

PARTIE 2: ALGORITHMIQUE & PROGRAMMATION

Chapitres 1 & 2 – Introduction à l'algorithmique et à la programmation

1 Typage des variables

Variable : une variable permet de stocker des informations. Elle est définie par un identifiant (un nom), un type, une valeur, une référence et des opérations.

Référence : une référence est un alias pointant vers une adresse mémoire.

Type: le typage correspond à la nature d'une variable (entier, chaîne de caractères, liste ...). En Python, le typage est **dynamique** ce qu'il signifie que Python reconnait le type de variable dès son affectation.

Valeur: la valeur correspond à la donnée que l'utilisateur veut stocker en mémoire.

Les objets non mutables sont ceux que l'on ne peut modifier après leur création : les entier (int), les flottants (flt), les chaines de caractères (str), les booléens, les tuples etc... Si on modifie un de ces types, une nouvelle référence est crée.

Les objets mutables peuvent être modifiés après leur création : les listes, les dictionnaires...

python

On ne peut pas modifier la variable ch sans changer de référence. C'est un type non mutable.

On peut modifier des éléments variable ch sans changer de référence. C'est un type mutable.

```
>>> ch = [a,b,c,d,e]
>>> ch[0] = 'a'
```

Exemple

Modification des listes

Modification des chaînes de caractères

2 Instructions et expressions

finition

Instruction

Expression

Exemple

3 Structures algorithmiques



éfinition

Structure Tant Que : une suite d'instructions continue à être réalisée tant qu'une condition prédéfinie ne change pas.

Structure conditionnelle Si, Alors, Sinon : permet d'exécuter une suite d'instructions suivant une condition booléenne.

Boucle Pour : boucle itérative qui se répète un nombre fini de fois.

```
Implémentation de la fonction factorielle
    n=4
    res=0
                                                                  res = 0
    if n==0:
                                                                  if n==0:
         res = 1
                                                                      res = 1
    else
                                                                  else :
        i=1
                                                                      res = 1
         res = 1
                                                                      for i in range(0,n):
        while i <= n:
                                                                          res = res*(i+1)
             res=res*i
             i+=1
```

4 Structure d'un programme

```
Bibliothèque
<tmp 1> × = biblio_rugosite.py × = main_rugosite.py

    Programme principal (main)

                                                                                                                                                          biblio_rugosite.py main_rugosite.py
                                                                                                                biblio_rugosite
                                                                                                                                               #!/usr7bin/env python

# .*- coding: utf-8 .*-
author = "Xavier Pessoles"
email = "xpessoles.ptsi@free.fr"
  #!/usr/bin/env python
# -*- coding: utf-8 -*-
author = "Xavier Pessoles"
_email_ = "xpessoles.ptsi@free.fr"
  import numpy as np
from scipy import fftpack
import matplotlib.pyplot as plt
                                                        Import de bibliothèques de fonction
                                                                                                                                                import math
from random import randint
                                                                                                         Signature de la fonction def generate profil(f0,a0,df0,f1,a1,f2,a2,f3,a3,b,L,nb):
  sys.path.append('C:\\Enseignement'
                                                   b\\Informatique\\Projets\\l3_Rugosite\\progr
                                                                                                         generate_profil
                                                                                                                                                   I Génération d'un profil de rugosité
  import biblio_rugosite
                                                        Appel d'une fonction de la
  f0,a0,df0 = .5,1,0.7
f1,a1 = 20,0.05
f2,a2 = 50, 0.01
bruit = 0.2
                                                        bibliothèque biblio_rugosite
                                                                                                             Spécifications de la
                                                                                                             fonction generate_profil
 x,y = biblio_rugosite.generate_profil(f0,a0,df0,f1,a1,f2,a2,0,0,bruit,0.5,2000)|
```