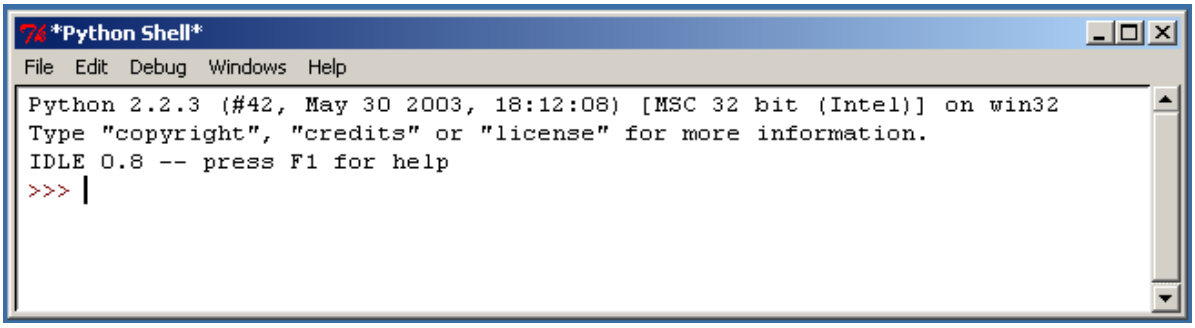
# INITIATION AU LANGAGE PYTHON
## séance n°1

## L'environnement de travail

Python présente la particularité de pouvoir être utilisé de plusieurs manières différentes. Vous allez d'abord l'utiliser ***en mode interactif***, c'est-à-dire d'une manière telle que vous pourrez dialoguer avec lui directement depuis le clavier. Cela vous permettra de découvrir très vite un grand nombre de fonctionnalités du langage.

Dans un second temps, vous apprendrez comment créer vos premiers programmes (scripts) et les sauvegarder sur disque dur ou sur clef USB. Si vous utilisez une interface graphique telle que *Windows*, *Gnome, WindowMaker* ou *KDE*, vous préférerez vraisemblablement travailler dans une « fenêtre de terminal », ou encore dans un environnement de travail spécialisé tel que ***IDLE***. Voici par exemple ce qui apparaît dans une fenêtre de Windows :




Les trois caractères « supérieur à » constituent le signal d'invite, ou *prompt principal*, lequel vous indique que Python est prêt à exécuter une commande.

## Calculer avec Python

Vous pouvez utiliser l'interpréteur comme une simple calculatrice de bureau. Testez par exemple les commandes suivantes dans l'environnement ***IDLE*** :

```
>>> 15+3

>>> 2 - 17                    # les espaces sont facultatifs

>>> 8 + 3 * 4                 # la hiérarchie des opérations mathématiques
                              # est-elle respectée ?
>>> (8+3)*4

>>> 20 / 3
```

Comme vous pouvez le constater, les opérateurs arithmétiques pour l'addition, la soustraction, la multiplication et la division sont respectivement +, -, * et /. Les parenthèses sont fonctionnelles.

## Affectation-réaffectation

Nous avons vu en cours l'opération **d'affectation**, qui consiste à donner à une *variable* une valeur (nombre entier, réel, chaîne de caractères, etc...).

En pseudo-langage, l'opération d'affectation a été symbolisée par le sigle "←"
Exemple : a←5 # J'affecte à la variable a la valeur 5

**En langage Python l'opération d'affectation s'écrit « = »**.

Testez par exemple les commandes suivantes :

```
>>> a=3
>>>a

>>>a+10

>>>a/2
```

Ne croyez pas que cette affectation soit définitive. On peut **réaffecter** à la variable « a » une nouvelle valeur en réutilisant le symbole « = ».

```
>>>a                #Quelle est la valeur actuelle de a ?

>>>a=a+5            #On réaffecte à a sa valeur précédente augmentée de 5
>>>a

                    #Que vaut a maintenant ?
```

**Remarque** : le signe « = » utilisé ici n'a absolument aucun rapport avec l'égalité au sens mathématique du terme.

Python a une convention :si ce que l'on écrit est vrai, il renverra **true** en réponse ; si c'est faux, il renverra **false**.

Par exemple, testez les commandes suivantes (en vous souvenant bien de la dernière valeur affectée à la variable a).

```
>>> a<1

>>>a>2

>>>a==8             #le sigle « == » est l'égalité classique, au sens mathématique usuel.

>>>a!=4             #le sigle « != » signifie « différent ».
```

Que va renvoyer l'instruction a>=4 ?

**Synthèse partielle** : Quelle différence fondamentale faîtes-vous entre l'écriture a=10 et a==10 ?

## Typage des variables

La variable a que nous avons rencontrée jusqu'ici appartient à une seule catégorie : celle des entiers. En anglais, cela se dit « Integer ».

Testez les commandes suivantes :

```
>>>a

>>>type(a)          #Quel est le résultat affiché ? Signification ?

>>>b=2.3            #Le séparateur décimal est le point, pas la virgule.

>>>type(b)          #Quel est le résultat affiché ? Signification ?

>>>c="Salut"

>>>type(c)

>>>d=[20,35,87,12]

>>>type(d)

>>>e=[5, "coucou",32.5,"Hugh"]          #Mélange de types dans un autre ?

>>>type(e)
```

Ces variables appartiennent à des **types** très différents. Nous verrons dans quelle mesure les utiliser séparément, puis ensemble. Rappeler ces types :

## Opérations sur les variables

Commençons par réaffecter à chacune des valeurs a,b c et d d'autres valeurs.
Testez les commandes suivantes:

```
>>>a=2
>>>b=5.5
>>>c="Hello "          #avec l'espace entre Hello et le guillemet de fin.
>>>d="vous !"

>>>a+b

>>>c+d

>>>print(c+d)

#Quelle différence remarquez-vous entre les deux commandes précédentes ?
>>>a+c                #On ne mélange pas les genres!

>>> a*2

>>>a**2
```

>>>a**3

>>>a**4

>>>a**5

Que signifie l'opération « ** » entre un nombre et un entier ?

>>>c*2

>>>c*3

Même question entre une chaîne de caractères et un entier non nul pour « * ».

**Exercices :**

1. Signalons au passage la disponibilité de l'opérateur ***modulo***, représenté par le symbole **%**. Cet opérateur fournit ***le reste de la division entière*** d'un nombre par un autre. Essayez par exemple :
   >>> 10 % 3 (et notez ce qui se passe !)
   >>> 10 % 5

   Cet opérateur vous sera très utile plus loin, notamment pour tester si un nombre **a** est divisible par un nombre **b**. Il suffira en effet de vérifier que **a % b** donne un résultat égal à zéro.

   Vérifier à l'aide de la commande modulo si le nombre 33294 est divisible par 3 puis par 17.

2. Écrire un programme qui permet d'échanger la valeur de 2 nombres a et b indépendamment des valeurs affectées à a et à b au départ.

# Récapitulatif

**A faire sur feuille, sans ordinateur**

## I) Affectation

**Définition** : La notion **d'affectation** (ou d'assignation) désigne l'opération par laquelle on établit un lien entre le nom de la variable et sa valeur (son contenu). On l'a noté en langage usuel avec le symbole ←

*Question 1* : Comment se note-t-elle en Python ?

*Question 2* : Comment fait-on pour affecter à une variable a la valeur 3 ?

*Question 3* : Cette affectation est-elle définitive ? Comment fait-on pour lui ré-affecter une autre valeur ?

*Question 4* : Quelle est la signification du symbole « == » en Python ?

*Question 5* : On écrit les lignes de commande suivantes. Quel résultat va renvoyer l'ordinateur ?

```
>>>a=5

>>>a

>>>a==3

>>>a==5

>>>a=18

>>>a<20

>>>a>10

>>>a=a+6

>>>a

>>>b=30

>>>b>a

>>>a+b
```

*Question 6* : On écrit les lignes de commande suivantes. Quel résultat va renvoyer l'ordinateur ?

```
>>>a=10
>>>b="salut"
>>>a+b
```

Même question avec :

```
>>>a=10
>>>b= 5.2
>>>a+b
```

## II) Typage des variables

*Question 7* : Rappeler les différents types de variables rencontrés.

*Question 8* :
a) Comment assigne-t-on à une variable a le type string ? (chaîne de caractères)

b) Que signifie l'opération **concaténer** deux chaînes de caractères ?

c) Que va renvoyer les commandes suivantes ?
```
>>>a="Bonjour "
>>>b="vous !"
>>>a+b

>>>print(a+b)
```

*Question 9* : Que va renvoyer la commande suivante ?
```
>>> a="Oui "
>>>a*3
```

*Question 10* : On admet que le premier élément d'une liste LIST se note LIST[0], le second LIST[1], etc...
On se donne une liste par la commande suivante :
```
>>>LIST=[5,48,-57,"ABC",451.9,"soleil",-789]
```
Que va renvoyer les commandes suivantes ?
```
>>>LIST[0]

>>>LIST[2]

>>>LIST[5]
```

Pour un complément sur la notion d'affectation avec Python, se référer à la fiche : **Plus sur l'affectation**.