

PARTIE 2 : ALGORITHMIQUE & PROGRAMMATION

CHAPITRES 1 & 2 – INTRODUCTION À L'ALGORITHMIQUE ET À LA PROGRAMMATION

1 Typage des variables

Définition

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 reconnaît 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 entiers (int), les flottants (float), les chaînes de caractères (str), les booléens, les tuples etc... Si on modifie un de ces types, une nouvelle référence est créée.

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

Exemple



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

```
>>> ch = 'abcde'
>>> ch[0] = 'a'
TypeError: 'str' object does not support item assignment
```

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'
```

En python, les listes et les chaînes de caractères commencent à l'indice 0.

Modification des listes

```
>>> liste = [8, "a", [4, 5]]
>>> print(liste[0])
8
>>> print(len(liste))
3
>>> print(liste[-1])
[4, 5]
>>> print(liste[2][1])
5
>>> liste.append("bcde")
>>> print(liste)
[8, 'a', [4, 5], 'bcde']
```

Modification des chaînes de caractères

```
>>> ch = "PTSI"
>>> print(ch[0:2])
PT
>>> ch = ch.replace('I', '')
>>> print(ch)
PT
```

2 Instructions et expressions

Définition

Expression : évaluation d'un calcul. Un résultat est retourné. (Exemple : 1+1)

Instruction : action utilisée dans un algorithme ou dans un programme. Une instruction peut inclure une expression. (Exemple : a = 1+1)

3 Structures algorithmiques

Définition

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 $n!$.

Exemple



```
n=4
res=0
if n==0:
    res = 1
else:
    i=1
    res=1
    while i<=n:
        res=res*i
        i+=1
```

```
n=4
res=0
if n==0:
    res = 1
else:
    res=1
    for i in range(0,n):
        res=res*(i+1)
```

4 Structure d'un programme

The image shows a code editor with two files: `main_rugosite.py` and `biblio_rugosite.py`. Annotations highlight the program structure:

- Programme principal (main)**: Points to the `main_rugosite.py` file.
- Import de bibliothèques de fonctions**: Points to the `import` statements in `main_rugosite.py`.
- Bibliothèque `biblio_rugosite`**: Points to the `biblio_rugosite.py` file.
- Signature de la fonction `generate_profil`**: Points to the `def generate_profil` line in `biblio_rugosite.py`.
- Appel d'une fonction de la bibliothèque `biblio_rugosite`**: Points to the `generate_profil` call in `main_rugosite.py`.
- Spécifications de la fonction `generate_profil`**: Points to the docstring in `biblio_rugosite.py`.