

PYTHON : FICHE RÉCAP

LES LISTES

Initialiser une liste

```
noms = ["Jean", "Sophie", "Martin"] # liste avec 3 chaines de caractères (crochets)
noms2 = ("Jean", "Sophie", "Martin") # tuple avec 3 chaines de caractères (parenthèses)
```

La liste permet de modifier les éléments et de rajouter/supprimer des éléments. Le tuple est en lecture seule.

On peut utiliser tous types de valeurs :

```
ages = [20, 25, 15]
```

Nombre d'éléments

```
noms = ["Jean", "Sophie", "Martin"]
nombre_elements = len(noms)          # On récupère la valeur 3
```

Indexer une liste

```
noms = ["Jean", "Sophie", "Martin"]
print(noms[0])                        # Affiche "Jean"
```

Les indexes commencent toujours à 0

```
for i in range(len(noms)):
    print(noms[i])
```

Enumérer une liste

On peut utiliser la boucle for pour récupérer directement les éléments d'une collection :

```
for nom in noms:
    print(nom)
```

Ajouter / Supprimer des éléments

```
noms.append("Toto")          # Rajoute l'élément "Toto" à la fin
noms.extend("Toto")          # Rajoute l'élément "Toto" à la fin (même chose)

noms.append(["Toto", "Titi"]) # Rajoute l'élément ["Toto", "Titi"] à la fin
noms.extend(["Toto", "Titi"]) # Rajoute deux éléments à la fin, d'abord "Toto", puis "Titi"

noms += ["Toto", "Titi"]     # Se comporte comme "Extend"

del noms[0]                  # Supprime le premier élément de la liste
```

Slices

```
noms = ["Jean", "Sophie", "Martin", "Christophe", "Zoe"]
```

Les slices d'utilisent avec un index de début (inclus) et de fin (exclus)

```
noms[1:]      # Le premier élément n'est pas pris : ["Sophie", "Martin", "Christophe", "Zoe"]
noms[:-1]     # Le dernier élément n'est pas pris : ["Jean", "Sophie", "Martin", "Christophe"]
noms[:2]      # Les deux premiers éléments sont pris : ["Jean", "Sophie"]
noms[1:2]     # On démarre à l'index 1 puis on s'arrête à l'index 2 (non compris) : ["Sophie"]
```

Tris

```
noms.sort()          # Trie par ordre alphabétique
ages.sort()          # Trie du plus petit au plus grand
```

```
noms.sort(reverse=True) # Inverse l'ordre
```

key permet de donner une fonction pour faire un tri personnalisé. Ici on trie par le nombre de caractères

```
noms.sort(key=lambda e: len(e))
```

La fonction sorted crée une nouvelle liste triée en sortie, au lieu d'altérer la collection noms (au contraire de 'sort' qui est "inplace")

```
noms_tries = sorted(noms)
```

Inverser deux éléments (swapper)

```
noms[0], noms[-1] = noms[-1], noms[0] # Ici on inverse le premier et le dernier élément
```

Join

Permet de concaténer les éléments d'une liste pour créer une chaîne de caractères

```
les_noms = "-".join(noms)    # on obtient : "Jean-Sophie-Martin-Christophe-Zoe"
```

Compréhensions de listes

Permet de construire une liste "en ligne". Ici on construit une liste qui contient le nombre de caractères de chaque nom.

```
len_noms = [len(nom) for nom in noms]
```

Il est possible de décider si on ajoute un élément ou non

```
len_noms = [len(nom) for nom in noms if len(nom) > 3]
```

Ou bien on peut conditionner la valeur

```
len_noms = [len(nom) if len(nom) > 3 else 0 for nom in noms]
```