



PYTHON : FICHE RÉCAP

LES CHAINES

Initialisation

Il est possible de définir une chaîne avec des guillemets simples ou des guillemets doubles :

```
a = 'Toto'
a = "Toto"
```

Concaténation

Pour concaténer des chaînes (les coller entre elles), on utilise le '+' :

```
a = "Je m'appelle " + "Tota"
```

Ici 'a' contient "Je m'appelle Tota"

La fonction print vous permet de passer plusieurs paramètres qui seront automatiquement concaténés en rajoutant un espace :

```
print("Je m'appelle " + "toto" + str(50)) # "Je m'appelle toto50"
print("Je m'appelle", 50) # "Je m'appelle 50"
```

Dans le 1er exemple, on passe un seul paramètre à la fonction print (et on concatène manuellement avec le +)

Dans le 2ème exemple, on passe deux paramètres à la fonction print (et c'est la fonction print qui va les concaténer "en interne")

Longueur d'une chaîne

Pour connaître le nombre de caractères dans une chaîne, on utilise "len"

```
a = "Toto"
print(len(a)) # Affiche 4
```

Index et slices

```
a = "Je m'appelle Tota"
```

```
print(a[0])          # Affiche le premier caractère : J
print(a[-1])         # Affiche le dernier caractère : a

print(a[3:])          # Affiche : m'appelle Tota
print(a[:-1])         # Affiche : Je m'appelle Tot
print(a[:2])          # Affiche : Je
```

Split

Split permet de découper un chaîne en plusieurs éléments dans une liste. On donne à split le critère pour découper, c'est à dire quel caractère va être détecté pour faire un découpage.

```
a = "Je m'appelle Tota"
split_a = a.split(" ")      # Ici on découpe la phrase par les caractères espaces

print(split_a)              # Affiche : ['Je', 'm'appelle', 'Tota']
```

Join

Join est "l'opération inverse du split". A partir d'une liste, on construit une chaîne concaténée avec le caractère "séparateur" qu'on fournit :

```
noms = ["Jean", "Paul", "Marc"]
print("-".join(noms))      # Affiche : Jean-Paul-Marc
```