

## Cours 2 - Les chaînes de caractères en python

### 1. Présentation des chaînes de caractères

Avec les nombres et les listes, les chaînes de caractères sont les éléments informatiques qui sont les plus utilisés. Ce sont des objets qui représentent du texte. Elles sont représentées entre guillemets ou entre apostrophes.

#### Exemple

```
a = "Bonjour tout le monde"
```

```
b = 'Bonsoir tout le monde'
```

### 2. Indexation d'une chaîne de caractères

Les chaînes de caractères sont indexés de la même manière que les listes.

ch[n] est le caractère à l'indice n de la chaîne ch

On peut utiliser l'indexation positive ou l'indexation négative.

#### 2-1. Indexation positive

##### Exemple

```
ch = 'bonjour'
```

'b'	'o'	'n'	'j'	'o'	'u'	'r'
0	1	2	3	4	5	6

ch[0] est le caractère 'b'

ch[1] est le caractère 'o'

...

ch[6] est le caractère 'r'

#### 2-1. Indexation négative

##### Exemple

```
ch = 'bonjour'
```

'b'	'o'	'n'	'j'	'o'	'u'	'r'
-7	-6	-5	-4	-3	-2	-1

ch[-1] est le caractère 'r'

ch[-2] est le caractère 'u'

...

ch[-7] est le caractère 'b'

### 3. Tranchage (ou slicing)

Le tranchage est l'extraction d'un groupe de caractères consécutifs d'une chaîne.

Le tranchage avec une chaîne de caractères fonctionne comme dans le cas d'une liste.

Syntaxe :

```
ch[a : b]
```

donne les caractères de ch compris entre les indices a et b-1.

Si on omet l'indice de début, la plage d'indices commence au début de la chaîne.

```
ch[ : b ]
```

donne les caractères compris entre l'indice de début de chaîne ch et l'indice b-1

Si on omet l'indice de fin, la plage d'indices se termine à la fin de la chaîne

```
ch[a : ]
```

donne les caractères compris entre l'indice a et la fin de la chaîne ch

#### Exemple

```
print( ch[3:6] )
```

```
# affiche jou
```

```
print( ch[:6] )
```

```
# affiche bonjou
```

```
print( ch[3:] )
```

```
# affiche jour
```

### 4. Opérations sur les chaînes

#### 4-1. Addition de deux ou de plusieurs chaînes

L'opération + permet la concaténation de chaînes de caractères.

ch1+ch2+ ... + chN concatène les chaînes ch1, ch2, ..., et chN en les mettant bout à bout.

#### Exemple

```
ch1="Bonjour"
```

```
ch2=" Tout le monde"
```

```
print(ch1+ch2) # affiche Bonjour Tout le monde
```

## 4-2. Multiplication d'une chaîne par un entier

L'opération `*` multiplie une chaîne autant de fois qu'on veut

### Exemple

```
ch = 'bonjour' * 3
print(ch)      # affiche bonjourbonjourbonjour
```

## 4-3. Opération in

L'opération `in` peut être utilisée pour tester si une chaîne de caractères est une sous-chaîne d'une autre chaîne de caractères.

`ch1 in ch2` donne `True` si `ch1` est une sous-chaîne de la chaîne `ch2` et `False` sinon.

### Exemple

```
ch = "C'est un trou de verdure où chante une rivière"
print("chante" in ch)      # affiche True
print("dur" in ch)         # affiche True
print("eau" in ch)         # affiche False
```

On peut utiliser l'opération `in` comme condition de la structure de contrôle `if`.

### Exemple

```
lettre = input("Tapez une lettre")
if lettre in "aeiouy":
    print(lettre + " est une voyelle")
else :
    print(lettre + " n'est pas une voyelle")
```

On peut aussi utiliser l'opération `in` avec une boucle `for` pour parcourir une chaîne.

### Exemple

```
ch="Quatre bonbons à 0,25 euro chacun coûtent un euro"
print('La chaîne ch est:')
print(ch)
print('Les chiffres qui sont dans la chaine ch sont:')
for car in ch:
    if car in "0123456789":
        print(car)
```

## 5. Quelques méthodes sur les chaînes de caractères

### 5-1. La méthode find

La méthode find donne l'indice de début, dans la chaîne, d'une sous-chaîne donnée en paramètre.

ch.find(sous\_chaine) : Donne l'indice de la première occurrence de la sous-chaîne dans ch ou -1 si la sous-chaîne ne se trouve pas dans ch.

#### Exemple

```
ch="Les enfants jouent dans la cour de la maison"
print(ch.find("jouent")) # donne l'indice de "jouent" dans la chaîne ch.
print(ch.find("la")) # donne l'indice du premier "la" dans la chaîne ch.
```

### 5-2. La méthode count

La méthode count donne le nombre d'occurrences (le nombre d'apparitions), dans la chaîne, d'une sous-chaîne donnée en paramètre.

ch.count(ch1) Donne le nombre de fois que ch1 se trouve dans la chaîne ch.

#### Exemple

```
ch="Les enfants jouent dans la cour de la maison"
print('La chaîne ch est: ', ch)
print('Le nombre de la dans la chaîne ch est: ', ch.count("la"))
print('Le nombre de s dans la chaîne ch est: ', ch.count("s"))
```

### 5-3. La méthode replace

La méthode replace remplace, dans la chaîne, toutes les occurrences d'une sous-chaîne donnée en paramètre par une autre sous-chaîne également donnée en paramètre.

Syntaxe

```
ch.replace(ch1, ch2)
# remplace dans ch toutes occurrences de ch1 par ch2
```

## Exemple

```
ch = "Les enfants jjjjouent dans lllla cour de lllla maison"
print('La chaîne ch est:')
print(ch)
ch = ch.replace("jjjj","j")
print('Après remplacement de jjjj par j, la chaîne ch est:')
print(ch)
ch = ch.replace("llll","l")
print('Après remplacement de llll par l, la chaîne ch est:')
print(ch)
```