

Partie II : Algorithmique et Programmation

SOMMAIRE

Chapitre 5 : Les chaînes de caractères

I. Définition :.....	2
II. Initialisation :.....	2
III. Accès aux valeurs d'une chaîne de caractères	3
IV. Chaînes de caractères sont immuables (immutable).....	3
V. Caractères spéciaux	3
VI. Les opérateurs de chaîne de caractères :.....	4
VII. Fonctions utiles sur les chaînes de caractères :.....	4
VIII. Méthodes utiles sur les chaînes de caractères :.....	5
IX. Les tranches :.....	6
X. Opérateur de formatage des chaînes de caractères (%)	6

Les chaînes de caractères :

I. Définition :

Les chaînes de caractères sont des variables contenant du texte. On appelle ça "chaîne de caractères" car ce sont plusieurs caractères qui se suivent.

Dans une chaîne de caractères on peut trouver une succession de lettres, des nombres, des chiffres, ...

Exemples de chaînes de caractères :

Bonjour	J'ai 21 ans	Programmation !
---------	-------------	-----------------

Le type chaînes de caractères est très utilisé en informatique, il permet dans plusieurs cas de simplifier la résolution des problèmes difficile à résoudre de manière numérique.

II. Initialisation :

On peut les créer simplement les variables de type chaînes de caractères par placement des caractères entre deux guillemets.

Exemples :

```
nom = "Mohamed"  
adresse = "Rue 2 Mars Casablanca"
```

Python traite les **guillemets** (double quotes) et les **primes** (les simples quotes) et même un **triplé de primes ou de guillemets**, de la même façon.

Exemples :

```
ch1 = "Bonjour Tout le monde ! "  
# création par double quotes  
ch2 = 'Bonjour tout le monde !'  
# création par simple quotes  
ch3 = '''Bonjour tout le monde ! '''  
# création par un triple de primes  
ch4 = """"Bonjour tout! """"  
# création par un triple de guillemets
```

Rq : Avec les triples quotes, on peut étaler la chaîne de caractères sur plusieurs lignes.

Exemple :

```
Ch = """"  
    Name : Youssef  
    Age : 32 ans  
""""
```

III. Accès aux valeurs d'une chaîne de caractères

Etant un cas particulier des listes, nous allons réutiliser les crochets et index pour manipuler les chaînes de caractères.

Voici un exemple de chaîne de caractères ainsi que l'affichage de la 1ère lettre :

```
chaîne = "HELLO"  
print ( chaîne[0] )
```

Vous pouvez donc voir que tout comme les listes, le premier caractère a pour index 0, le deuxième a pour index 1, etc...

H	E	L	L	O
0	1	2	3	4
-5	-4	-3	-2	-1

N.B : Les indices négatifs sont utilisés pour accéder aux valeurs d'une chaîne de caractères à partir de la fin. L'indexation dans ce cas commence à **-1**.

IV. Chaînes de caractères sont immuables (immutable)

En python, les chaînes de caractères ne peuvent pas subir des modifications de la même manière que les liste. Ainsi l'instruction suivante est non permise :

```
A= "Hello"  
A[0] = "P" # instruction non permise.
```

Mais, on peut créer des variations d'une chaîne de caractères en créant de nouvelles chaînes de caractères.

```
A= "Hello"  
A = "P" + A[1 :] => Pello
```

V. Caractères spéciaux

Pour afficher certains caractères spéciaux à l'intérieur d'une chaîne de caractère, on peut se servir du caractère (\) appelé back-slash.

Notation back-slash	Description
\a	Bell or alert
\b	Retour arriere (backspace)
\n	Retour à la ligne (newline)
\r	Retour chariot (carriage return)
\t	Tabulation (Tab)
\s	Espace (spase)

VI. Les opérateurs de chaîne de caractères :

Voici quelques opérateurs qui pourraient vous être utiles :

Opérateur :	Description :
+	Ajoute une chaîne à la suite d'une autre chaîne1 = "Bonjour" chaîne2 = " Youssef" chaîne = chaîne1 + chaîne2 print(chaîne) On obtient : Bonjour Youssef
*	Multiplie la chaîne chaîne1 = "bout" chaîne = chaîne1*3 print(chaîne) On obtient : boutboutbout

VII. Fonctions utiles sur les chaînes de caractères :

Voici quelques fonctions :

Fonction :	Description :
len(...)	Retourne la taille de la chaîne chaîne1 = "bout1" taille = len(chaîne1) print(taille) On obtient : 5
str(...)	Transforme un chiffre en texte chaîne = str(6) print(chaîne) On obtient : le caractère '6'
eval(...)	Evaluer une chaîne chiffrée ou une expression val = eval('6') print(val) x=5 res = eval("x+2") print(res) On obtient : 6 7

VIII. Méthodes utiles sur les chaînes de caractères :

Voici quelques méthodes :

Méthode :	Description :
chaîne.upper()	<p>Retourne la chaîne en majuscule.</p> <pre>chaîne = "bout1" majuscule = chaîne.upper() print(majuscule)</pre> <p>On obtient : BOUT1</p>
chaîne.lower()	<p>Retourne la chaîne en minuscule.</p> <pre>chaîne = "BOUT1" minuscule = chaîne.lower() print(minuscule)</pre> <p>On obtient : bout1</p>
chaîne.find("...")	<p>Retourne l'index du caractère s'il est présent dans la chaîne.</p> <pre>chaîne = "bout1" index = chaîne.find("u") print(index)</pre> <p>On obtient :</p> <p>Si le caractère n'est pas présent, on obtient -1.</p>
chaîne.index("... ")	<p>Retourne l'index du caractère s'il est présent dans la chaîne.</p> <pre>chaîne = "bout1" index = chaîne.index("u") print(index)</pre> <p>On obtient : 2.</p> <p>Si le caractère n'est pas présent, on a une erreur.</p>
chaîne.split("...")	<p>Retourne la chaîne sous forme de liste en fonction du séparateur.</p> <pre>chaîne = "Salut ca va ?" liste = chaîne.split(" ") print(liste)</pre> <p>Comme ici le séparateur est l'espace, on obtient : ['Salut', 'ca', 'va', '?'].</p>
"...".join(...)	<p>Retourne une liste sous forme de chaîne en fonction du séparateur.</p> <pre>liste = ["Salut", "ca", "va", "?"] chaîne = " ".join(liste) print(chaîne)</pre> <p>Comme ici le séparateur est l'espace, on obtient : Salut ca va ?.</p>

<code>chaine.replace(old, new)</code>	<p>Retourne une chaîne résultat du remplacement des occurrences de la chaîne old par la chaîne new.</p> <pre>ch = "Bonjour Omar" res = ch.replace("jour", "soir") print(res)</pre> <p>On obtient : Bonsoir Omar</p>
---------------------------------------	---

IX. Les tranches :

Les chaînes sont des cas particuliers des listes donc ce qu'on peut faire avec les listes, on peut les faire avec les chaînes. C'est pour ça que l'on peut aussi utiliser les tranches.

On reconnaît les tranches avec la présence des deux points entre les crochets ([:]). L'index (inclus) avant les deux points correspond au début de la tranche. L'index (exclus) d'après correspond à la fin de tranche.

```
chaine = "Mohamed"
print(chaine[1:3])
print(chaine[:3])
print(chaine[1:])
print(chaine[:])
```

On obtient : **oh**, puis **:Moh**, puis **:ohamed**, puis **:Mohamed**

Disons que je veux mettre en majuscule la 1ere lettre du prénom de l'utilisateur.

```
prenom = input("Entrez votre prénom : ")
prenom_maj = prenom[0].upper() + prenom[1:]
print (prenom_maj)
```

Je mets en majuscule la 1ère lettre puis j'ajoute la suite grâce à une tranche.

X. Opérateur de formatage des chaînes de caractères (%)

Une des choses les plus importantes en PYTHON c'est l'opérateur (%) de formatage de chaînes de caractères.

```
print("Mon nom est %s j'ai %d ans " % ('Youssef', 22)) => Mon nom est Youssef j'ai 22 ans
```

Symbole de Formatage	Conversion
%c	Caractère
%s	Chaîne de caractères
%d	Entier signé
%u	Entier non signé
%f	Réel