PYTHON — LES LISTES

REGROUPER PLUSIEURS VALEURS

1. À quoi ça sert ?

Une liste stocke plusieurs valeurs dans une variable :
des prénoms, des notes, des températures…

Les éléments sont ordonnés et modifiables.

2. Créer une liste

```
fruits = ["pomme", "banane", "kiwi"]
notes = [12, 15, 9]
```

- Les crochets [] délimitent la liste.
- Les virgules séparent les éléments.
- Les textes sont placés entre guillemets.

Liste vide : `fruits = []`

3. Accéder à un élément

Chaque position est appelée indice. On commence à 0.

"pomme"	"banane"	"kiwi"
0	**1**	**2**

```
print(fruits[0])     # affiche pomme
print(fruits[2])     # affiche kiwi
print(fruits[-1])    # dernier élément : kiwi
```

4. Modifier un élément

```
fruits[1] = "poire"
print(fruits)
```

Résultat : ['pomme', 'poire', 'kiwi']

Seul l'élément à l'indice 1 est remplacé.

PYTHON — LES LISTES

AJOUTER, SUPPRIMER ET PARCOURIR

5. Ajouter ou supprimer un élément

Ces exemples sont indépendants.

Ajouter à la fin avec append() :

```
fruits = ["pomme", "banane", "kiwi"]
fruits.append("orange")
```

→ `['pomme', 'banane', 'kiwi', 'orange']`

Supprimer une valeur avec remove() :

```
fruits = ["pomme", "banane", "kiwi"]
fruits.remove("banane")
```

→ `['pomme', 'kiwi']`

remove() supprime la première occurrence.
Si la valeur est absente, Python signale une erreur.

6. Compter les éléments avec len()

```
fruits = ["pomme", "banane", "kiwi"]
print(len(fruits))   # affiche 3
```

La longueur est 3, mais le dernier indice est 2.

7. Parcourir une liste avec for

```
for fruit in fruits:
    print(fruit)
```

Affiche pomme, banane, puis kiwi sur trois lignes.
À chaque tour, fruit prend la valeur de l'élément suivant.

PYTHON — LES LISTES

TESTER, CALCULER ET MÉMORISER

8. Vérifier la présence avec in

```
fruits = ["pomme", "banane", "kiwi"]
if "kiwi" in fruits:
    print("Il reste du kiwi !")
else:
    print("Il n’y a plus de kiwi.")
```

9. Exemple : calculer une moyenne

```
notes = [12, 15, 9]
total = 0
for note in notes:
    total = total + note
moyenne = total / len(notes)
print("Moyenne :", moyenne)
```

Résultat : Moyenne : 12.0

On additionne les notes, puis on divise par leur nombre.
Cet exemple suppose que la liste n’est pas vide.

Les erreurs à éviter

- 3 éléments : indices 0, 1, 2. L’indice 3 est hors limites.
- Écrire liste.append(40), pas append(40) seul.
- Avant remove(), vérifier la présence avec in.
- Liste non vide : dernier indice = len(liste) - 1.

À RETENIR

Créer	`liste = [10, 20, 30]`
Lire / modifier	`liste[0] / liste[0] = 15`
Ajouter / supprimer	`liste.append(40) / liste.remove(20)`
Compter	`len(liste)`
Parcourir	`for element in liste:`
Tester une présence	`20 in liste`