

➤ Indicage

`len()` : Donne la longueur d'une chaîne de caractères, d'une liste ou d'un tuple.
`...[n]` : Renvoie le caractère d'une chaîne ou le terme d'une liste (ou tuple) d'indice n .
`...[n : m]` : Renvoie les caractères ou termes situés entre les indices n et $m - 1$. (c'est-à-dire le $(n + 1)^{\text{ème}}$ et le $m^{\text{ème}}$ termes)

Attention : Les indices commencent à 0.

Remarque : On peut aussi prendre des indices négatifs, le dernier est -1 .

Astuce : `[: m]` commence au début et `[n :]` va jusqu'à la fin.

Exemples

```
mot = "bonjour"
len(mot)      renvoie 7
mot[3]        renvoie "j"
mot[-1]       renvoie "r"
mot[3:6]      renvoie "jou"

voyelles = ("a","e","i","o","u","y")
len(voyelles) renvoie 6
voyelles[3]   renvoie "o"
voyelles[-2]  renvoie "u"
voyelles[1:3] renvoie ("e","i")
```

➤ Opérations

`ch1 + ch2` : Concaténation : permet de mettre bout à bout deux chaînes (listes ou tuples).
`chaîne * n` : Répétition : permet de répéter n fois une même chaîne (liste ou tuple).
`if c in chaîne` : L'instruction `in` permet de tester si un caractère ou une chaîne de caractères sont présents dans une chaîne de caractères (un élément ou une sous-liste dans une liste, tuple...)
`for c in chaîne` : L'instruction `in` permet aussi de parcourir une chaîne (liste ou tuple).
`del liste[n]` : `del` permet de supprimer un élément d'une liste (seulement) par la donnée de son indice.

➤ Conversions

`chaîne.split(" ")` Permet de convertir une chaîne de caractères en une liste en coupant à chaque espace.
`" ".join(liste)` Permet de convertir une liste en un chaîne de caractères en intercalant un espace entre chaque mot de la liste. (On peut remplacer " " par n'importe quel mot)

Remarques : On peut convertir une chaîne en nombre entier (*int*) ou en nombre décimal (*float*), ou au contraire convertir un nombre en chaîne de caractères (*str*).
On peut aussi récupérer le code ASCII (*ord*) ou au contraire obtenir le caractère (*chr*).
On peut même évaluer la valeur d'une chaîne de caractères si cela à un sens (*eval*).

➤ Méthodes

- Quelques méthodes **ne modifiant pas l'objet** sur lequel elles agissent :

`chaîne.lower()` Permet de mettre toute la chaîne en minuscules.
`chaîne.upper()` Permet de mettre toute la chaîne en majuscules.
`chaîne.find(mot)` Permet de trouver l'indice où se trouve un mot dans une chaîne.
`chaîne.count(mot)` Permet de compter combien de fois apparaît un mot dans une chaîne.
`chaîne.replace(mot1, mot2)` Permet de remplacer le mot1 par le mot2 partout dans la chaîne.
`liste.count(mot)` Permet de compter combien de fois apparaît un mot dans une liste.
`liste.index(mot)` Permet de trouver le premier indice où apparaît un mot dans une liste.

- Quelques méthodes **modifiant l'objet** sur lequel elles agissent :

`liste.reverse()` Permet d'inverser l'ordre d'une liste.
`liste.sort()` Permet de trier une liste dans l'ordre croissant.
`liste.append(mot)` Permet d'ajouter un mot en fin de liste.
`liste.remove(mot)` Permet de supprimer la première occurrence d'un mot dans une liste.