

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'interprète Python en tapant **python** puis **ENTER** en ligne de commande.

2) Arithmétique (quelques instructions).

Effectuer, avec l'interprète 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 division euclidienne 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'interprète Python. Cette suite d'instructions s'appelle un programme ou code source.

1) Affichage

En mode programme, les résultats des expressions calculé 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 interprété par Python. Exemple : **print("2 + 1")** ne calcul 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

```
a = 17
b = 38 - a
print(a)
print(a)
```

affiche deux entiers sur l'écran (sur deux lignes successives)

```
17
21
```

si on souhaite afficher les deux entiers sur la même ligne, on peut remplacer les deux instructions **print** par une seule, en séparant les deux éléments à afficher par une virgule.

```
print(a , b)
17    21
```

Plus généralement, on peut utiliser **print** avec un nombre arbitraire d'éléments qui peuvent être des nombres ou des chaînes de caractères.

```
a = 17
b = 38 - a
print("La somme de ", a, " et de ", b, " vaut ", a + b)
```

L'exécution de ce programme affiche le message suivant :

La somme de 17 et de 21 vaut 38

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()
a = int(s)
print("Le nombre suivant est ", a + 1)
```

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

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

On peut changer ce comportement en fournissant une autre chaîne de caractères à accoler à l'affichage principale. Exemple :

```
s = input()
a = int(s)
print("Le nombre précédent est ", a - 1)
print("Le le nombre suivant est ", a + 1)

print("Le nombre précédent est ", a - 1, end= " ")
print("Le le nombre suivant est ", a + 1)
```

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

```
a = int("2è")    ValueError: invalid literal for int() with base 10: '2è'
```

III- Bibliothèque Turtle