

Découverte de Python

Environnement de travail sur EduPython

La console (Shell) : "en bas"

Il s'agit d'une interface interactive que l'on peut utiliser comme une calculatrice. L'invite est le symbole `>>>` qui indique que Python attend une instruction.

Dans la console, on ne peut taper qu'une ligne d'instructions à la fois : en appuyant sur la touche "Entrée" on provoque l'exécution des instructions indiquées.

Opérations arithmétiques de base

addition	+		
soustraction	-		
multiplication	*	reste de la div. euclidienne	%
division	/	quotient de la div. euclidienne	//
puissance	**		

Exercice 1

Taper les commandes suivantes dans la console et noter les résultats obtenus.

```
>>> -25+3      >>> 5*2      >>> 2*5      >>> 5**2
>>> 2**5      >>> 10/2     >>> 10//2     >>> 5/2
>>> 5//2      >>> 10%3     >>> 3%10      >>> (5+3)/(4*2)
```

Commande print

- Pour afficher un texte : `print('mon texte')` (avec des ' ' !)
- Pour afficher une valeur : `print(valeur)` (sans ' ' !)
- On peut combiner les deux : `print('mon texte',valeur)`

Exercice 2

Tester les commandes suivantes dans la console.

```
>>> print('Hello world !')
>>> print(3)      >>> print(5+2)      >>> print(1/3)
>>> print('Nombre trois : ',3)      >>> print('5+5 = ',5+5)
>>> print('Un nombre : ',12,' Un autre : ',-3)
```

L'éditeur de texte : "en haut"

L'éditeur de texte est un espace dans lequel on peut écrire des scripts contenant plusieurs lignes : une instruction par ligne.

Ces scripts pourront être enregistrés, puis exécutés dans la console.

Pour exécuter un script : cliquer sur l'icône représentant une flèche verte.

Il est possible (même souhaitable !) d'enregistrer puis d'ouvrir des fichiers contenant des scripts. Veillez à bien enregistrer dans vos dossiers personnels !

Exercice 3

Ouvrir une nouvelle fenêtre d'éditeur et y taper le script suivant :

```
10
2**2+3**2
print(-5)
print(4/5)
print('bonjour')
```

Exécuter ce script (il vous sera peut-être demandé d'enregistrer le fichier) et regarder le résultat affiché dans la console. Que constate-t-on ?

Attention !

Par défaut, les résultats des calculs effectués dans un script exécuté depuis l'éditeur de texte ne s'affichent pas ! Pour afficher une valeur depuis l'éditeur, il faut donc nécessairement utiliser la commande `print`.

Dans un script, il est possible de placer des "commentaires" en les faisant précéder du symbole `#`. Ces commentaires, destiné à rendre le script plus compréhensible pour un éventuel lecteur, seront ignorés lors de l'exécution.

Exercice 4

Effacer le script précédent, taper celui-ci, puis exécuter :

```
for k in range(5): # Cette syntaxe sera étudiée plus tard
    print('Indice=',k)
```

Bilan :

- On utilisera l'éditeur de texte pour rédiger des programmes nécessitant plusieurs lignes d'instructions (multiples calculs, tests logiques, boucles, définition de fonction...).
- On utilisera la console uniquement pour de "petites tâches" : tester une commande, afficher une valeur, évaluer une fonction...

Variables

Une variable est un emplacement mémoire dans lequel on peut stocker une donnée (un nombre, une liste de nombres, un texte, etc...). On peut nommer une variable par une lettre ou par une chaîne de caractères. Notons à ce sujet que python est sensible à la casse (majuscules/minuscules).

Affectation

En programmation, une *affectation* est le fait d'attribuer une valeur à une variable. En Python, le symbole d'affectation est tout simplement le signe `=`.

Affectation

`ma_variable = valeur` Exemples : `x = 10`, `S = -3`, `nombre = 22`.

⚠ Attention !

Le symbole `" = "` en Python n'a pas le même sens qu'une égalité mathématique !

On peut écrire `x = 3` (je range la valeur 3 dans la variable `x`)
mais pas `3 = x` (cela n'a pas de sens car 3 n'est pas une variable...)

✎ Exercice 5

Tester les commandes suivantes (dans l'ordre de lecture) dans la console.

```
>>> a=2    >>> print(a)    >>> a = 4    >>> print(a)
>>> nombre=12    >>> print('La valeur du nombre est',nombre)
>>> 4=a    >>> 2=2
```

Calculs avec des variables

Une fois des variables définies, il est bien-sûr possible de les utiliser dans des calculs !

✎ Exercice 6

Tester les commandes suivantes (dans l'ordre de lecture) dans la console.

```
>>> a=3    >>> b=5
>>> print('b=',b)    >>> b=b+1    >>> print('Maintenant b=',b)
>>> c = a+b    >>> print(c)
```

L'instruction `b=b+1` s'appelle une *incrément*.

Profitons-en pour mentionner la syntaxe alternative suivante, souvent utilisée :

📖 Incrément

Au lieu de `x = x + 1`, on peut utiliser la syntaxe : `x += 1` (on augmente `x` de 1).
On peut aussi écrire `x += 8` (on augmente `x` de 8), `x -= 2` (on diminue `x` de 2)...

✎ Exercice 7

1) Deviner ce qui est affiché par le script suivant. On pourra tester en l'exécutant dans l'éditeur de texte.

```
A=1
B=3
A=A+B
B=A*B
print(B)
```

2) Sans écrire aucun chiffre, compléter le script suivant pour que l'affichage final soit : `'a=7 et b=2'`.

```
a=2
b=7

print('a=',a,'et b=',b)
```

Différents types de variables

En Python, les variables peuvent contenir des objets de natures différentes : on dit qu'elles sont de différents types. Les principaux types sont les suivants :

- **Les entiers relatifs** : `int` (comme "integer" en anglais).
- **Les décimaux** : `float` (comme "floating number" en anglais).
- **Les chaînes de caractères** : `str` (comme "string" en anglais).
- **Les booléens** : `bool` (variable à deux états possibles : `True` ou `False`)
- **Les listes** : `list`

✎ Exercice 8

Exécuter le script suivant. Que signifient ces affichages ?

```
a=5
b=-4.2
c=10/2
d='bonjour'
print(type(a))
print(type(b))
print(type(c))
print(type(d))
```