

Université Larbi Ben Mhidi

Faculté des Sciences de la Terre, de la Géographie et de l'Aménagement du Territoire

Département Géographie et aménagement du territoire

Matière : Programmation

**Niveau : 1ère année Licence Géographie et aménagement du territoire —
Semestre 2**

Crédits : 2 | Coefficient : 1

Enseignant : Bezzaz Soumia

Année universitaire : 2025-2026

LES LISTES : STRUCTURES DE DONNÉES ORDONNÉES

1. Qu'est-ce qu'une liste ?

Une **liste** en Python est une collection ordonnée et modifiable d'éléments. Contrairement aux chaînes de caractères, les éléments d'une liste peuvent être de types différents (nombres, textes, etc.). Les listes sont définies en plaçant tous les éléments entre crochets [], séparés par des virgules.

Exemple : Une liste de villes ou de coordonnées géographiques.

```
villem_algerie = ["Alger", "Oran", "Constantine", "Annaba"]  
coordonnees_point = [36.75, 3.05] # Latitude et Longitude  
d'Alger
```

2. Accéder aux éléments d'une liste

Chaque élément d'une liste possède un **index**, qui est sa position dans la liste. En Python, les index commencent à 0 pour le premier élément.

Syntaxe : `ma_liste[index]`

Exemple : Accéder à la première ville ou à la longitude.

```
villes_algerie = ["Alger", "Oran", "Constantine", "Annaba"]
print(villes_algerie[0]) # Affiche "Alger"
print(villes_algerie[2]) # Affiche "Constantine"
```

```
coordonnees_point = [36.75, 3.05]
print(coordonnees_point[1]) # Affiche 3.05 (La Longitude)
```

3. Modifier des éléments d'une liste

Les listes sont **mutables**, ce qui signifie que vous pouvez changer la valeur d'un élément après que la liste a été créée. Pour cela, on utilise l'index de l'élément à modifier.

Syntaxe : `ma_liste[index] = nouvelle_valeur`

Exemple : Corriger le nom d'une ville ou mettre à jour une coordonnée.

```
villes_algerie = ["Alger", "Oran", "Constantine", "Annaba"]
print("Avant modification :", villes_algerie)
villes_algerie[3] = "Sétif" # Remplace "Annaba" par "Sétif"
print("Après modification :", villes_algerie)
```

```
coordonnees_point = [36.75, 3.05]
coordonnees_point[0] = 36.77 # Nouvelle Latitude
print("Nouvelles coordonnées :", coordonnees_point)
```

4. Ajouter des éléments à une liste

Il existe plusieurs méthodes pour ajouter des éléments à une liste :

4.1. `append()` : Ajouter à la fin

La méthode `append()` ajoute un élément à la fin de la liste.

Syntaxe : `ma_liste.append(element)`

Exemple : Ajouter une nouvelle ville à notre liste.

```
villes_algerie = ["Alger", "Oran", "Constantine"]
print("Liste initiale :", villes_algerie)
villes_algerie.append("Tlemcen")
print("Après append() :", villes_algerie)
```

4.2. `insert()` : Insérer à une position spécifique

La méthode `insert()` permet d'ajouter un élément à un index précis. Les éléments existants à partir de cet index sont décalés vers la droite.

Syntaxe : `ma_liste.insert(index, element)`

Exemple : Insérer une ville au début ou au milieu de la liste.

```
villes_algerie = ["Oran", "Constantine"]
print("Liste initiale :", villes_algerie)
villes_algerie.insert(0, "Alger") # Insère "Alger" à l'index
0
print("Après insert(0, 'Alger') :", villes_algerie)
villes_algerie.insert(2, "Sétif") # Insère "Sétif" à l'index
2
print("Après insert(2, 'Sétif') :", villes_algerie)
```

5. Supprimer des éléments d'une liste

Plusieurs méthodes permettent de retirer des éléments d'une liste :

5.1. `remove()` : Supprimer par valeur

La méthode `remove()` supprime la **première occurrence** de la valeur spécifiée. Si la valeur n'est pas trouvée, une erreur `ValueError` est générée.

Syntaxe : `ma_liste.remove(valeur)`

Exemple : Retirer une ville de la liste.

```
villes_algerie = ["Alger", "Oran", "Constantine", "Oran"]
print("Liste initiale :", villes_algerie)
villes_algerie.remove("Oran") # Supprime la première
occurrence de "Oran"
print("Après remove('Oran') :", villes_algerie)
```

5.2. `clear()` : Vider toute la liste

La méthode `clear()` supprime tous les éléments de la liste, la rendant vide.

Syntaxe : `ma_liste.clear()`

Exemple : Vider une liste de relevés.

```
relevés_temperature = [25.5, 26.1, 24.9, 27.0]
print("Relevés initiaux :", relevés_temperature)
relevés_temperature.clear()
print("Après clear() :", relevés_temperature)
```