

Python – résumé

Les variables : Une variable est un espace mémoire dans lequel il est possible de stocker une valeur (une donnée). Le nom des variables en Python peut être constitué de lettres minuscules (a à z), de lettres majuscules (A à Z), de nombres (0 à 9) ou du caractère souligné (_). Néanmoins, un nom de variable ne doit pas débiter ni par un chiffre, ni par _ et ne peut pas contenir de caractère accentué.

Python est sensible à la casse, ce qui signifie que les variables Test, test ou TEST sont différentes. Enfin, n'utilisez jamais d'espace dans un nom de variable puisque celui-ci est le séparateur d'instructions.

Les types de donnée : Sous Python, il n'est pas nécessaire de définir le type des variables avant de pouvoir les utiliser. Il suffit d'assigner une valeur à un nom de variable pour que celle-ci soit automatiquement créée avec le type qui correspond au mieux à la valeur fournie. Python dispose de différents types de variables : entier, réel, chaîne de caractères, booléen, liste, etc.

```
a = 20 // entier ou int
pi = 3.14159 // réel ou float
msg = "Quoi de neuf ?" // chaîne de caractères ou str
cond=a>10 // booléen ou bool
tab = [9, 7, 6, 9] // liste ou list
```

Les opérateurs mathématiques

Opérations	Symboles	Exemples
addition	+	2 + 5 donne 7
soustraction	-	8 - 2 donne 6
multiplication	*	6 * 7 donne 42
puissance	**	5 ** 3 donne 125
division	/	7 / 2 donne 3.5 (en Python 3)
modulo	%	7 % 3 donne 1 (reste de la division entière)
quotient	//	7 // 3 donne 2 (quotient de la division entière)

Les opérateurs booléens

Opérations	Symboles
ET booléen	and
OU booléen	or
NON booléen	not

Affichage en console

Pour afficher du texte à l'écran on utilise la fonction print().

Attention, les parenthèses sont obligatoires en Python

Il est possible de mélanger du texte avec du texte mais aussi avec le contenu de variables.

```
>>> print("je suis un texte")
je suis un texte
>>> print('je suis un texte')
je suis un texte
>>> print("j'ai bien compris")
j'ai bien compris
>>> print("Mon","prénom")
Mon prénom
>>> print("Mon","prénom",sep=" ")
Mon prénom
>>> a=20
>>> print("la valeur de a est",a)
la valeur de a est 20
```

Annotations :

- On peut utiliser le guillemet ou la simple cote
- L'espace est ajouté automatiquement
- permet de modifier le séparateur

Saisie clavier

Pour que Python puisse lire ce que vous tapez au clavier, vous allez utiliser la fonction input(). Attention, avec Python input() renvoie systématiquement une chaîne de caractères. Il faut utiliser int(input()) ou float(input()) pour saisir un entier ou un réel

```
>>> message=input("Entrer un nom : ") // lecture d'une chaîne de caractères
>>> n=1
>>> a=int(input("Entrer"+str(n)+"entier: ")) // lecture de n entier ici n=1
str() permet de transformer le int en string et int() l'inverse
>>> b=float(input("Entrer un réel: ")) // lecture d'un réel
```

Affectation

En langage naturel, on écrit :

- a prend la valeur 300
- on ajoute 3 au contenu précédent de la variable a

En pseudo-code, on écrira ceci :

- a ← 300
- a ← a + 3

En Python, on écrit :

- a = 300
- a=a+3

Remarques :

- Quand on met une valeur dans une variable, toute valeur existant déjà dans la variable est effacée (écrasée).
- « a = b » ne veut pas dire la même chose que « b = a ».

Fonctions : Pour éviter de réécrire un code ou de mieux exploiter ses résultats, il suffit de l'inclure dans une fonction qui est défini avec ses paramètres d'entrée. L'appel de la fonction se fait simplement ensuite par son nom suivi des arguments (qui sont les valeurs de chaque paramètre).

Exemple	Ecriture de la fonction	Appel de la fonction	Utilisation
<pre>long=float(input("longueur ?")) larg=float(input("largeur ?")) peri=2*(long+larg) print("le perimetre est de ",peri)</pre>	<pre>def perimetre(long,larg): p=2*(long+larg) return p</pre>	<pre>print(perimetre(7,4.25))</pre>	<pre>P1=perimetre(6.24,3.12) P2=perimetre(7.14,2.50) print(P1>P2)</pre> <p>Le return permet de renvoyer une valeur qui peut être stocké dans une variable</p>