

**Université frères Mentouri
Faculté des sciences de la nature et de la vie
Département biochimie et biologie cellulaire et moléculaire**

Cours :

« *Introduction au langage Python* »

Spécialité: M1-Biochimie Appliquée

Dr. EUTAMENE AICHA

Année universitaire 2019/2020

Chapitre 3:

Les chaînes de caractères

Chapitre 3 : les chaînes de caractères

1. Définition

- Les chaînes de caractères sont des listes de caractères qui se suivent et chacun à sa position.

Exemple: chaine = "master biochimie"

Remarque 1:

Une chaîne de caractère peut être traitée comme un seul objet (une seule chaîne) ou un ensemble d'objets (divisée en caractères).

Exemple: Print chaine
Print chaine[0]

Chapitre 3 : les chaînes de caractères

2. La longueur d'une chaîne de caractères

La longueur d'une chaîne de caractère est égale au nombre de caractères qu'elle contient. En utilise la fonction **len(chaine de caractères)** pour afficher la taille d'une chaîne.

Exemple:

```
chaine = " Master"  
Print len(chaine) // affiche la valeur 6
```

Chapitre 3 : les chaînes de caractères

Remarque 2:

Il est possible d'obtenir une sous-chaîne à partir d'une chaîne initiale. Cela est fait en utilisant un intervalle représentant la position du caractère de début et de n de la sous chaîne.

Exemple:

```
chaine = "Bonjour"  
Print chaine [0:3] // affichera Bonj  
Print chaine [1:3] // affichera onj  
Print chaine [1:] // affichera onjour  
Print chaine [:3] // affichera Bonj  
Print chaine[:-1] // affichera Bonjour
```

Chapitre 3 : les chaînes de caractères

Remarque 3:

Il est possible de concaténer (raccorder) deux chaînes de caractères ensemble et cela a l'aide de l'opérateur +.

Exemple:

```
chaine1 = "Master"  
chaine2 = "biochimie"  
chaine3 = chaine1 + chaine2  
Print chaine3 //affichera master biochimie
```

Chapitre 3 : les chaînes de caractères

3. Parcourir une chaîne de caractères

On peut parcourir tous les caractères d'une chaîne à l'aide d'une boucle (**While** ou **For**).

Exemple 1 (while):

```
chaine = "master"  
While index < len(chaine):  
    Print chaine[index]  
    index = index + 1
```

Exemple 1 (For..in):

```
chaine = "master"  
For caractere in chaine:  
    Print caractere
```

Chapitre 3 : les chaînes de caractères

Exercices

- **Exercice 1:**

Ecrivez un script qui détermine si une chaîne contient ou non le caractère e.

- **Exercice 2:**

Ecrivez un script qui compte le nombre d'occurrences du caractère e dans une chaîne.

- **Exercice 3:**

Ecrivez un script qui retourne l'inverse d'une chaîne de caractère

- **Exercice 4:**

En partant de l'exercice 3, écrivez un script qui détermine si une chaîne de caractères donnée est un palindrome (c'est-à-dire une chaîne qui peut se lire indifféremment dans les deux sens), comme par exemple radar ou sos.

Chapitre 4:

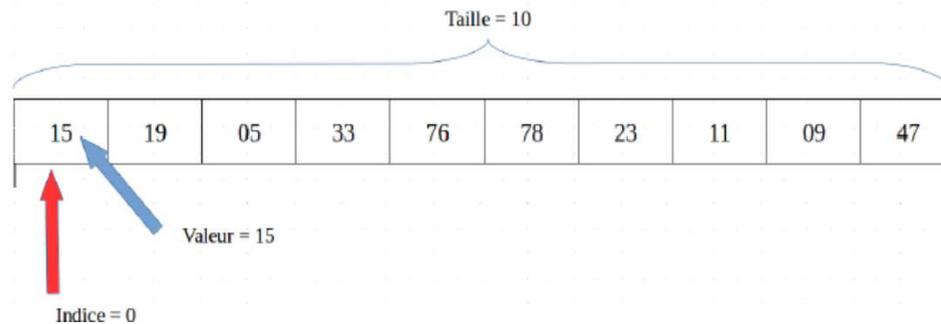
Les listes (tableaux)

Chapitre 4: Les listes à une dimension

1. Définition

- Une liste est une structure de données contenant un ensemble de valeurs homogènes ou hétérogènes **qui n'ont pas forcément le même type** et qui sont repérée par un nombre (indice).

Exemple:



Chapitre 4: Les listes à une dimension

2. Création d'une liste

Une liste est créée en affectant `[]` à une variable.

Exemple:

```
liste = []
```

3. Remplissage d'une liste

On remplit une liste soit lors de la création en mettant les valeurs entre `[]`, soit en utilisant la méthode **`append ()`**.

Exemple:

```
liste = [1, 2, 3] ma liste.append (1, 2, 3)
```

Chapitre 4: Les listes à une dimension

4. accès aux éléments d'une liste

On accède aux éléments d'une liste à l'aide de l'indice de la case, soit directement soit à l'aide d'une boucle.

Exemple:

```
print ma liste [2]  
ma liste [2] = 15  
for i in ma liste :  
    print i
```

5. fonctions utiles pour les listes

- **del** ma liste[i] : permet de supprimer l'élément à la case "i"
- ma liste.**remove("valeur")** : permet de supprimer l'élément dont la valeur est « valeur »
- ma liste.**reverse()** : permet d'inverser une liste.

Chapitre 4: Les listes à une dimension

5. fonctions utiles pour les listes(suite)

- **len**(ma liste) : renvoie la longueur de la liste.
- ma liste.**index("valeur")** : renvoie l'indice de la case contenant la valeur "valeur".

Chapitre 4: Les listes à une dimension

Exercices

- **Exercice 1:**

Ecrire un script Python qui déclare et remplit une liste de 7 valeurs numériques en les mettant toutes à zéro.

- **Exercice 2:**

Ecrire un script Python qui déclare une liste de 9 notes, dont on fait ensuite saisir les valeurs par l'utilisateur.

- **Exercice 3:**

Ecrire un script Python qui permet d'identifier le nombre le plus grand parmi N nombres stockés dans une liste.

Chapitre 4: Les listes à deux dimensions

1. Définition

Une liste à deux dimensions est une structure de données contenant un ensemble de valeurs homogènes ou hétérogènes qui n'ont pas forcément le même type, repérées par un nombre (indice) représenté par l'intersection d'une ligne et d'une colonne.

Exemple:

	0	1	2
0	0,0	0,1	0,2
1	1,0	1,1	1,2
2	2,0	2,1	2,2

Chapitre 4: Les listes à deux dimensions

2. Création des listes à deux dimensions

Il existe plusieurs façons pour créer une liste à deux dimensions sous Python, l'une d'elles consiste à créer une liste vide et y ajouter des listes vides avec **append()**. Nous pouvons construire n'importe quelle liste **rectangulaire (ou irrégulière)** de cette façon.

Exemple : création une liste à deux dimensions vide

```
ma liste 2D = []  
ma liste 2D.append([])  
ma liste 2D.append([])
```

Remarque 1:

Dans l'exemple précédent, la liste ma liste 2D va contenir deux lignes (listes) et un nombre (jusqu'a présent) inconnu de colonnes.

Chapitre 4: Les listes à deux dimensions

Exemple: de création d'une liste à 2 dimensions remplie

```
ma liste 2D = []  
ma liste 2D.append([1,2,3])  
ma liste 2D.append([4,5,6])
```

Remarque 2:

Dans l'exemple précédent, on obtient une matrice qui va contenir deux lignes (listes) et 3 colonnes on dit que c'est une matrice 2x3.

Chapitre 4: Les listes à deux dimensions

3. accès aux éléments d'une liste 2D

Pour accéder aux éléments d'une liste 2D, il suffit de préciser le numéro de ligne et le numéro de colonne de la case.

Exemple:

```
print ma liste 2D = [0][0]
>> 1
print ma liste 2D = [0][1]
>> 2
print ma liste 2D = [1][0]
>> 4
```

Chapitre 4: Les listes à deux dimensions

4. Accès aux éléments d'une liste 2D

On peut également accéder aux éléments d'une liste 2D en utilisant deux boucles imbriquées.

Exemple: accès aux éléments d'une liste 2D

```
for i in ma liste 2D:  
    . for j in i:  
        .. print j
```

Exemple: accès aux éléments d'une liste 2D

```
for i in range(len(l2)) :  
    . for j in range(len(l2[i])) :  
        .. print l2[i][j]
```

Chapitre 4: Les listes à deux dimensions

Exercices

- *Exercice 1:*

écrire un script python qui permet de parcourir une liste 2D contenant des nombres entiers et d'afficher les nombres supérieurs a 0.

- *Exercice 2:*

écrire un script python qui permet de parcourir une liste 2D (matrice 4x4) contenant des nombres entiers et d'afficher les nombres se situant a la diagonale de la matrice.