

# Séq. 1 – La modularité dans Python

## Objectifs

- Utiliser de bibliothèques (travailler avec un fichier texte/csv, mesurer un temps d'exécution, dessiner des graphes, ...) ou des API
- Exploiter la documentation
- Créer des modules simples et les documenter.

A NOTER : L'ensemble des informations officielles sur python se trouve sous [www.python.org](http://www.python.org)

A NOTER : Pour les allergiques à l'anglais, le site de l'association francophone python :

[www.afpy.org](http://www.afpy.org)

## 1 Modules « builtin »

Sans avoir à télécharger, ni installer quoi que ce soit, vous avez accès à des modules « builtin » ou « construit avec ». La liste est là : <https://docs.python.org/fr/3/py-modindex.html>

### A faire vous même 1.

La fonction racine carré n'est pas une fonction « builtin ». Il faut aller la chercher dans le module math.

- Dans la console python testez ces lignes :

```
>>> from math import sqrt
>>> sqrt(9)
3
```

- Une autre façon de faire :

```
>>> import math
>>> math.sqrt(9)
3
```

- Ou en utilisant un alias :

```
>>> import math as m
>>> m.sqrt(9)
3
```

### A faire vous même 2.

S'il est bien fait, un module python intègre sa propre documentation.

- Dans la console python testez ces lignes :

```
>>> import math
>>> help(math)
```

- Vous avez la liste de toutes les fonctions disponibles dans ce module avec une courte explication
- Cherchez comment récupérer une valeur approchée du nombre réel pi
- Testez la fonction sinus
- Testez une autre fonction de votre choix

### A faire vous même 3.

L'aide peut être récupérée et lue via des pages web. Dans un shell linux tapez :

```
$ python -m pydoc -p 1234
```

Vous venez de lancer un mini-serveur web

- Consultez la documentation et testez le module `random`
- Écrivez un petit script qui simule le tirage du loto

### A faire vous même 4.

- Pour le plus rapides et/ou les plus motivés, voici une liste de modules intéressants :
  - `time`, `datetime`, `calendar` : gestion de l'heure et de la date
  - `sys` : adaptation de Python au système hôte. Un peu « fourre tout »
  - `os` : se substitue à certaines commandes linux
  - `os.path` : manipulation de chemin de fichiers
  - `stat` : informations sur les fichiers (sécurité, dates, UID, ...)
  - `tempfile` : fichiers et répertoires temporaires

- `glob`, `fnmatch` : recherche et correspondance des fichiers avec patterns Unix (« \*.py? », ...)
- `shutil`, `filecmp` : déplacement, copie, comparaison de fichiers et répertoires

### A faire vous même 5.

- Explorez plus en détail le module : `os.path` : manipulation de chemin de fichiers (<https://docs.python.org/fr/3/library/os.path.html#module-os.path>) et particulièrement :
  - `os.path.abspath(path)`
  - `os.path.basename(path)`
  - `os.path.dirname(path)`
  - `os.path.exists(path)`
  - `os.path.join(path, *paths)`
  - `os.path.split(path)`
- Écrivez un script :
  1. Qui demande à l'utilisateur un nom de fichier
  2. Qui donne le répertoire courant (`os.getcwd` <https://docs.python.org/fr/3/library/os.html#os.getcwd>)
  3. Qui donne le chemin complet du fichier (concaténation du répertoire courant et du nom du fichier)

## 2 Modules pouvant être installés avec pip

La communauté des développeurs de python se sont organisés pour mettre en ligne leurs modules. Tout est sur le site de PyPI (Python Package Index) : <https://pypi.org/>  
Comme l'installation de paquets linux, il existe un outil pour télécharger et installer automatiquement un de ces modules : `pip`

Vous pouvez le faire dans un shell linux :

```
$ pip install pillow #installe le module pil pour manipulation d'image
```

Ou alors directement dans la console python de spyder :

```
>>> !pip install pillow
```

### A faire vous même 6.

Reprenez tous les modules utilisés en 1ère NSI.

- Module `csv` (<https://docs.python.org/fr/3/library/csv.html#module-csv>)
  - Téléchargez un fichier csv : [http://ninoo.fr/LC/Term\\_NSI/seq1\\_modularite/villes\\_virgule.csv](http://ninoo.fr/LC/Term_NSI/seq1_modularite/villes_virgule.csv)
  - Utilisez `csv.DictReader` sur ce fichier
  - Utilisez `csv.DictWriter` pour générer un fichier csv sur tous les élèves de Term NSI avec nom, prenom, date de naissance, classe, nickname
- Module `timeit` (<https://docs.python.org/fr/3/library/timeit.html#module-timeit>)
  - Écrivez une fonction `exposant(a,b)` qui calcule  $a^1$ ,  $a^2$ ,  $a^3$  jusqu'à  $a^b$  et qui retourne toutes ces valeurs dans une liste.
  - Utilisez `timeit.timeit` pour évaluer les temps d'exécution de `exposant(2023,9)`
- Module `matplotlib` (<https://matplotlib.org/>)
  - Téléchargez le fichier des relevés de températures départementales : [http://ninoo.fr/LC/Term\\_NSI/seq1\\_modularite/temperature-quotidienne-departementale.csv](http://ninoo.fr/LC/Term_NSI/seq1_modularite/temperature-quotidienne-departementale.csv)
  - Avec `csv` et `matplotlib`, tracez un graphe de toutes les températures des Cotes d'Armor
  - Ajoutez-y les températures du Var.

### A faire vous même 7.

- Installez le module `pillow` (ou `pil` : <https://pillow.readthedocs.io/en/latest/handbook/index.html>)
- Modifiez une image de votre choix (faites une rotation, une symétrie, ...)

### A faire vous même 8. (pour les rapides)

- Installez le module `numpy`
- Testez ses possibilités

Un article sur les modules python qui peuvent être utiliser en sciences :

<https://linuxfr.org/news/python-pour-les-sciences-une-presentation>

## 3 Construire soi-même ses propres modules

### 3.1 Les Modules

Tous les fonctions, constantes et objets peuvent être enregistrés dans un fichier python (.py). Si ce fichier se trouve dans un répertoire déclaré dans le PYTHONPATH alors il est possible de faire un import.

#### A NOTER :

Quand un module est exécuté ou importé, son corps de programme est automatiquement exécuté. Pour éviter que ce corps de programme soit exécuté à l'import, il faut commencer par écrire :

```
if __name__ == '__main__':  
    #Ici le corps du programme
```

#### A faire vous même 9.

- Construisez un module (nommée aires\_et\_volumes) de fonctions calculant les aires (carré, rectangle, disque, triangle, trapèze, ...)
- Ajoutez-y les calculs de volumes si possible s'appuyant sur un calcul d'aire (prisme droit, cylindre, sphère, cône, pyramide, ...)
- Écrivez un corps de programme proposant un menu permettant d'exécuter les fonctions :  
*Que voulez vous faire ?*  
0- Pour quitter  
1- Pour ...  
2- Pour ...

### 3.2 La documentation

Chaque fonction peut être accompagné d'une docstring (""" ... """).

Cette docstring fournit une documentation ou aide en ligne qu'il est possible d'afficher avec la commande help().

Pour les fonctions, souvent la docstring présente les paramètres d'entrée nécessaires et les valeurs retournées par la fonction.

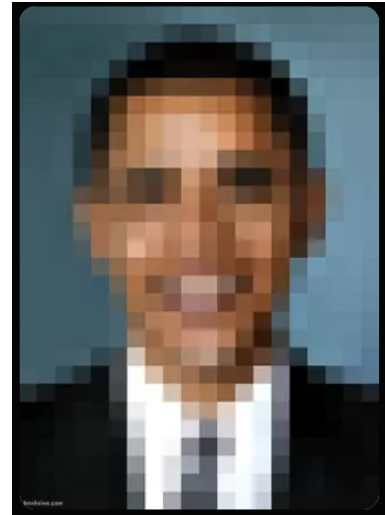
Il est même possible de rédiger une docstring de présentation globale du module.

A NOTER : Le module sphinx permet d'obtenir une documentation style web détaillé.

**A faire vous même 10. !!! MINI-PROJET PAR GROUPE NOTE !!!**

A vous appuyant sur le module PIL, proposez plusieurs fonctions permettant de modifier une image.

1. Écrivez une fonction permettant de faire une symétrie suivant l'axe horizontal
2. Écrivez une fonction permettant de faire de la détection de contour  
([https://snt.ledantec-numerique.fr/wp-content/uploads/2020/03/4\\_traitement\\_image\\_detection\\_contour.pdf](https://snt.ledantec-numerique.fr/wp-content/uploads/2020/03/4_traitement_image_detection_contour.pdf))
3. Soyez original, écrivez une fonction de votre choix permettant de faire un travail sur les couleurs (éclaircissement, contraste, renforcement de certaines couleurs, ...)
4. Soyez original, écrivez une fonction de votre choix permettant de faire un travail sur les positions des pixels (trucage avec fond vert, image dans image, ...)



5. Écrivez un corps de programme proposant un menu permettant d'exécuter les fonctions :  
Que voulez vous faire ?  
0- Pour quitter  
1- Pour ...  
2- Pour ...