

VALEURS NUMÉRIQUES : COMPARAISONS ET CONVERSIONS

I- Convertir une variable

À utiliser	Pour transformer le type...	Exemples
<code>int(x)</code>	Le type float en type int	<code>int(2.5)</code> retourne 2 <code>int(-2.5)</code> retourne -2
<code>float(x)</code>	Le type int en type float	<code>float(2)</code> retourne 2.0
<code>str(x)</code>	Les types int, float ou list en type str	<code>str(3)</code> retourne '3' <code>str(3.0)</code> retourne '3.0' <code>str([1,2,3])</code> retourne '[1, 2, 3]'
<code>list(x)</code>	Le type str en type list	<code>list("test")</code> retourne ['t', 'e', 's', 't']
<code>[x]</code>	Les type float ou int en type list	[3] ou [3.0]

II- Quelques opérations sur les variables

Les opérations élémentaires que l'on peut effectuer sur les variables sont les suivantes :

Opérateur	Utilité
<code>+=</code>	Rajouter à la variable
<code>-=</code>	Enlever à la variable
<code>*=</code>	Multiplier la variable.
<code>/=</code>	Diviser la variable

Opérateur	Utilité
<code>==</code>	Test d'égalité
<code>!=</code>	Test de non égal
<code><</code>	Test d'infériorité
<code>></code>	Test de supériorité
<code><=</code>	Test : inférieur ou égal
<code>>=</code>	Test : supérieur ou égal

Voici quelques exemples :

Syntaxe rentrée dans la console	Résultat obtenu
<code>x = 4</code> puis <code>x += 3</code>	7
<code>X,Y = "J'aime" , " la prépa"</code> puis <code>X + Y</code>	"J'aime la prépa"
<code>3*[1,2,3]</code>	[1,2,3,1,2,3,1,2,3]
<code>[4,5,6] + [1,2,3]</code>	[4,5,6,1,2,3]

5 < 4	False
[1,2] == [2,1]	False
"Bonjour" == "bonjour"	False
False or True	True
False and True	False
not False	True

Exercice 1

Taper dans la console les instructions suivantes:

```
w = 10 ↵      w += 1 ↵      w *= 2 ↵
```

Que renverrait la commande w ↵ ? Expliquer

.....

Exercice 2

1) Taper dans la console les instructions suivantes:

```
x = "13" ↵      y = "25" ↵
```

Que renvoient les commandes suivantes? Le vérifier.

```
x + y ↵ .....
```

```
x * y ↵ .....
```

```
3 * x ↵ .....
```

2) Quels sont les types des quatre variables suivantes? Le vérifier.

```
13      13.0      "13"      0 == 1
```

.....

III- Programme Python

- Pour écrire un nouveau programme, il suffit de cliquer sur *fichier* puis *nouveau fichier*.
- Le premier réflexe doit être de **sauvegarder** votre programme avec un nom cohérent.
- Une fois votre programme écrit, n'oubliez pas de le **sauvegarder** avant de l'exécuter.
- Pour l'exécuter, il suffit d'utiliser le raccourci **CTRL + S** (sur Windows) ou bien l'onglet adapté.
- Un programme s'écrit toujours dans l'ordre suivant :
 - 1) **en-tête** : commentaire commençant par #, indiquant la date de création du programme et le nom d'auteur,
 - 2) importation des **bibliothèques** (modules),
 - 3) déclaration des **fonctions** que vous importez.

Exercice 3

Ouvrir un nouveau programme dans l'éditeur. Sauver immédiatement ce programme dans le même dossier que l'exercice précédent en le nommant `premieres_commandes`. Dans ce fichier taper les commandes:

```
2 ** 5
print (3 * 5)
x=5 ** 2
```

Exécuter le programme. Quels calculs ont été effectués? affichés? stockés?

.....
.....

Exercice 4

Créer un nouveau programme qui contient les lignes suivantes :

```
x=10
y=15
z=x+y
x=y
y=z
print(x+y+z)
```

Deviner ce qui va s'afficher à l'exécution. Le vérifier.

.....

IV- Utilisation des modules

On va détailler ici certaines modules (bibliothèques) de Python.

Afin d'utiliser une bibliothèque ou module dans Python il faut au préalable **l'importer dans l'environnement Spyder**. Il y a plusieurs façons de faire. Par exemple avec la bibliothèque math :

- Importation d'une fonction particulière : `from math import sin` par exemple, permet d'importer spécifiquement la fonction sin.
- Importation de la bibliothèque : `import math` importe la bibliothèque, les fonctions sont alors accessibles par `math.sin`, `math.cos`, par exemple.
- Importation de la bibliothèque avec un **alias** : on abrège le nom de la bibliothèque avec un alias. Avec `import math as m` les fonctions sont alors accessibles par `m.sin`, `m.cos`... C'est la méthode qu'on utilisera la plupart du temps pour importer un module.

Vous utiliserez plusieurs modules au cours de l'année : math, numpy, random, matplotlib...

Exercice 5 (Utiliser une bibliothèque)

- 1) Créer un nouveau programme nommé `bibliotheque_maths` et l'enregistrer toujours dans le même dossier.
- 2) Inscire l'instruction adéquate pour importer le module math sous l'alias m.
On dispose maintenant, entre autre, des fonctions sin, cos, exp, log (et pas ln...), sqrt, des nombres e , π ...
Pour faire appel à l'une de ces commandes il faut la faire précéder de m., par exemple : `m.sqrt(2)` pour $\sqrt{2}$.
- 3) Définir dans le programme $x = \pi$ et $y = \frac{355}{113}$ (fraction censée être une approximation correcte de π).
Écrire des instructions qui permettent l'affichage des valeurs de x et y .
- 4) Écrire une instruction permettant d'afficher l'écart entre x et y (la valeur absolue sous python s'obtient avec la commande `abs()`).
Combien obtient-on? Commenter.

V- Corrections

Exercice 1

```
w = 10 ↵      w += 1 ↵      w *= 2 ↵
```

La commande `w ↵` renverrait $w = 22$ car on affecte la valeur 10 à w puis on lui ajoute 1 enfin, on multiplie w par 2.

Exercice 2

1) `x = "13" ↵ y = "25" ↵`

`x + y ↵` renvoie '1325'

`x * y ↵` renvoie une erreur

`3 * x ↵` renvoie '131313'

2) 13 est de type entier, 13.0 est de type flottant, "13" est de type chaîne de caractères et `0 == 1` est de type booléen.

Exercice 3

```
2*5 # Effectué mais pas stocké ni affiché.
```

```
print(3*5) # Effectué et affiché mais pas stocké.
```

```
x = 5*2 # Effectué et stocké mais pas affiché.
```

Exercice 4

```
x = 10
```

```
y = 15
```

```
z = x+y
```

```
x = y
```

```
y = z
```

```
print (x+y+z)
```

$x = 10, y = 15, z = 25$ puis $x = 15$ et $y = 25$ donc $x + y + z = 65$.

Exercice 5

1)

2) On écrit : `import maths as m`

3) On écrit :

```
x = m.pi
```

```
y = 355 / 113
```

```
print(x)
```

```
print(y)
```

4) On écrit: `print(abs(x-y))` et on obtient que l'écart est de $2,668 \times 10^{-7}$