**Liste des commandes utiles en Python**

*Le document listes des fonctions utiles dans le cadre de l’UE 4B033, Python for physiology modelling (pypm). Les commandes sont valables pour la version 3.7 de Python.*

Sommaire

[1. Les bases 1](#_Toc2261315)

[1.1. Fonctions de base 1](#_Toc2261316)

[1.2. Les listes 1](#_Toc2261317)

[1.3. Les dictionnaires 1](#_Toc2261318)

[1.4. Les boucles 1](#_Toc2261319)

[1.5. Les fonctions 1](#_Toc2261320)

[1.6. Les fichiers 1](#_Toc2261321)

[2. Les modules 2](#_Toc2261322)

[1.7. Numpy 2](#_Toc2261323)

[3. Truc et astuces 2](#_Toc2261324)

# Les bases

## Fonctions de base

float() Force à avoir un chiffre à virgule

int(x) Force à avoir un chiffre entier

str(x) Indique qu'il s'agit d'une liste de caractères

list(x)

type(x) Type de la variable

dir(" ")

help(" ".upper)

print(x) Affiche à l'écran une ou plusieurs variable

.format()

('\n') Permet un retour à la ligne

range()

len()

imput()

max()

min()

map(rna, s)

reversed()

zip()

enumerate()

str.upper() Passe toutes les données en majuscules

str.count('A')

str.replace()

str.upper()

str.translate()

.search()

.create()

.strip()

.rstrip()

.split()

"".join()

.find()

.endswith()

## Les listes

.append()

.remove()

.index()

.reverse() Inverse la liste « définitivement » ; ne fonctionne pas avec un string

.extend()

## Les dictionnaires

.items()  Renvoie une liste de de tuples => [(clé 1, valeur 1), (clé 2, valeur 2)]

.keys() Renvoie toutes les clés d’un dictionnaire

.values() Renvoie toutes les valeurs d’un dictionnaire

.get() Renvoie la valeur si on donne la clé

.has\_key() Version 2.7 non ?

## Les conditions

### Boucle For

Exemple de raccourcie :

print('A' + ' '.join(str(e) for e in profile[0]))

break

continue

### Boucle While

### If / Else / Elif

### Les comparateurs

!= / == / < / > / =< / =>

## Les fonctions

## Les fichiers

f = open()

if f.mode == ''r'' Applique une condition suivant le type d’ouverture du fichier

.read()  This method returns the bytes read in string

.readline()

.readlines()

.write()

.writelines ()

.close()

## Les classes

# Les modules

## 

Il y a plusieurs façons d’importer des fonctions venant d’autres script python (= modules)

import x Importe tout le module, les fonctions inclus dans le modules devront du coup être appelées comme cela : module.fonction() donc ici x.fonction()

from x import fonction1 Cela permet d’importer qu’une seule fonction d’un module. Le problème c’est que si on a déjà crée une fonction1 dans notre script, le script plante car il ne sait plus à quelle fonction se référé (celle du script ou celle du module).

import module as x Permet de modifier le nom du module quand on importe une fonction venant de celui là. module.fonction() devient x.fonction()

import x O

## Random

import random

random.randint()

## Numpy

import numpy as np

np.linspace()

np.zeros()

np.asarray()

np.empty()

np.cumsum()

np.expand\_dims()

np.random.rand()

## Matplotlib

import matplotlib as plt

plt.plot()

plt.show()

## Math

import math

math.sqrt()

math.factorial()

## Scipy

import scipy

scipy.stats.rvs()

## Itertools

import itertools

itertools.permutations()

## BioPython

import Bio

Bio.SeqIO.parse()

Bio.SeqUtils.GC() Compte le % de GC dans une séquence

Bio.Seq()

Bio.Alphabet.IUPAC.unambiguous\_rna

Bio.Alphabet.IUPAC.unambiguous\_dna

Bio.SeqIO.parse()

## pip

# Truc et astuces

if \_\_name\_\_ == '\_\_main\_\_':

print(x)

Si ce code est exécuté en tant que script principal (appelé directement avec Python et pas importé comme un module), alors la partie en dessous est exécuter (ici la fonction print() ).

# Autre méthode  
# b = b + a  
# a = b => ça c'est faux  
  
# # il faut faire  
# c = b + a  
# a = b  
# b = c  
# # ou / équivalent :  
# a, b = b, a + b