

I- Mode Interactif.

1) Ouvrir l'interpréteur Python en ligne de commande

- Se connecter avec un login et mot de passe
- Lancer le terminal de **windows** par la commande **cmd**
- Lancer l'interpréteur Python en tapant **python** puis **ENTER** en ligne de commande.

2) Arithmétique (quelques instructions).

Effectuer, avec l'interpréteur Python, les opérations:

- addition ($3 + 2 = 5$)
- soustraction ($2 - 6 = -4$)
- multiplication ($5 * 3 = 15$)
- la division classique ($7 / 2 = 3.5$)
- le quotient d'une division euclidienne : ($7 // 2 = 3$)
- le reste d'une division euclidienne ($7 \% 2 = 1$)
- Attention aux divisions euclidiennes dont le diviseur est négatif ($-7 / 2 = 4$ et $-7 \% 2 = 1$)
- puissance ($2 ** 3 = 8$)

Quelques erreurs avec les instructions:

$1 + * 3$: **SyntaxError: invalid syntax**

$1/(5-5)$: **ZeroDivisionError: division by zero**

$1,3 + 2,4$: (1, 5, 4) : Le virgule est un séparateur, il faut utiliser le point

$1.3 + 2.4$: 3.7

3) Variables

$a = 1 + 2$

$a + a ** 2 = 12$

$\text{cube} = a * a * a$

$c = 1$

$c = c + 2 = 3$

Quelques erreurs avec les variables :

$2a = 10$: **SyntaxError: invalid syntax** Le nom d'une variable ne peut pas commencer par un chiffre.

$\text{def} = 3$: **SyntaxError: invalid syntax** **def** est un mot réservé pour Python

$b + 1$: **NameError: name 'b' is not defined**

$1 = 2$: **SyntaxError: cannot assign to literal**

II-Mode Programme

Le mode programme de Python consiste à écrire une suite d'instructions dans un fichier et les faire exécuter par l'interpréteur Python. Cette suite d'instructions s'appelle un programme ou code source.

1) Affichage

En mode programme, les résultats des expressions calculées ne sont plus affichés sur l'écran. En Python il faut utiliser la « fonction » **print**

Exemples : **print(3)** , **print(2 + 1)** , **print(3 ** 2)** , etc.

2) Affichage de texte

On peut donner à **print** un message à afficher, écrit entre guillemets.

Exemple : **print("Bonjour tous le monde !")**

Le texte écrit entre guillemets est appelé une chaîne de caractère. Une chaîne de caractère n'est pas interprétée par Python. Exemple : **print("2 + 1")** ne calcule rien et affiche **2 + 1**

Quelques erreurs avec la fonction (l'instruction) `print` :

`print 3` *SyntaxError: Missing parentheses in call to 'print'. Did you mean print(...)?*

`print("3)` *SyntaxError: unterminated string literal (detected at line 1)*

`print(1 + "3)` *TypeError: unsupported operand type(s) for +: 'int' and 'str'*

3) Séquence d'instructions

Les lignes commençant par le caractère `#` sont des commentaires.

Elles ne seront pas interprétées par Python.

Affectation de variable, opération avec les variables et affichage avec la fonction `print`

Affichage successif avec retour à la ligne.

```
a= 7
b= 12 -a
print(a)
print(b)
print(a * b)
```

Si on souhaite afficher plusieurs éléments sur la même ligne, on peut remplacer plusieurs `print` par une seule en séparant les éléments par une virgule.

```
a= 7
b= 12 -a
print("Le produit de ", a, " et de ", b, " vaut ", a * b, " .")
```

4) Interagir avec l'utilisateur

Pour permettre l'interaction du programme avec l'utilisateur, on utilise l'instruction `input` pour récupérer les caractères tapés au clavier par l'utilisateur. L'instruction `input` renvoie donc une chaîne de caractères qu'on place dans une variable. Pour transformer cette chaîne de caractères en entier, il faut utiliser l'instruction `int`

Exemple :

```
s = input("Entrer votre age : ")
print("Vous avez saisi ", s, " ", type(s))
a= int(s)
print("L'année dernière, vous aurez ", a+1, " ans")
```

L'instruction `type(s)` renvoie `<class 'str'>` pour dire que la variable `s` contient une chaîne de caractères (l'instruction `input` renvoie toujours une chaîne de caractères.).

5) L'instruction *print* et retour à la ligne.

Par défaut, l'instruction *print* provoque un retour à la ligne.

Exemple :

```
s = input("Entrer votre age : ")
a= int(s)
print("Vous avez ", a, " ans")
print("et l'année prochaine vous avez ", a+1, " ans")
```

On peut changer ce comportement en fournissant une autre chaîne de caractère à accoler à l'affichage principale. Cette chaîne de caractère peut être vide.

Exemple :

```
s = input("Entrer votre age : ")
a= int(s)
print("Vous avez ", a, " ans", end=" ")
print("et l'année prochaine vous avez ", a+1, " ans")
```

Quelques erreurs avec l'instruction *int* :

```
ValueError: invalid literal for int() with base 10: '2é'
```

6) Un programme complet.

```
# Calcul de l'age
saisi = input("Entrer votre année de naissance : ")
annee = int(saisi)
age = 2048 - annee # on calcul l'age par soustraction
print("Vous aurez ", age, " ans en 2048.")
```

III-Mode Interactif et Mode Programme

Bien que le mode programme est le mode d'utilisation standard, le mode interactif reste très utile pour mettre au point d'un programme ou encore tester un élément du langage Python où on a un doute. Il est donc indispensable de savoir utiliser les deux modes.