

Nous utiliserons Python avec l'environnement de développement **IDLE**

IDLE est un environnement de développement intégré pour Python , il propose un certain nombre d'outils :

- Un éditeur de textes (pour écrire le programme ou script)
- Un interpréteur (pour exécuter le programme)
- Un débogueur (pour tester le programme)

1. La gestion des espaces

Dans le langage Python , on peut passer des lignes pour plus de clarté , ce n'est pas pris en compte lors de l'exécution du programme. Par contre , on ne peut pas ajouter d'espace en début de ligne : on appelle **indentation** un décalage en début de ligne , elle permet de délimiter un bloc d'instructions .

Exemple :

```
>>> a=5
>>> if a>0:                                # teste si un nombre est positif
    print("a est positif")
a est positif
'''
```

Le « alors » n'apparaît pas , c'est l'indentation qui délimite le bloc d'instructions

Commentaires précédés de #

2. Majuscule ou minuscule

Les instructions Python s'écrivent en minuscule . On peut utiliser les majuscules dans les noms des variables , mais attention à bien respecter l'écriture que vous avez choisie (on dit « respecter la casse »)

3. Le rôle du symbole =

- Le symbole = n'est pas celui de l'égalité mais celui de **l'affectation d'une valeur à une variable** . On stocke une valeur numérique ou du texte dans une mémoire .
La syntaxe est **NomVariable = valeur**
- **Pour afficher la valeur d'une variable : print(NomVariable)**
- Le symbole == teste si une égalité est vraie ou fausse

Exemple : Tester les lignes suivantes avec Python

>>a=21 >>print(a) >>affichage	>>a==20 >>affichage ??	>>a==21 >>affichage ??	>>b=5 >>b=b+1 >> print(b) >> affichage ??
-------------------------------------	---------------------------	---------------------------	--

Interpréter l'affichage obtenu

Remarque : dans l'interpréteur il est inutile de taper « print »

Affectations multiples :

```
>>> a,b=10,22
>>> a,b= a+b,a-b
>>>print(a,b)
>>>affichage
```

expliquer l'affichage obtenu

Exercices :

1. Ouvrez l'interpréteur. Pour chaque question, exécutez les commandes et commentez-les :

>>> 3+2	>>> s=12 >>> m=24 >>> s=s+m >>> 3*s	>>> s=5 >>> s**2	>>> n=13 >>> n/2	>>>n//2	>>> n%2
---------	--	---------------------	---------------------	---------	---------

2. Que fait le programme suivant ?

```
from lycee import *
x=demande('Entrez une valeur pour x')
y=demande('Entrez une valeur pour y')
x,y=x+y,x-y
x,y=x+y,x-y
print "maintenant, x=",x,"et y=",y
```

4. Chaînes de caractère

Quand on veut définir une **chaîne de caractère**, on la délimite par des **guillemets " ou par une apostrophe '**
 Pour **afficher** une chaîne de caractère, il faut utiliser la commande **print("Chainedecaractère")**

Exemple : Exécuter les commandes suivantes et observer l’affichage obtenu

>>> n=5 >>> print(" le carré de ",n,"est",n**2)	>>> abc= "deux mots" ' >>> print(abc)	>>> print("abc")
--	--	------------------

5. Demande d’un nombre .

La commande **input()** permet d’entrer une valeur mais renvoie une chaîne de caractères . Donc si c’est un nombre que l’on veut manipuler, il faudra utiliser la fonction **int()** s’il s’agit d’un entier ou **float()** s’il s’agit d’un décimal

Exemple :

```
print("entrer un nombre positif")
chaine=input()
nb=float(chaine)
if nb > 0 :
    print("ce nombre convient")
```

La variable chaine prend la valeur entrée par l'utilisateur (sous forme de chaîne de caractère)

La variable nombre transforme la chaîne de caractère en nombre réel

6. Test et condition .

Si alors ...	Si ... alors ...sinon...	Si ... alors ... sinon si ... sinon ...
if test : <i>Bloc d’instructions si la condition est réalisée</i> <i>Remarque :</i> Le "alors" n’apparaît pas en Python, c’est l’indentation qui délimite le bloc à exécuter.	if test : <i>Bloc d’instructions si la condition est réalisée</i> else : <i>Bloc d’instructions si la condition n’est pas réalisée</i>	if test1 : <i>Bloc d’instructions si la condition au test 1 est réalisée</i> elif test 2 : <i>Bloc d’instructions si la condition au test 2 est réalisée</i> else : <i>Bloc d’instructions si aucune condition n’est réalisée</i>

Exemples :

<pre>>>> # Premier exemple de condition >>> a = 5 >>> if a > 0: # Si a est supérieur à 0 ... print("a est supérieur à 0.") ... </pre>	<pre>>>> age = 21 >>> if age >= 18: # Si age est supérieur ou égal à 18 ... print("Vous êtes majeur.") ... else: # Sinon (age inférieur à 18) ... print("Vous êtes mineur.")</pre>
<pre>>>> if a > 0: # Positif ... print("a est positif.") ... elif a < 0: # Négatif ... print("a est négatif.") ... else: # Nul ... print("a est nul.")</pre>	

A voir très bientôt : Les boucles en Python ! Mais pour l’instant...

Exercices : Ecrire en Python

1. Un programme qui demande un nombre et qui affiche son opposé
2. Un programme qui demande à l'utilisateur son année de naissance et qui affiche son âge
3. Un programme qui permet de saisir le nom de l'utilisateur et de renvoyer « Bonjour » suivi du nom
4. Un programme qui teste si un nombre donné est pair ou impair.