|  |  |  |
| --- | --- | --- |
| **UNIVERSITE BADJI Mokhtar– ANNABA**  **FACULTE DES SCIENCES DE L’INGENIORAT**  **DEPARTEMENT D’INFORMATIQUE** |  | Intelligence Artificielle et Traitement de l’information **(IATI)**  Méthodes de Prétraitement de l’Information **(MPI)**  Réalisé par **: A. Melouah** |

**Python**

**Les chaines de caractères**

**Sommaire**

1- Présentation

2- Déclaration

3- Opérations

4- Méthodes

1. **Présentation**

Les chaînes de caractères (String) et *un type de donnée composite*. En d’autres termes, une chaine est une entité composée elle-même d'un ensemble d'entités plus petites : les caractères. Il n'y a pas le type caractère (character) en Python, un caractère est un string de longueur 1

1. **Déclaration**

Il y a deux manières pour déclarer, sur une seule ligne, un string : (1) par utilisation des apostrophes (") ; ou (2) par utilisation des guillemets (‘).

1. str1 = "Bonjour"
2. str2 = 'Bonjour
3. str3 = "C’est toi ?»
4. str4 = 'regarde la “chose”'

Une chaine sur plusieurs lignes doit être délimitée par 3 apostrophes au début de la chaine et 3 apostrophes à la fin de la chaine.

1. str = """Bonjour

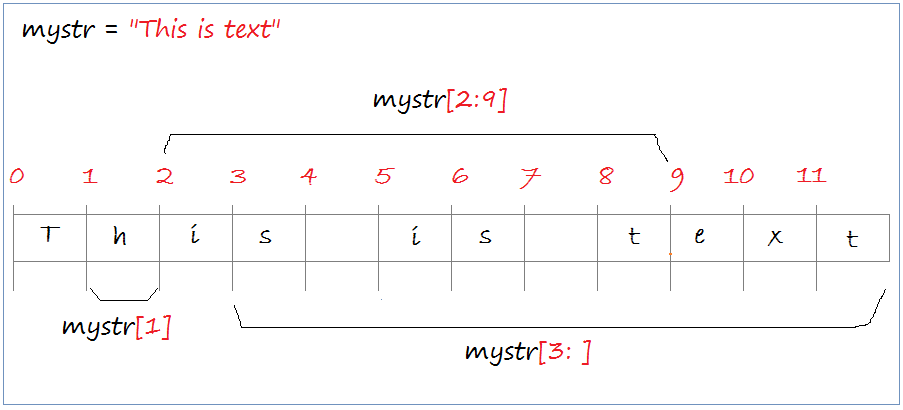
      C’est moi"""

Il existe des caractères d'échappement comme \n  (pour le retour à la ligne) ou caractère \t pour une tabulation qui peuvent être insérer dans une chaine.

Des chaînes brutes (Raw String)- Supprime la signification actuelle des caractères d'échappement. La syntaxe des chaînes brutes est exactement la même que pour les chaînes normales à l'exception de "**l'opérateur de chaîne brute**", la lettre "r", qui précède les guillemets. Le "r" peut être en minuscule (r) ou majuscule (R) et doit être placé immédiatement avant la première citation.

1. print (r'\n\t') ==> \n\t
2. print (R'\n\t') ==> \n\t
3. **Opérations**

Les chaînes de caractères peuvent être considérées comme des listes (de caractères)



L’utilisation d’un élèment de la chaine est similaire à celui de la liste

|  |  |  |
| --- | --- | --- |
| **Opérateur** | **Description** | **Exemple** |
| [i] | Elément à la position i | a = "Hello" a[1] ==> "e" |
| [n:m] | Tranche - n ièmecaractère est inclus, mais pas le m ième | a = "Hello" a[1:4] ==> "ell" a[1: ] ==> "ello" |

Il est impossible de modifier le contenu d’une chaîne définie. Par exemple, si Str est une chaine définie comme suit : Str="je ne change pas" ; l’instruction suivante : Str[0]="J" renvoie un message d'erreur.

Modifier une chaîne de caractères implique la construction d’une nouvelle. Comme il est possible de générer une liste, qui elle est modifiable, puis revenir à une chaîne de caractères.

La conversion d'une liste de chaînes de caractères en une chaîne de caractères fait appelle à la méthode .join()

Exemple

l[1] : liste='je suis la, je suis la'.split()

l[2] : liste

Out[1]: ['je', 'suis', 'la,', 'je', 'suis', 'la']

l[3] : liste[2]='ici'

l[4] : liste

Out[4]: ['je', 'suis', 'ici', 'je', 'suis', 'la']

l[5] : chaine=' '.join(liste)

l[6] : chaine

Out[6]: 'je suis ici je suis la'

|  |  |  |
| --- | --- | --- |
| **Opérateur** | **Description** | **Exemple** |
| [len](http://docs.python.org/library/functions.html?highlight=len#len)(nom) | déterminer le nombre de caractères présents dans une chaîne | [len](http://docs.python.org/library/functions.html?highlight=len#len)(‘nom’)=3 |
| + | Concaténation (concatenate) - Ajoute des valeurs de part et d'autre de l'opérateur | "Hello" +"Python" ==> "Hello Python" |
| \* | Répétition - Crée de nouvelles chaînes, concaténant (concatenate) plusieurs copies de la même chaîne. | "Hello"\*2 ==> "HelloHello" |
| in | Adhésion - Renvoie true si un caractère existe dans la chaîne donnée. | a = "Hello" 'H' in a ==> True |
| not in | Adhésion - Renvoie true si un caractère n'existe pas dans la chaîne donnée. | a = "Hello" 'M' not in a ==> True |

Exemple :

car = "e"

voyelles = "aeiouyAEIOUYàâéèêëùîï"

if car in voyelles:

print(car, "est une voyelle")

Exercice

Écrivez une fonction **estUnChiffre()** qui renvoie vrai, si l'argument transmis est un chiffre, et faux sinon.

1. **Méthodes**

Pour afficher toutes les méthodes il suffit de

In [1]: x = 'je suis une chaine'

In [3]: dir(x)

Pour s’informer sur une méthode (help) on utilise

In [3]: help(x.split)

Help on built-in function split:

split(...) method of builtins.str instance

S.split(sep=None, maxsplit=-1) -> list of strings

Return a list of the words in S, using sep as the

delimiter string. If maxsplit is given, at most maxsplit

splits are done. If sep is not specified or is None, any

whitespace string is a separator and empty strings are

removed from the result.

|  |  |  |
| --- | --- | --- |
| **Méthode** | **Action** | **Exemple** |
| lower() | renvoient un texte en minuscule | X=’TEST’  x=X.lower() |
| upper() | renvoient un texte en majuscule | Xx=x.upper() |
| capitalize() | mettre en majuscule la première lettre seulement | X1=x.capitalize() |
| split(sep=None, maxsplit=-1) | découpe une chaîne de caractères en plusieurs éléments appelés champs, en utilisant comme séparateur sep (espace(s) blanc(s) par défaut). | In[1]: animaux = "girafe+tigre+singe+souris"  In[2]: animaux.split('+',maxsplit=2)  Out[2]: ['girafe', 'tigre', 'singe+souris'] |
| find() | recherche une chaîne de caractères passée en argument. Si l'élément recherché est trouvé, alors l'indice du début de l'élément dans la chaîne de caractères est renvoyé, sinon la valeur -1 est renvoyée. Si l'élément recherché est trouvé plusieurs fois, seul l'indice de la première occurrence est renvoyé | In[4] : 'je suis la, je suis la'.find('suis')  Out[4]: 3 |
| replace() | substitue une chaîne de caractères par une autre | In[5] :'je suis la, je suis la'.replace('la', 'ici')  Out[5]: 'je suis ici, je suis ici' |
| count() | compte le nombre d’occurrences d'une chaîne de caractères passée en argument | In[5] :'je suis la, je suis la'.count('la')  Out[5]: 2 |
| startswith() | vérifie si une chaîne de caractères commence par une autre chaîne de caractères | In[6] :" j'ai des vides ".startswith('bon')  Out[67]: False  In[6] :"bonjour".startswith('bon')  Out[68]: True |
| strip() | permet de « nettoyer les bords » d'une chaîne de caractères | In[6] :" j'ai des vides ".strip()  Out[6]: "j'ai des vides" |