

Programmation Python

Youssef ALJ

Généralités

Pourquoi Python?

- Principales caractéristiques du langage Python :
 - Langage **interprété**.
 - Langage **multi-paradigme**.
 - Langage **multi-plateforme**.
- On peut faire plein de choses avec Python :
 - Développement web (Django, Flask)
 - Très utilisé dans le domaine de la science des données
 - ...

Langage interprété vs. Compilé

Langage interprété :
Interprète analyse et
traduction du programme
pour chaque exécution.

Exemples :

Lisp

Matlab

Python

...

Langage compilé : Le
compilateur effectue les
opérations d'analyse et
de traduction une fois
pour toutes.

Exemples :

C

C++

Java

...

Langage multi-paradigme

- Paradigme objet :
 - Définir une entité appelée classe qui rassemble données et traitements.
- Programmation fonctionnelle :
 - Le programme est écrit sous forme d'une ou plusieurs fonctions.

Langage multi-plateforme

- Le programme écrit en windows fonctionne sous linux, macos, etc.

Mise en place de l'environnement de programmation

Installation de Python

- Télécharger Python avec anaconda :
 - <https://www.anaconda.com/products/individual>
 - Cliquer sur Download puis choisir la plateforme de votre PC.
 - Suivre les instructions pour finir l'installation.
- Aller dans windows → programmes → anaconda → spyder.

Spyder (Python 3.7)

Fichier Édition Recherche Source Exécution Débuguer Consoles Projets Outils Affichage Aide

C:\Users\Public\Documents\enseignement_pub\enseignements\python\cours\programmes

C:\Users\Public\Documents\enseignement_pub\enseignements\python\cours\programmes\variables.py

```
1 a = 5
2 b = 3.0
3 c = "salut "
4 L = [1, 2, 3]
5 T = (4, 5, 6)
6
```

Nom	Type	Taille	Valeur
L	list	3	[1, 2, 3]
T	tuple	3	(4, 5, 6)
a	int	1	5
b	float	1	3.0
c	str	1	salut

Aide Explorateur de variables Graphes Fichiers

Console 1/A

```
In [31]: a + b
Out[31]: 8.0

In [32]: a * b
Out[32]: 15.0

In [33]: 5 * a
Out[33]: 25

In [34]: c + c
Out[34]: 'salut salut '

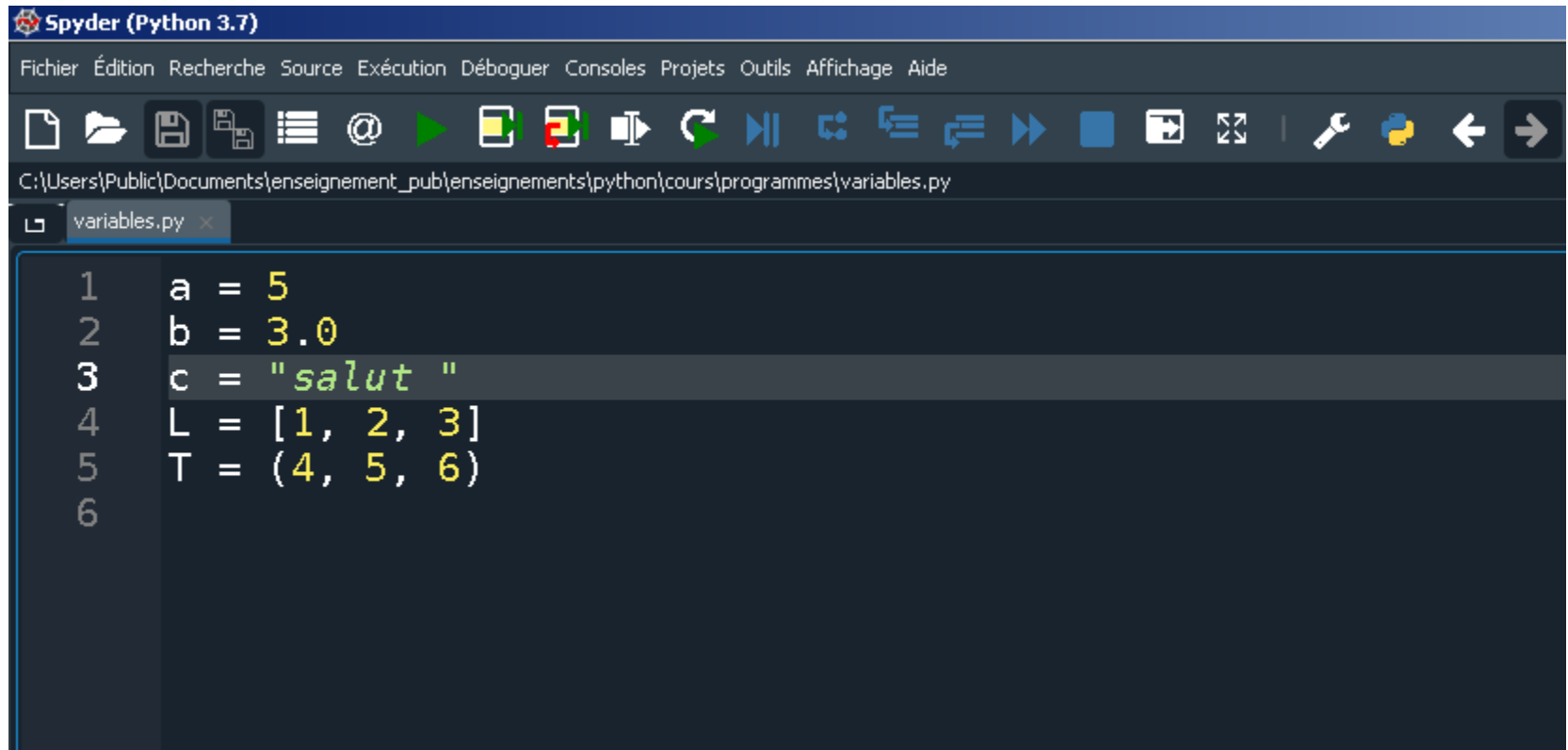
In [35]: 3 * c
Out[35]: 'salut salut salut '

In [36]: |
```

Ouvrir un fichier

conda: base (Python 3.7.6) master [5] Line 3, Col 12 UTF-8 CRLF RW Mem 67%

L'éditeur



The image shows the Spyder Python IDE interface. The title bar reads "Spyder (Python 3.7)". The menu bar includes "Fichier", "Édition", "Recherche", "Source", "Exécution", "Déboguer", "Consoles", "Projets", "Outils", "Affichage", and "Aide". The toolbar contains various icons for file operations, execution, and debugging. The status bar shows the file path: "C:\\Users\\Public\\Documents\\enseignement_pub\\enseignements\\python\\cours\\programmes\\variables.py". The editor window displays the following code:

```
1 a = 5
2 b = 3.0
3 c = "salut "
4 L = [1, 2, 3]
5 T = (4, 5, 6)
6
```

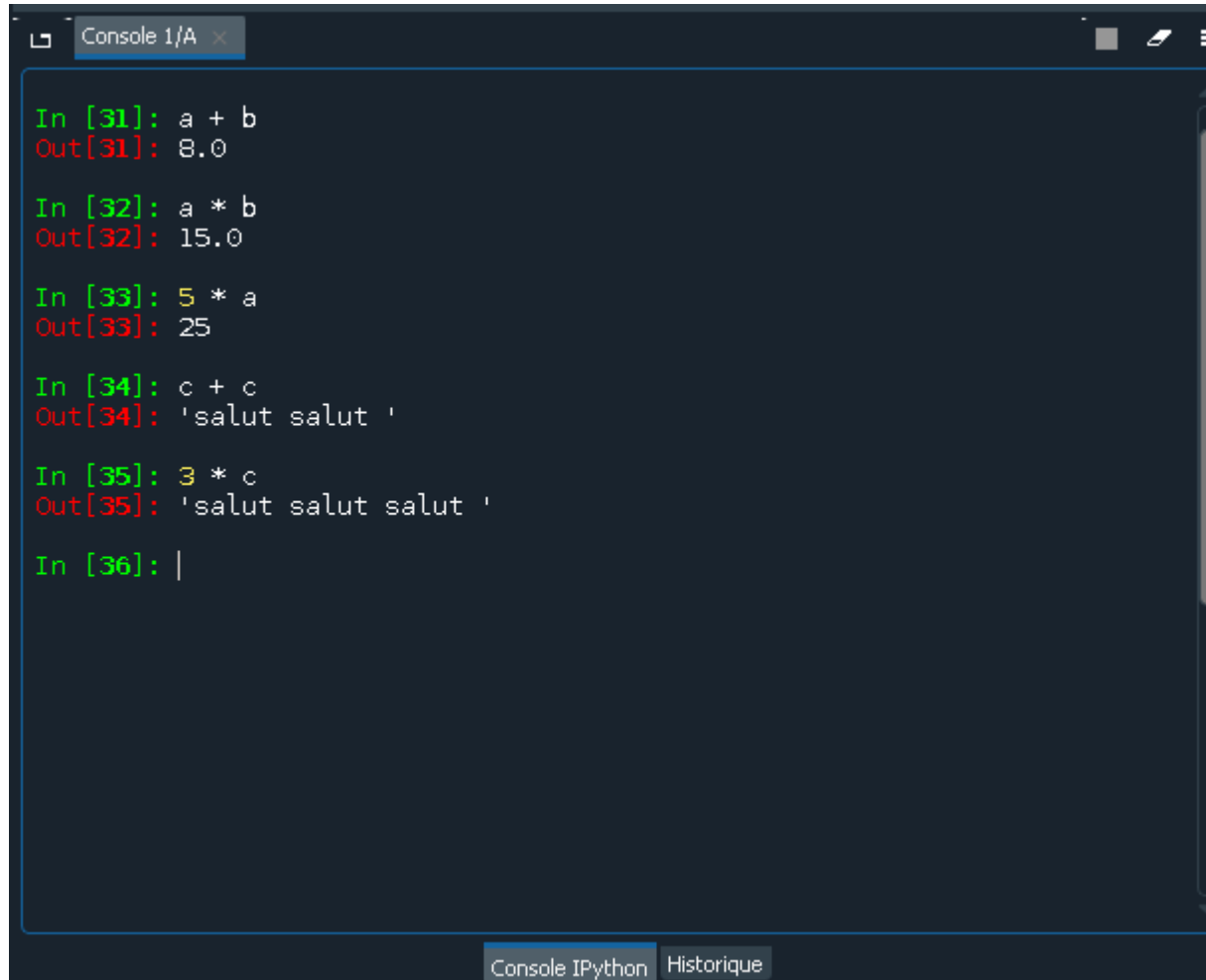
L'explorateur de variables



Nom	Type	Taille	Valeur
L	list	3	[1, 2, 3]
T	tuple	3	(4, 5, 6)
a	int	1	5
b	float	1	3.0
c	str	1	salut

Aide Explorateur de variables Graphes Fichiers

L'interprète



The image shows a screenshot of a Jupyter Notebook's console window. The window has a title bar with a close button and a tab labeled 'Console 1/A'. The console displays a series of Python input and output pairs. The inputs are shown in green, and the outputs are shown in red. The code includes arithmetic operations and string concatenation. At the bottom of the window, there are two buttons: 'Console IPython' and 'Historique'.

```
Console 1/A x
```

```
In [31]: a + b
Out[31]: 8.0

In [32]: a * b
Out[32]: 15.0

In [33]: 5 * a
Out[33]: 25

In [34]: c + c
Out[34]: 'salut salut '

In [35]: 3 * c
Out[35]: 'salut salut salut '

In [36]: |
```

Console IPython Historique

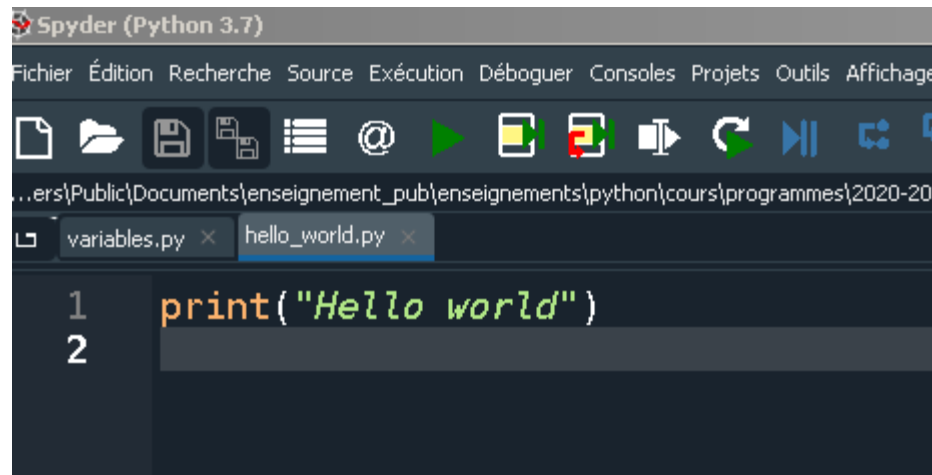
Utilisation de l'éditeur et de l'interprète

- L'éditeur :
 - Pour écrire des fonctions des programmes.
 - Toute opération qu'on veut sauvegarder.
- L'interprète :
 - Les petits calculs.
 - Tests des fonctions écrites dans l'éditeur ou des fonctions python.
 - Tout ce qui n'a pas vocation à être sauvegardé.
- L'explorateur de variables :
 - Permet de visualiser l'état courant des variables après exécution.

Éléments du langage

Hello world

- Notre premier programme sera d'afficher Hello world à l'écran.



The image shows a screenshot of the Spyder Python IDE interface. The title bar at the top reads 'Spyder (Python 3.7)'. Below it is a menu bar with options: 'Fichier', 'Édition', 'Recherche', 'Source', 'Exécution', 'Déboguer', 'Consoles', 'Projets', 'Outils', and 'Affichage'. A toolbar with various icons for file operations and execution is located below the menu bar. The file explorer on the left shows a directory path: '...ers\Public\Documents\enseignement_pub\enseignements\python\cours\programmes\2020-2021'. The file explorer shows two files: 'variables.py' and 'hello_world.py', with 'hello_world.py' selected. The main editor window displays the following Python code:

```
1 print("Hello world")
2
```

Commentaires

```
1  # Ceci est un commentaire sur une ligne
2
3  ◀  """
4     Ceci est
5     un commentaire
6     sur plusieurs lignes
7     """
```


La fonction print

- La fonction print permet d'afficher un message.
- Elle permet aussi d'afficher le contenu de n'importe quelle variable python.

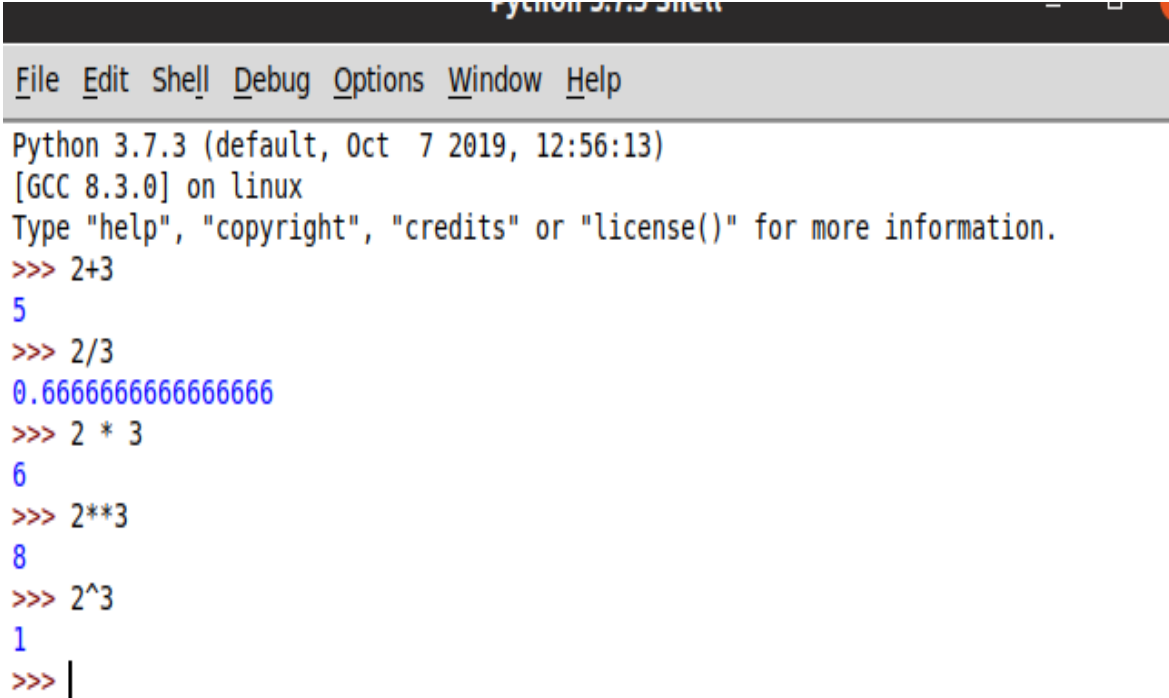
```
a = 2
b = 3

produit = a * b

print("Le produit de a et b est", produit)
# Une manière plus élégante pour l'affichage
print("Le produit de", a, "et", b, "est", produit)
```

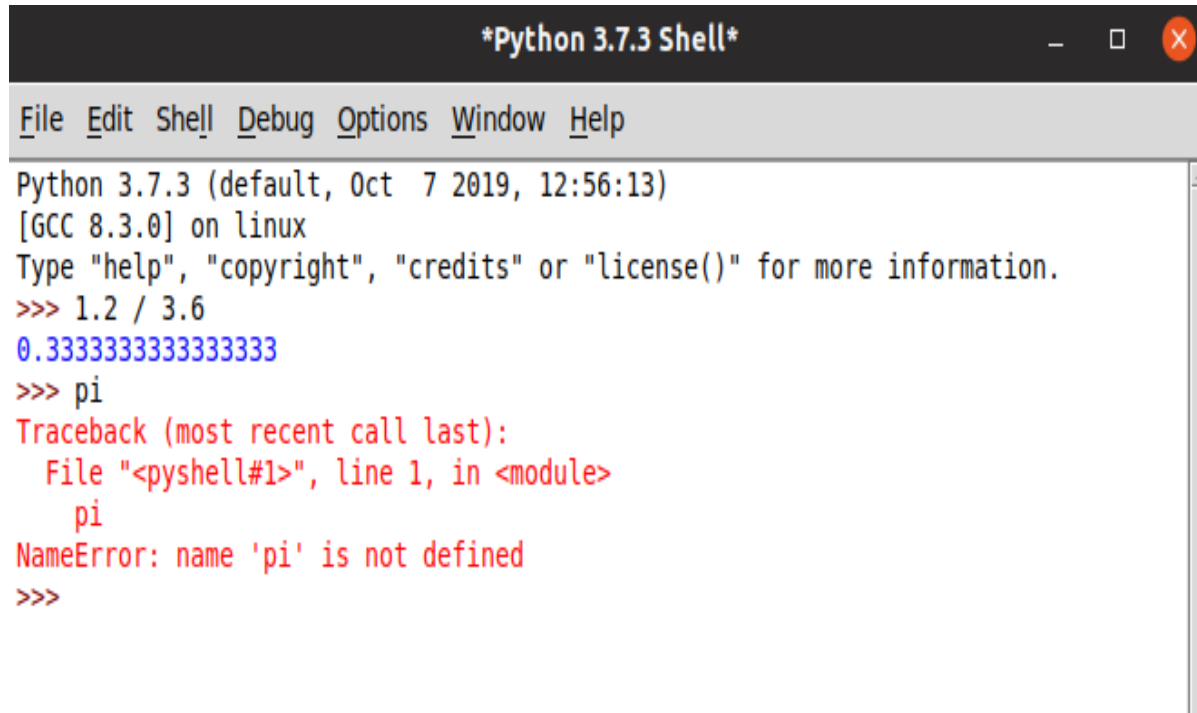
Les entiers

- Les diverses opérations



```
Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
Python 3.7.3 (default, Oct 7 2019, 12:56:13)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>> 2+3
5
>>> 2/3
0.6666666666666666
>>> 2 * 3
6
>>> 2**3
8
>>> 2^3
1
>>> |
```

Les flottants (1ere partie)



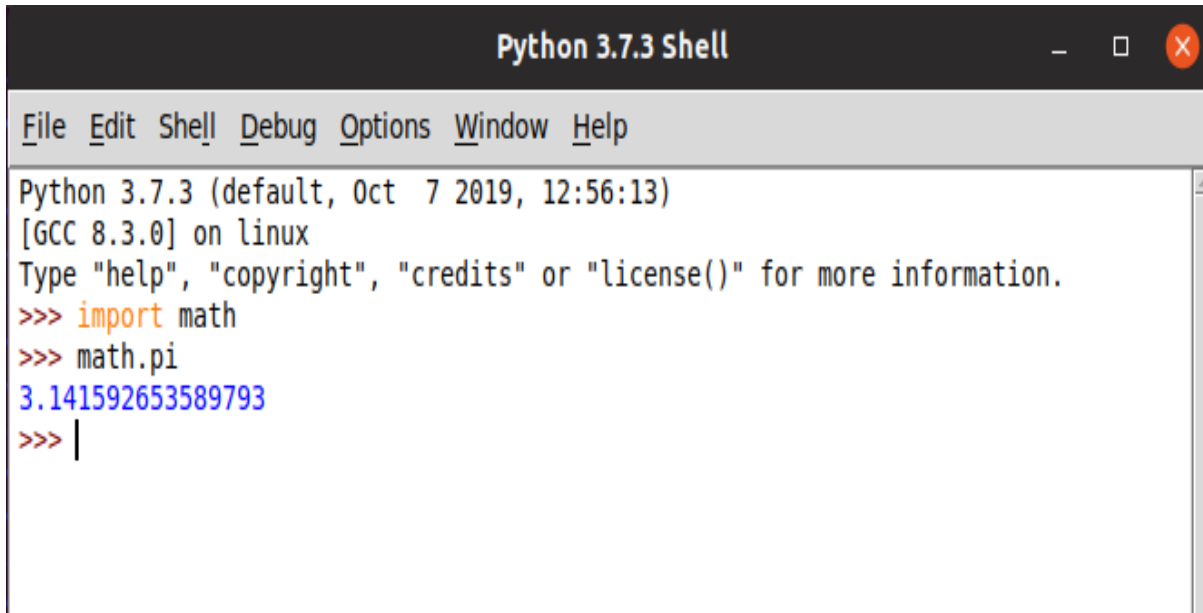
```
*Python 3.7.3 Shell*  
File Edit Shell Debug Options Window Help  
Python 3.7.3 (default, Oct 7 2019, 12:56:13)  
[GCC 8.3.0] on linux  
Type "help", "copyright", "credits" or "license()" for more information.  
>>> 1.2 / 3.6  
0.3333333333333333  
>>> pi  
Traceback (most recent call last):  
  File "<pyshell#1>", line 1, in <module>  
    pi  
NameError: name 'pi' is not defined  
>>>
```

Les erreurs en Python

- La ligne la plus importante : la dernière.
 - Ex : `NameError: name 'pi' is not defined`
- Les lignes précédentes : le « traceback » :
 - Le traceback permet de trouver rapidement l'origine de l'erreur
 - `most recent call last` : l'appel le plus récent est affiché à la fin.
- Dans notre exemple, Python ne connaît pas 'pi'

Les erreurs en Python

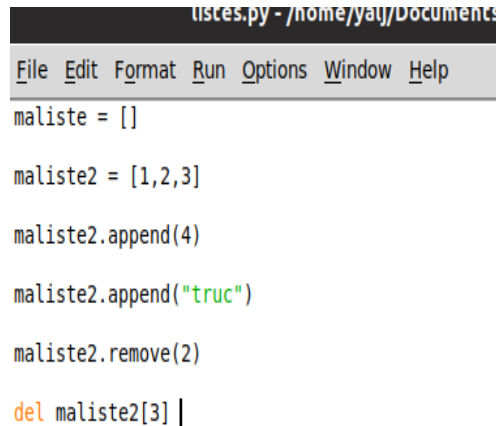
Pour y remédier on charge le module de maths.



```
Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
Python 3.7.3 (default, Oct 7 2019, 12:56:13)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>> import math
>>> math.pi
3.141592653589793
>>> |
```

Listes

- Explorer les résultats des commandes suivantes :



```
listes.py - /home/yay/Documents
File Edit Format Run Options Window Help
maliste = []
maliste2 = [1,2,3]
maliste2.append(4)
maliste2.append("truc")
maliste2.remove(2)
del maliste2[3] |
```

Les listes sont mutables : on peut modifier leur contenu une fois créée.

Tuples

- Les tuples **ne sont pas mutables**.

```
A = (1, 2, 3)
```

```
B = (4, 5, 6, 7)
```

```
>>> A[0]
```

```
1
```

```
>>> A[1] = 6
```

```
Traceback (most recent call last):
```

```
File "<pyshell#14>", line 1, in <module>
```

```
A[1] = 6
```

```
TypeError: 'tuple' object does not support item assignment
```

Structures de contrôle

La boucle for

Utilisée pour effectuer n fois une certaine tâche où n est défini par le programmeur.

Analyser le programme suivant :

```
boucle_for.py - /home/yalj/Documents/enseignement/python/cours_new  
File Edit Format Run Options Window Help  
for i in range(5):  
    print(i)
```

Chercher sur internet la syntaxe de 'range'

Exercice : afficher les nombres entiers naturels pairs inférieures ou égaux à 100.

Boucle While

Utilisée pour effectuer une tâche tant qu'une condition est réalisée.

Exemple : analyser la boucle suivante :

```
i=0
while 2 ** i < 2019 :
    print (2 ** i)
    i = i + 1
```

- **Exercice** : écrire un programme qui affiche le plus petit entier n tel que $4 + 5 + 6 + \dots + n$ dépasse 12345

Tests

- On utilise 'if' !!!!
- Attention on teste l'égalité avec '=='.
- Exemple :

```
print("Entrer un nombre plus petit que 100.")
nb = float(input())

if nb < 100:
    print("Ce nombre convient.")
else:
    print("Ce nombre est trop grand.")
```

Division Euclidienne

- Quotient de a par b : 'a // b '
- Reste de a par b : 'a % b'
- **Exercice** : écrire un programme qui permet en utilisant le quotient ou le reste de la division euclidienne d'afficher les nombres pairs.
- **Exercice** : écrire un programme qui permet d'afficher tous les nombres impaires inférieurs à 100.