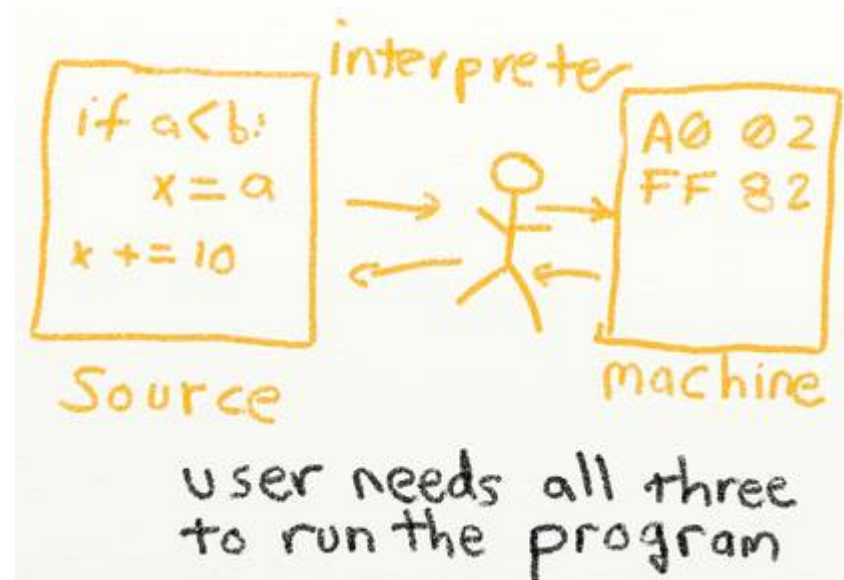
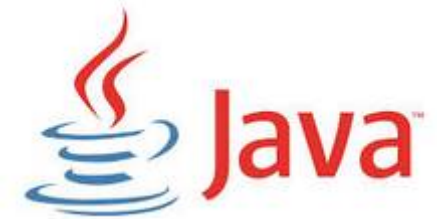


Langages de programmation:



Un ordinateur ne peut pas comprendre autre chose que le code binaire formé de "0" et de "1".

Exemple:

le code binaire correspondant à une affectation simple telle que

$$A \leftarrow B + C . D$$

est :

000100111011000101001010101010

010000101010101010101010101010

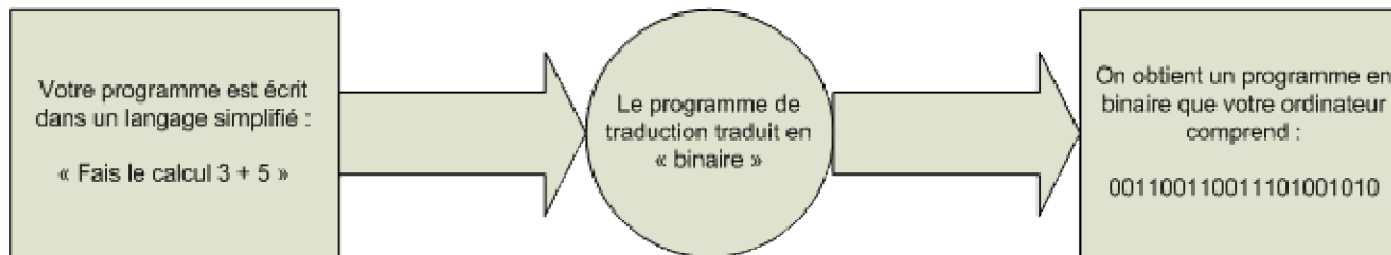


Conclusion:

Un humain peut difficilement lire ou rédiger un programme d'ordinateur écrit sous forme de "0" et de "1".

Pour faciliter la communication entre l'ordinateur et l'homme, il a fallu inventer des nouveaux langages .

Ces langages, facilement compréhensibles par l'homme et écrits (en anglais) selon une syntaxe bien précise, sont **traduits** en suite **en binaire**.



```
if a = b then
begin
a = c+d;
d = d+1;
end
```

Code source



Compilation

```
01101100 10010101 11011100
10101100 11001110 11011010
01010110 11001010 10001011
01101100 10010101 11011100
01010110 11001010 10001011
```

Code binaire

La traduction peut se faire de deux manières différentes:

Le compilateur:

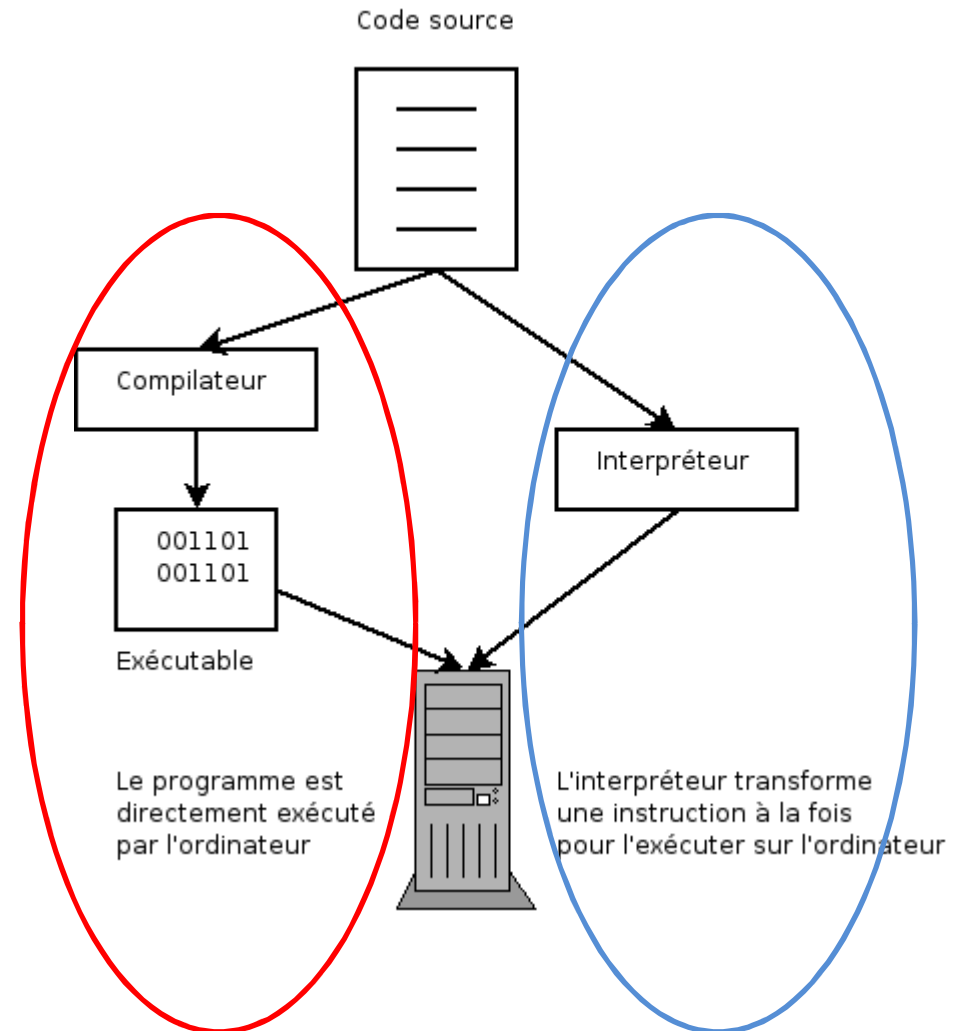
Un compilateur prend le programme saisi par l'utilisateur et le traduit en code machine. L'ordinateur exécute alors le code machine.

L'interpréteur:

Ces programmes lisent le code source et le traduisent en langage machine à la volée.

Remarque:

l'utilisation d'un interpréteur rendra l'exécution plus lente que pour un programme exécuté directement en langage machine.



Quelques langages :

Langages de programmation compilés :

- Java
- C
- C++
- C#

Langages de programmation interprétés :

- Python
- Ruby

Langages de programmation web :

- JavaScript
- PHP
- HTML

Langage pour les données :

- SQL
- MATLAB



[tutorialspoint.com](https://www.tutorialspoint.com)

✓ En 1989, le hollandais **Guido van Rossum** a commencé le développement du langage de programmation Python.

✓ Actuellement, Python en est à sa version 3.3.2. Le langage Python est gratuit, sous **licence libre**.

✓ Python est un langage **multiplateforme**, c'est-à-dire disponible sur plusieurs architectures (compatible PC, certains smartphones, Raspberry Pi...) et systèmes d'exploitation (Windows, Linux, Mac, Android pour smartphones...).

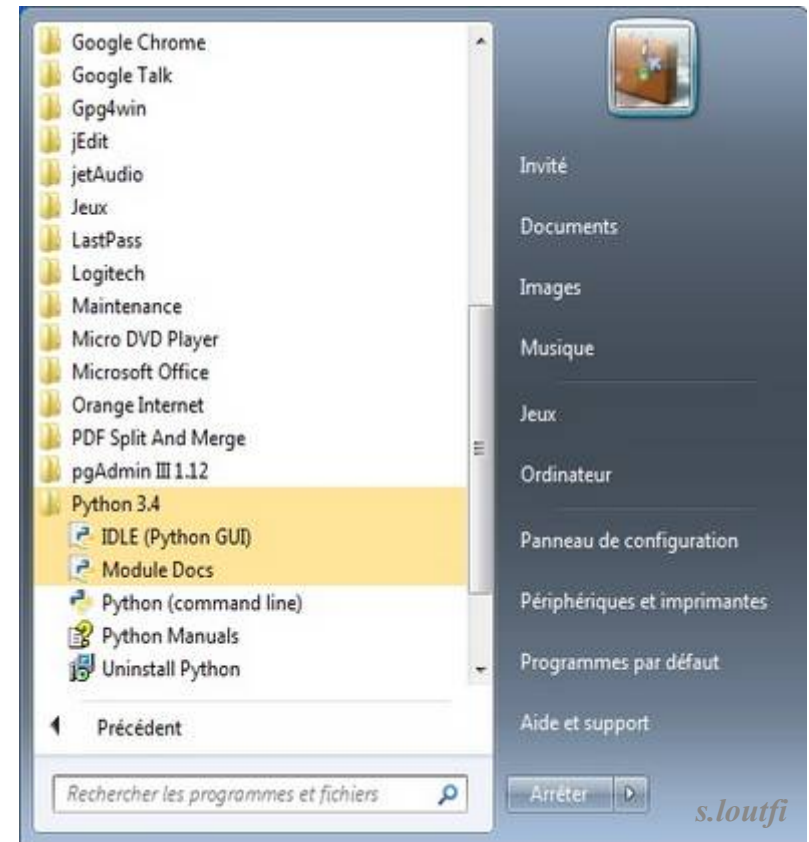
✓ Python possède plusieurs bibliothèques permettant de faire :

- du calcul scientifique (bibliothèque NumPy)
- des graphiques (bibliothèque matplotlib)
- du traitement d'image (bibliothèque PIL)
- des applications avec interface graphique (bibliothèques Tkinter, wxPython, PyGTK ...)
- des jeux vidéo en temps réel (bibliothèque Pygame)
- des applications Web (serveur Web Zope ; framework Web Django, Karrigell ; framework JavaScript Pyjamas)
- interfacier des systèmes de gestion de base de données (bibliothèque MySQLdb ...)
- des applications réseau (framework Twisted)
- communiquer avec des ports série RS232, Bluetooth... (bibliothèque PySerial)

Installation sous Windows :

Sous Windows, pour installer la **version 3.3.2** de Python avec l'**environnement de développement IDLE**, il suffit de télécharger puis d'exécuter le fichier d'installation qui se trouve sur le site officiel : <http://www.python.org/getit>

Une fois installé, vous pouvez lancer **IDLE** en allant dans :
Démarrer → Programmes → Python 3.3.2 → IDLE (Python GUI)

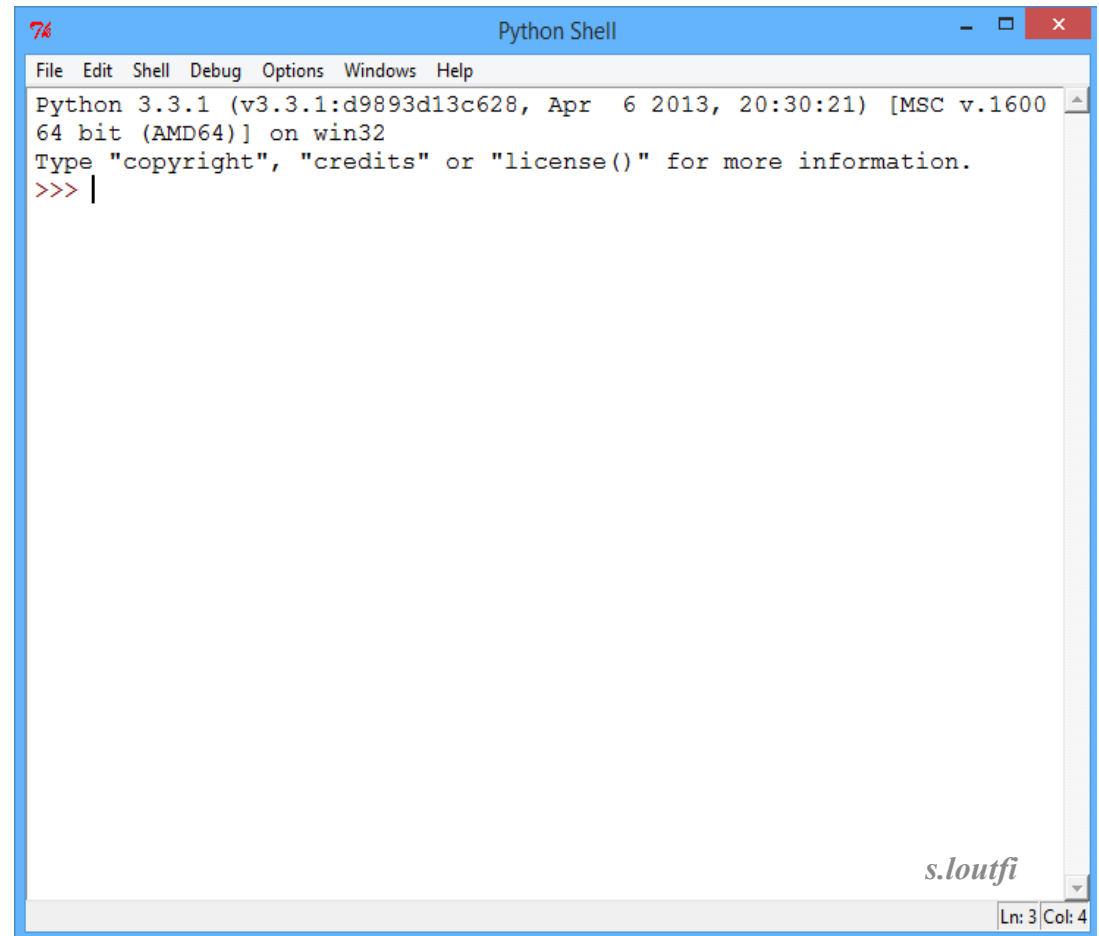


C'est quoi IDLE ?

IDLE est un environnement de développement intégré (Integrated Development Environment) pour Python.

Il propose un certain nombre d'outils :

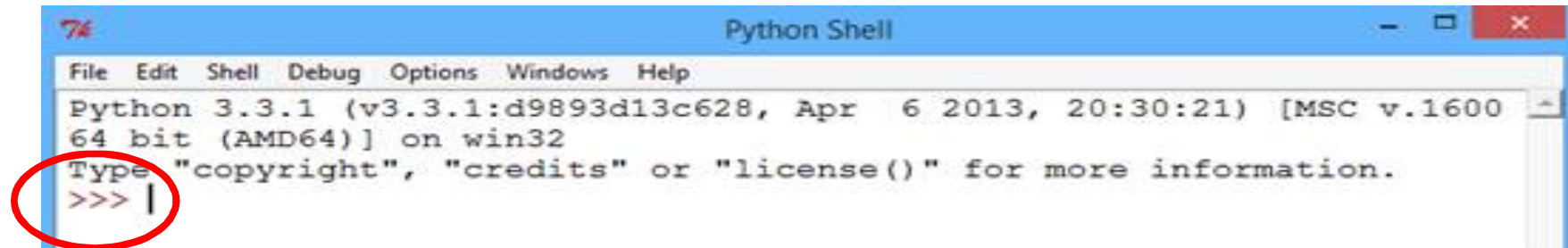
- un éditeur de texte:
pour écrire le programme.
- un interpréteur
pour exécuter le programme.
- un débogueur
pour tester le programme.



```
Python 3.3.1 (v3.3.1:d9893d13c628, Apr 6 2013, 20:30:21) [MSC v.1600  
64 bit (AMD64)] on win32  
Type "copyright", "credits" or "license()" for more information.  
>>> |
```

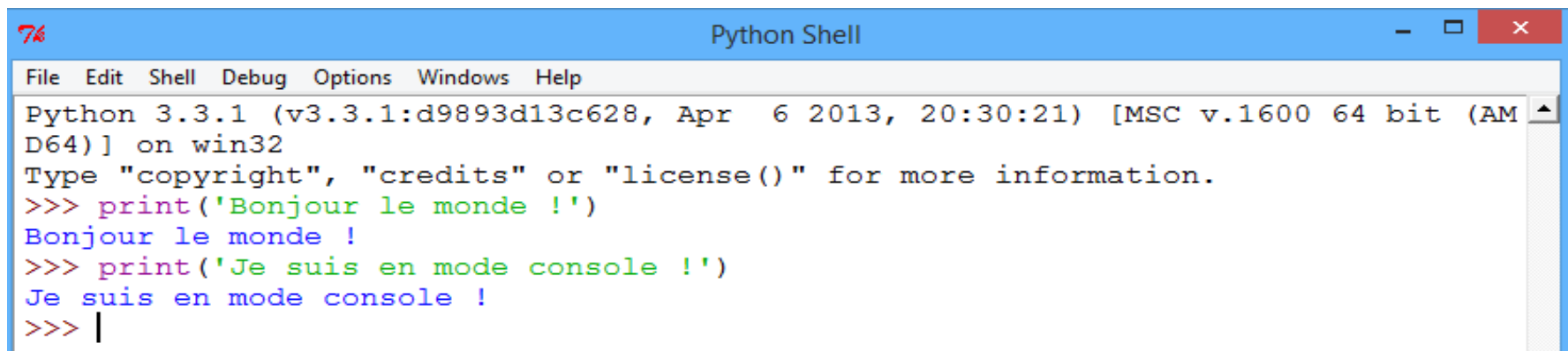
Python en mode interactif

✓ Lorsqu'on ouvre l'environnement IDLE on obtient une fenêtre appelée interpréteur (ou console, ou terminal, ou shell).



```
Python Shell
File Edit Shell Debug Options Windows Help
Python 3.3.1 (v3.3.1:d9893d13c628, Apr 6 2013, 20:30:21) [MSC v.1600
64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> |
```

✓ Là nous pouvons utiliser le langage Python en mode interactif. Les expressions sont rédigées à la suite des chevrons >>> puis évaluées avec la commande *Entrée*.



```
Python Shell
File Edit Shell Debug Options Windows Help
Python 3.3.1 (v3.3.1:d9893d13c628, Apr 6 2013, 20:30:21) [MSC v.1600 64 bit (AM
D64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print('Bonjour le monde !')
Bonjour le monde !
>>> print('Je suis en mode console !')
Je suis en mode console !
>>> |
```

Remarque: Les trois chevrons >>> forment ce que l'on appelle le « prompt » ou « invite » s.loutfi

Python en mode interactif

Introduisez les instructions suivantes puis valider en appuyant sur « Entrée ».
Notez le résultat obtenu à chaque fois.

1. Opérations de base

```
>>> 2+3
```

```
>>> 2*3
```

```
>>> 2**3
```

```
>>> 5/2
```

```
>>> 5//2
```

```
>>> 5%2
```

```
>>> 2-5*3
```

2. Variables:

```
>>> age= 18
```

```
>>> age
```

```
>>> print('mon age est ',age,'ans')
```

```
>>> nom='Dupond'
```

```
>>> prenom='Pierre'
```

```
>>> identite=nom+' '+prenom
```

```
>>> identite
```

3. Affichage :

```
>>> print('hello')
```

```
>>> print(hello)
```

```
>>> print('2+3')
```

```
>>> print(2+3)
```

```
>>> print('la somme de 2+3 est :',2+3)
```

4. Types de variables:

```
>>> a=5
```

```
>>> type(a)
```

```
>>> b=5.0
```

```
>>> type(b)
```

```
>>> c='hello'
```

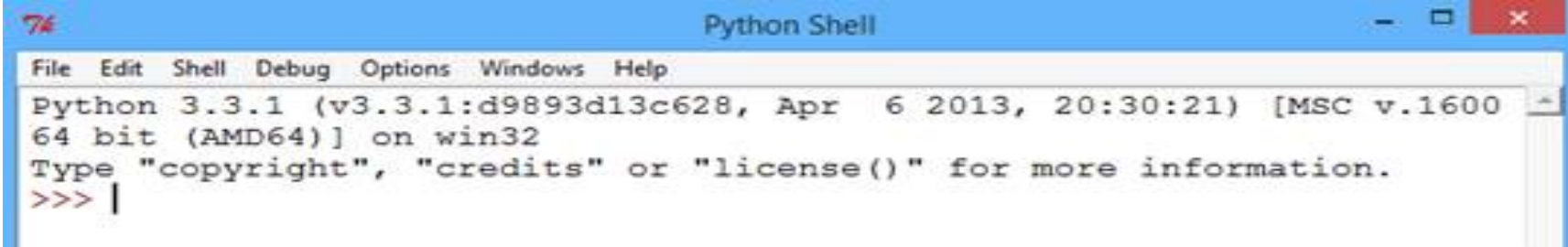
```
>>> type(c)
```

```
>>> d=True
```

```
>>> type(d)
```

Python en mode script

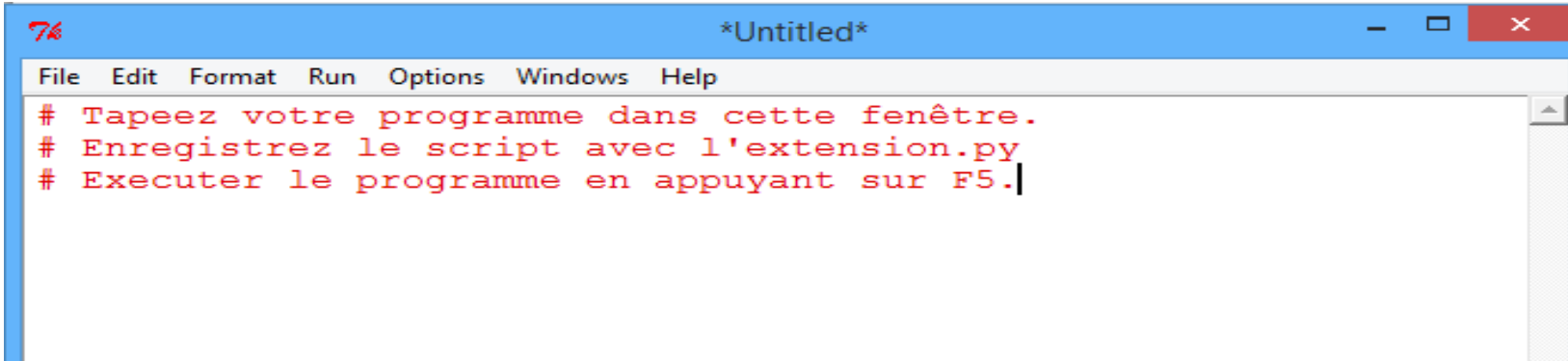
✓ Lorsqu'on ouvre l'environnement IDLE on obtient une fenêtre appelée interpréteur (ou console, ou terminal, ou shell).



```
Python 3.3.1 (v3.3.1:d9893d13c628, Apr 6 2013, 20:30:21) [MSC v.1600  
64 bit (AMD64)] on win32  
Type "copyright", "credits" or "license()" for more information.  
>>> |
```

✓ Là nous pouvons utiliser le langage Python en mode script afin de sauvegarder nos programmes pour pouvoir les réutiliser:

File → New Window (ouverture d'un éditeur de texte).



```
# Tapez votre programme dans cette fenêtre.  
# Enregistrez le script avec l'extension.py  
# Executer le programme en appuyant sur F5.|
```

Python en mode script

Mon premier programme :

- ✓ Lancer Python en cliquant sur IDLE (Python GUI),
- ✓ faire File – New Window,
- ✓ taper le programme ci dessous,
- ✓ le sauvegarder (File - Save as) : **somme.py**
- ✓ le lancer en appuyant sur F5.

```
File Edit Format Run Options Windows Help
#entrer le premier nombre
a=int(input('entrer un nombre entier a='))
#entrer le deuxieme nombre
b=int(input('entrer un nombre entier b='))
#realiser la somme
somme = a+b
#afficher la resultat
print('la somme de a et de b est ;',somme)
```

la fonction « input »

```
var=(input('texte'))
# affiche la phrase texte et attend
# une entrée au clavier qui sera stockée
# dans la variable var ( !!type(var)=str))
```

Programmes à réaliser

Programme 1 : surface et périmètre d'un rectangle

Ecrivez un programme qui permet de calculer la surface et le périmètre d'un rectangle.
L'utilisateur donnera la largeur et la longueur du rectangle .

Programme 2 : vitesse d'un mobile

Ecrivez un programme qui permet de calculer la vitesse d'un mobile.
L'utilisateur donnera la distance parcourue ainsi que le temps de parcours.

Programme 3 : convertisseur de température

Ecrivez un programme qui permet de donner la valeur de température en Kelvin et en Fahrenheit
L'utilisateur donnera la température en degré Celsius.

Programme 4 : fiche d'adhérent

Ecrivez un programme qui permet de donner le résultat suivant:

```
Adherent N°:34  
Nom : Dupond  
prenom : Remi  
age :18  
adresse :4 rue de la paix Metz
```

Les exceptions

Le programme ci-dessous permet d'afficher le nombre entier entré par l'utilisateur

```
File Edit Format Run Options Windows Help
var=int(input('entrez un entier : '))
print('le nombre entier est:',var)
```

Entrée correcte : exécution correcte du programme

```
>>>
entrez un entier : 25
le nombre entier est: 25
>>> |
```

Entrée erronée : affichage d'un message d'erreur et sortie du programme.

```
>>>
entrez un entier : A
Traceback (most recent call last):
  File "C:/Python33/input2.py", line 1, in <module>
    var=int(input('entrez un entier : '))
ValueError: invalid literal for int() with base 10: 'A'
>>> |
```

On parle de levée d'exception de type `valueError`

Gestion des exceptions

Pour gérer les exceptions et éviter l'arrêt brutal du programme, on utilise conjointement les instructions try et except.

✓ Le programme ci-dessous (à tester) permet de garder la main après **une** erreur:

```
File Edit Format Run Options Windows Help
var=(input('entrez un entier : '))|
try:
    var=int(var)
except:
    # ce bloc s'exécute si une exception est levée
    print('ce n\'est pas un entier')
    var=(input('Attention il faut un entier : '))
print(var)
```

✓ Le programme ci-dessous (à tester) permet de garder la main **tant que** le nombre entré n'est pas bon !:

```
File Edit Format Run Options Windows Help
var=(input('entrez un entier : '))
while True:
    try:
        var=int(var)
        break # on sort de la boucle while|
    except:
        # ce bloc s'exécute si une exception est levée
        print('ce n\'est pas un entier')
        var=(input('Attention il faut un entier : '))
print(var)
```


Le module math

Qu'est ce qu'un module?

Un **module** est un **fichier** dans lequel on range des fonctions se rapportant à une même application. Il est sauvegardé sous forme d'un fichier dont le nom a la forme *nomdumodule.py*

Comment utiliser un module?

Pour utiliser un module, il faut l'importer. Pour cela il faut utiliser une des instructions suivantes au début du script :

✓ `from nomdumodule import *` : permet d'importer toutes les fonctions du module.

```
>>> from math import *
>>> sqrt(36)
6.0
>>> cos(0)
1.0
```

à tester

✓ `from nomdumodule import fonction` : permet d'importer uniquement la *fonction* .

```
>>> from math import sqrt
>>> sqrt(36)
6.0
>>> cos(0)
Traceback (most recent call last): File "<stdin>", line 1, in <module>
NameError: name 'cos' is not defined
```

à tester

Programmes à réaliser

Programme 1 : Surface et périmètre d'un cercle

Ecrivez un programme qui permet de calculer le périmètre et la surface d'un cercle.
L'utilisateur donnera le diamètre du cercle.

Programme 2 : Loi de Wien

Ecrivez un programme qui permet de calculer température de la surface d'un astre .
L'utilisateur donnera la longueur d'onde de l'intensité lumineuse maximale émise par l'astre.

Programme 3 : Forces gravitationnelles

Ecrivez un programme qui permet de calculer la force exercée par la terre sur une autre planète.
L'utilisateur donnera la planète en question , sa masse et la distance qui sépare le centre des deux planètes.

Programme 4 : Poids d'un corps en fonction de l'altitude

Ecrivez un programme qui permet de calculer le poids d'un corps en fonction de l'altitude.
L'utilisateur donnera la valeur de l'altitude souhaitée.