

⚠️ *Python fait une distinction entre les objets muables et immuables.*

Les objets muables

Les objets **muables** sont **modifiés en mémoire** à chaque fonction qu'on leur applique. On dit qu'il sont passés par **référence**.

Exemples d'objets muables :

- listes
- dictionnaires
- sets

Les objets immuables

Les objets immuables ne sont pas modifiés, et **les fonctions qu'on leur applique retourne une copie modifiée de l'objet tandis que l'objet de base reste inchangé**. On dit qu'ils sont passés par **valeur**. Le résultat de l'opération va devoir être affecté à une variable pour être utilisé.

Exemples d'objets immuables :

- chaîne de caractères
- nombres
- tuples

Exemples

```
string_immuable = "coucou ca va ?"
string_immuable.upper()

print(string_immuable) # La string demeure inchangée car elle est immuable

liste_muable = [1, 4, 3, 2]
liste_muable.sort()

print(liste_muable) # La liste sera triée en mémoire et changée pour toujours car
elle est muable
```

Pour corriger notre `.upper()`, il va nous falloir stocké le résultat dans une variable qui peut-être elle même.

```
string_immuable = "coucou ca va ?"  
nouvelle_string = string_immuable.upper()  
string_immuable = string_immuable.upper()  
  
print(nouvelle_string, string_immuable) # La string à enfin changé
```