(pi) | >>> cos(60)
```

```
>>> cos(pi/3) | >>> sin(pi/3) | >>> degrees(pi/2)
```

```
>>> from random import * | 5 fois : >>> random() | 10 fois : >>> randint(5,10)
```

✓ AIDE MEMOIRE

Calculs simples, priorités :	
Python respecte-t-il les priorités?	OUI NON
Pour :	Je tape dans la console :
calculer $5-8+2$ qui vaut	>>>
calculer $5 \times 2,5$ qui vaut	>>>
calculer la puissance 2^3 qui vaut	>>>
calculer l'expression $5,2 \times 10^2$ qui vaut	>>>
Quotients:	
calculer la valeur exacte du quotient $186 : 10$ qui est	>>>
calculer la partie entière du quotient $186 : 10$ qui est	>>>
calculer le quotient de la division euclidienne de 186 par 10 qui est	>>>
calculer le reste de la division de 186 par 10 qui est	>>>
Type de nombres (int ou float):	
déterminer le type du nombre -25 qui est	>>>
déterminer le type du nombre 25.0 qui est	>>>
déterminer le type du quotient $25/5$ qui est	>>>
Fonctions intégrées à Python:	
supprimer la partie décimale de 26,5712 et obtenir	>>>
calculer l'arrondi à l'entier de 26,5712 qui est	>>>
calculer l'arrondi à 10^{-2} de 26,5712 qui est	>>>
Accès à des fonctions mathématiques supplémentaires:	
calculer la racine carrée de 81 qui est	>>>
calculer l'arrondi à 10^{-15} de π qui est 3,141592653589793	>>>
calculer l'arrondi à 10^{-2} de π qui est	>>>
calculer $\cos \pi$ qui vaut	>>>
obtenir une valeur aléatoire entière entre 1 et 6 compris	>>>
obtenir une valeur aléatoire réelle entre dans l'intervalle $[0;1[$	>>>

2. VARIABLES, ENTREES, SORTIES

✓ ACTIVITE:

Entrez dans la console de Python la série d'instructions suivantes en validant par ENTREE.
Observez les résultats obtenus puis complétez le tableau « aide mémoire ».

Affectation d'une variable: On affecte une valeur à une variable en utilisant les signe « = ». Cette variable est stockée dans la RAM du processeur tant que le programme s'exécute. Elle peut être modifiée à tout moment.

```
>>> a | >>> a=5 | >>> a | >>> a+1 | >>> b=a | >>> b | >>> b*3 | >>> c,d=18,2.35
>>> type(c) | >>> type(d) | >>> c+d | >>> type(c+d) | >>> a=2*a | >>> a | >>> a=a+1
>>> a | >>> a+=1 | >>> a | >>> a+=1 | >>>
```

Contrairement à d'autres langages tels que Java ou C, Python effectue un **typage dynamique** des variables. Lors de l'affectation, le type (int ou float) lui est automatiquement attribué. Nous allons constater que c'est le cas aussi pour de nouvelles variables : les chaînes de caractère (string). Leur valeur se déclare toujours entre guillemets ou apostrophes.

Les chaînes:

```
>>> lundi | >>> "lundi" | >>> type("lundi") | >>> mois= "septembre" | >>> type(mois)
>>> "Nous sommes en "+mois (concaténation) >>> a="Hello " | >>> b="World!"
>>> c=a+b (concaténation) >>> c | >>> trois="3" | >>> trois+1 | >>> trois+"1"
>>> trois_converti=int(trois) (transtypage) >>> trois_converti
```

Affichage de l'index d'un caractère dans une chaîne et de sa longueur: C'est la position du caractère dans une chaîne en commençant toujours par 0 pour le premier caractère. La longueur du mot s'obtient avec la fonction de Python: **len(chaine)**

```
>>> jour="mardi" | >>> jour[0] | >>> jour[2] | >>> jour[4]
>>> len("anticonstitutionnellement") | >>> specialite="NSI" | >>> longueur=len(specialite)
>>> longueur | >>> type(longueur).
```

La commande print(): Nous avons jusqu'ici simplement entré le nom de la variable pour afficher sa valeur. Ceci n'est possible qu'en mode console. Lorsque nous passerons en mode programmation l'affichage de la sortie doit être appelé par print(variable) ou print(variable1,variable2, variable3...)

```
>>> print(2+3) | >>> type("coucou") | >>> print("2+3") | >>> a,b=2,3 | >>> print(a+b)
>>> print(a, "plus", b)
```

Les entrées: Si l'utilisateur a besoin d'interagir avec un programme par une saisie au clavier, on utilise la fonction `input()`. Elle affiche une éventuelle invite à l'écran en attendant qu'il entre une donnée au clavier et la valide par `ENTREE`. La valeur retournée par cette fonction est toujours une chaîne (type str) mais on peut en changer le type (transtypage). Il est souvent utile d'affecter la valeur de cette fonction à une variable.

```
>>> prenom= input("Comment tu t'appelles? ") (saisir un nom) >>> type(prenom)
```

```
>>> print(prenom) || >>> print("Salut", prenom, ", ça va?")
```

```
>>> nombre= float(input("Entrez un nombre pour connaître son carré: ")) (saisir un nombre)
```

```
>>> carre=nombre**2 || >>> print("Le carré de", nombre, "est", carre)
```

✓ AIDE MEMOIRE

Affectation d'une variable:	
Pour:	Je tape dans la console :
faire une affectation directe de la valeur 20.5 à une variable <code>temperature_celsius</code>	>>>
faire une affectation multiple des valeurs 5 et 10 aux variables <code>cinq</code> et <code>dix</code>	>>>
faire une affectation de <code>temperature_Kelvin</code> en référence à <code>temperature_celsius</code> (0 degré Kelvin correspond à 273,15°C)	>>>
faire une affectation par auto référence (on veut que la nouvelle valeur de <code>b</code> soit son double augmenté d'une unité)	>>> b=2 >>>
faire une incrémentation (on veut que la valeur <code>b</code> augmente d'une unité)	>>> ou: >>>
Les chaînes, index et longueur	
faire une affectation directe de la valeur <i>Charles Lepierre</i> à une variable <code>lycee</code>	>>>
connaître la quatrième lettre de la valeur de cette variable	>>>
connaître le nombre de caractères de cette chaîne	>>>

Les commandes `print()` et `input()`

afficher "Lycée Charles Lepierre" dans la console en utilisant la variable précédente	>>>
demander une entrée de texte ("Entrez votre texte: ") et l'attribuer à une variable <i>txt</i> puis afficher sa valeur	>>> >>>
demander une entrée d'un nombre entier ("Entrez votre nombre: ") et l'attribuer à une variable <i>nombre</i> puis afficher sa valeur	>>> >>>