



Chapitre 5

Structures de données

Les Listes et les Tuples

Pour les étudiants en génie civil – M1 RIB

Dr. Hachimi Dahhaoui
Université de Tlemcen

Mercredi 26 octobre 2025



Diapo 2 – Pourquoi les listes ?

Problème à poser :

python

 Copier le code

```
r1 = 15.2
```

```
r2 = 16.5
```

```
r3 = 14.8
```

```
r4 = 17.0
```

```
r5 = 15.6
```

💬 *“Et si on avait 100 valeurs ?”*

→ *“On a besoin d’une seule variable pour tout ranger : une liste.”*



Diapo 3 – Définition

↪ *Une liste est un ensemble ordonné et modifiable d'éléments.*

Exemples :

```
python
```

 Copier le code

```
courses = ["pain", "lait", "oeufs"]  
chantier = ["béton", "acier", "bois"]
```

💬 *“C’est comme une boîte à compartiments où chaque objet a une place.”*





Diapo 4 – Schéma visuel

```
courses = ["pain", "lait", "oeufs"]
```

Dessin d'une liste : [0] pain | [1] lait | [2] œufs

💬 “Chaque élément a un numéro d'ordre, qu'on appelle indice.”



Diapo 5 – Créer une liste

python

 Copier le code

```
ma_liste = [10, 20, 30]
```

```
liste_vide = []
```

 *“Une liste peut contenir des nombres, du texte ou les deux.”*

 *Les indices commencent toujours à zéro.*



Diapo 6 – Accéder à un élément

python

 Copier le code

```
courses = ["pain", "lait", "oeufs"]  
print(courses[0]) # Premier élément  
print(courses[-1]) # Dernier élément
```

💬 “Les indices positifs partent du début, les négatifs partent de la fin.”



Diapo 7 – Modifier une valeur

python

 Copier le code

```
courses[1] = "café"  
print(courses)
```

💬 *“On peut changer une valeur à tout moment.”*

```
courses = ["pain", "lait", "oeufs"]
```

```
['pain', 'café', 'oeufs']
```





Diapo 8 – Ajouter et supprimer

python

 Copier le code

```
courses.append("sucre")    # Ajouter à la fin
courses.insert(1, "beurre") # Insérer à une position
courses.remove("pain")     # Supprimer
```

💬 “append() ajoute toujours à la fin, remove() enlève la première occurrence.”

```
['pain', 'café', 'oeufs', 'sucre']
```

```
['pain', 'beurre', 'café', 'oeufs', 'sucre']
```

```
['beurre', 'café', 'oeufs', 'sucre']
```




Diapo 9 – Supprimer avec pop() et del

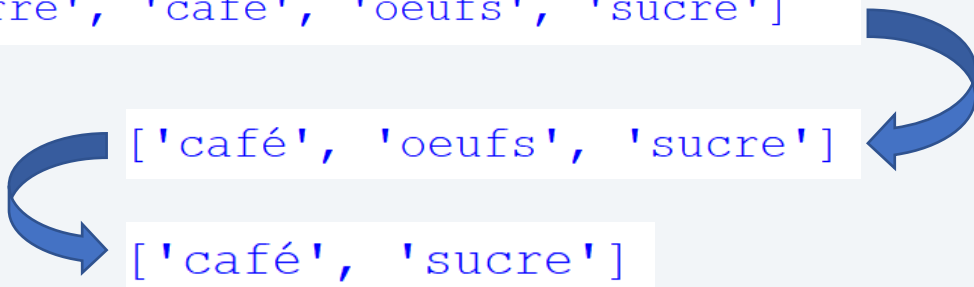
python

 Copier le code

```
courses.pop(0)  
del courses[1]
```

💬 “pop() enlève et renvoie l’élément supprimé, del supprime sans retour.”

['beurre', 'café', 'oeufs', 'sucre']



Diapo 10 – Exemple concret

python

 Copier le code

```
resistances = [15.2, 16.5, 14.8, 17.0, 15.6]  
moy = sum(resistances) / len(resistances)  
print("Moyenne :", moy)
```

💬 *“On peut faire des calculs directement sur une liste.”*

Moyenne : 15.819999999999999





Diapo 11 – Fonctions utiles

Fonction	Rôle
len(liste)	nombre d'éléments
sum(liste)	somme
max(liste) / min(liste)	valeurs extrêmes
sort() / reverse()	tri



Diapo 12 – Application en génie civil

python

 Copier le code

```
resistances.sort()  
print("Triées :", resistances)
```

```
resistances = [15.2, 16.5, 14.8, 17.0, 15.6]  
resistances.sort()  
print("Triées :", resistances)
```

```
Triées : [14.8, 15.2, 15.6, 16.5, 17.0]
```



💬 “On calcule la résistance moyenne d’un béton, puis on trie les valeurs du plus faible au plus fort.”



Diapo 13 – Erreurs fréquentes

- ❑ IndexError : mauvais indice
 - ❑ ValueError : supprimer une valeur absente
 - ❑ TypeError : addition entre types différents
- 💬 “Python nous aide à comprendre nos erreurs.”

Diapo 14 – Exercice 1 (interactif)

- ☐ Crée une liste de 4 nombres,
- ☐ Ajoute un élément, supprime le premier,
- ☐ et affiche la moyenne.

Diapo 14 – Solution exercice 1

python

 Copier le code

Création d'une Liste de 4 nombres

```
valeurs = [10, 20, 30, 40]
```

Ajouter un élément

```
valeurs.append(50)
```

Supprimer Le premier élément

```
valeurs.pop(0)      # ou del valeurs[0]
```

Calculer La moyenne

```
moyenne = sum(valeurs) / len(valeurs)
```

Affichage du résultat

```
print("La liste finale :", valeurs)
```

```
print("La moyenne des valeurs est :", moyenne)
```

Diapo 14 – Résultat exercice 1

yaml

 Copier le code

La liste finale : [20, 30, 40, 50]

La moyenne des valeurs est : 35.0

python

 Copier le code

```
if moyenne > 30:
```

```
    print("Bonne moyenne ✅")
```

```
else:
```

```
    print("Moyenne faible ⚠️")
```


Diapo 15 – Piles et files

Pile (LIFO) :

python

 Copier le code

```
pile = [1, 2, 3]
pile.append(4)
pile.pop()
```

File (FIFO) :

python

 Copier le code

```
file = ["Ali", "Mina", "Youssef"]
file.append("Nadia")
file.pop(0)
```

 “Les camions bétonnières se chargent et déchargent selon une file (FIFO).”



Diapo 16 – Boucle sur une liste

python

 Copier le code

```
for item in ["pain", "lait", "oeufs"]:  
    print("J'achète :", item)
```

```
J'achète : pain  
J'achète : lait  
J'achète : oeufs
```



💬 *“for” permet de parcourir tous les éléments un par un.”*



Diapo 18 – Exercice 2

Crée une liste de matériaux et affiche chacun avec la phrase :

“Matériau utilisé : ...”



Diapo 18 – Solution Exercice 2

python

 Copier le code

```
# Création d'une Liste de matériaux
matériaux = ["béton", "acier", "bois", "verre", "terre cuite"]

# Parcourir La Liste avec une boucle for
for mat in matériaux:
    print("Matériau utilisé :", mat)
```

🔄 Résultat attendu :

less

 Copier le code

```
Matériau utilisé : béton
Matériau utilisé : acier
Matériau utilisé : bois
Matériau utilisé : verre
Matériau utilisé : terre cuite
```



Diapo 18 – Listes en compréhension

python

 Copier le code

```
valeurs = [10, 20, 25, 15]
filtrées = [v for v in valeurs if v > 15]
print(filtrées)
```

💬 *“Permet de créer une nouvelle liste à partir d’une autre.”*

[20, 25]





Diapo 19 – Exemple génie civil

Filtrer les briques conformes (résistance > 15 MPa)

python

 Copier le code

```
resistances = [10, 18, 22, 13]
conformes = [r for r in resistances if r >= 15]
print(conformes)
```

[18, 22]



Diapo 21 – QCM rapide

- 1 Que fait `append()` ?
- 2 Quelle fonction renvoie le nombre d'éléments ?
- 3 Que fait `[x*2 for x in range(3)]` ?

Diapo 22 – Les tuples (Introduction)

python

 Copier le code

```
dimensions_brique = (0.29, 0.14, 0.09)
```

💬 “Un tuple, c’est comme une liste, mais figée : on peut la lire, pas la modifier.”



Diapo 23 – Pourquoi un tuple ?

💬 *“Les dimensions d’une brique normalisée ou d’une poutre ne changent pas.”*



Diapo 23 – Accéder aux éléments

python

 Copier le code

```
print(dimensions_brique[0])
```

💬 *“Même principe que la liste, mais sans modification possible.”*

```
dimensions_brique = (0.29, 0.14, 0.09)  
print(dimensions_brique[0])
```

0.29





Diapo 24 – Déballer un tuple

💬 *“Déballer, c’est attribuer chaque valeur à une variable.”*

python

 Copier le code

```
L, l, h = dimensions_brique  
volume = L * l * h  
print("Volume :", volume)
```

```
dimensions_brique = (0.29, 0.14, 0.09)  
L, l, h = dimensions_brique  
volume = L * l * h  
print("Volume :", volume)
```

Volume : 0.003654





Diapo 26 – Comparaison liste / tuple

Caractéristique	Liste	Tuple
Modifiable	☒ Oui	☒ Non
Syntaxe	[]	()
Exemple	[1, 2, 3]	(1, 2, 3)
Utilisation	Données variables	Données fixes



Diapo 26 – Exercice 3

Crée un tuple $(4, 3, 0.25)$ représentant un mur (L, H, e) .

Calcule le volume, puis tente de changer $e \rightarrow$ erreur.



Diapo 26 – Solution exercice 3

python

 Copier le code

```
# Création du tuple représentant Le mur
mur = (4, 3, 0.25)  # (Longueur, Hauteur, Épaisseur)

# Déballage du tuple
L, H, e = mur

# Calcul du volume
volume = L * H * e
print("Le volume du mur est :", volume, "m³")

# Tentative de modification (provoque une erreur)
mur[2] = 0.30
```



Diapo 26 – Résultat exercice 3

vbnet

 Copier le code

Le volume du mur est : 3.0 m³

Traceback (most recent call last):

...

TypeError: 'tuple' object does not support item assignment

Diapo 27 – Résumé général

- ☐ Liste = modifiable
- ☐ Tuple = non modifiable
- ☐ Tri, filtrage, calcul, parcours
- ☐ Base de toute manipulation de données

Merci !
Des questions ?

Contact : hachimi.dahhaoui@univ-tlemcen.dz

Fin du Chapitre 5

À très bientôt pour la suite !