LYCÉE LISSANE EDDINE IBN AL-KHATIB
CENTRE CPGE –LAAYOUNE-

Cours: Les structures de données

-Les chaînes de caractères-(str)

Professeur: ARROU ABDESSELAM

Les Structures de données

Une structure de données(**containers**) est une structure qui regroupe plusieurs de données.

Type de base	Non modifiables Ou immutable	Modifiable Ou mutable	Accès
Simple(int, float, booleen)	X		-
Chaîne de caractères (str)	X		Séquentiel
Tuples(tuple)	X		Séquentiel
Listes(list)		X	Séquentiel
Ensembles(set)		X	Non séquentiel
Dictionnaire(dict)		X	Par clé

Définition d'une chaine de caractères

❖Définition :

On utilise les chaînes de caractères(suites de caractères) pour traiter des textes(**Exp: nom , prénom ou adresse postale**) .

Une chaîne de caractères est **une suite finie de caractères consécutifs**, qu'on note entre **apostrophes ' '** ou **guillemets " ".**

Exemple :

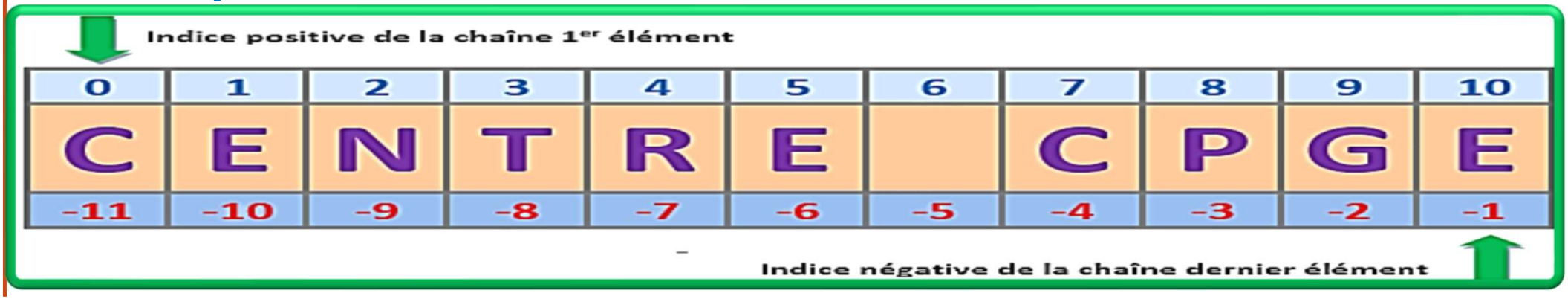
```
chaine="CENTRE CPGE" ou chaine='CENTRE CPGE'

Chaine="" #représente une chaine vide
```

Définition d'une chaine de caractères

❖Représentation d'une chaine :

Une chaine de caractère est représentée sous la forme d'un tableau(**List**) chaque caractères est rangée dans une case et identifié par son indice(**0 à N-1 ou -1 à –N**) .

Exemple :



Définition d'une chaine de caractères

❖**Remarque:**

Les chaînes ne sont pas des objets modifiables(immutable):

Cad on ne peut donc pas leur ajouter des caractères ou en enlever, ni trier leurs caractères par ordre croissant, ni modifier les caractères un par un.

Exemple:

```
>>> ch="ABCDEFGH"
>>> ch[0]="X" #changer A de l'induce 0 par la valeur "x"
Traceback (most recent call last):
  File "<pyshell#11>", line 1, in <module>
    ch[0]="X" #changer A de l'induce 0 par la valeur "x"
TypeError: 'str' object does not support item assignment
```

Définition d'une chaine de caractères

❖**Remarque:**

Les chaînes ne sont pas des objets modifiables(immutable):

Pour modifier une chaîne de caractères on doit construire une nouvelle chaîne qui peut remplacer la précédente avec le même identificateur.

Exemple:

```
>>>
>>> ch="ABCDEFGH"
>>> ch="X"+ch[1:]#changer A de l'indice 0 par la valeur "X"
>>> ch
'XBCDEFGH'
```

L'accès à un caractère d'une chaîne

on peut accéder aux caractères d'une chaîne par leurs **indices** dans la chaîne, et extraire des sous-chaînes d'une chaîne.

Les indices peuvent être positive(0 à N-1)ou négative(1 à -N).

Exemple:

```
>>> ch="Centre CPGE"
>>> ch[0] #renvoie le 1er caractère
'C'
>>> ch[-1] #renvoie le dernier caractère
'E'
>>> ch[7:]#extraire la sous chaine "CPGE"
'CPGE'
>>> ch[0:6]#extraire la sous chaine "Centre"
'Centre'
```

L'accès à un caractère d'une chaîne

Syntaxe :

Soit **Ch** est une chaine de caractères.

`Ch[i]`	**Accès au caractère d'indice i**
`Ch[-1]` ou `Ch[len(Ch)-1]`	**Accès au dernier caractère**
`Ch[i:j]` ou `Ch[i:j:1]`	**Extraire la sous-chaîne de l'élément i (inclus) à l'élément j (exclu) pas par défaut égal 1**
`Ch[i:j:k]`	**Extraire la sous-chaîne de l'élément i (inclus) à l'élément j (exclu) pas égal k**
`Ch[i:]`	**Extraire la sous-chaîne depuis l'élément i (inclus) jusqu'à la fin de la chaine.**
`Ch[:j]`	**Extraire la sous-chaîne de début indice 0 jusqu'à l'indice j (exclu).**
`Ch[:]`	**Extraire la sous-chaîne de début jusqu'à la fin de la chaine**

L'accès à un caractère d'une chaîne

Exemple :

```
>>> ch="Centre CPGE"
>>> taille=len(ch)#taille de la chaine
>>> taille
11
>>> ch[6]
' '
>>> ch[0]
'C'
>>> ch[-1]
'E'
>>> ch[:5] # equivalent de ch[0:5]
'Centr'
>>> ch[-3:]#equivalent de ch[-3:-1]
'PGE'
>>> ch[6:-2]
' CP'
>>> ch[-9:-4]
'ntre '
```

Remarque

Python ne supporte pas le type caractère. De là un caractère n'est plus qu'une chaine de **caractère de longueur 1**.

Exemple :

```
>>> ch="A"
>>> t=len(ch) #taille de la chaine
>>> t
1
```

Les opérations sur les chaines

Les opérations à appliquer sur une chaîne de caractères :

`+`	**La concaténation de deux chaines**
`*`	**La duplication d'une chaine (Répétition)**
`in` ou `not in`	**Test d'appartenance renvoie True ou False**
`>,<,<=,>=,!=, ==`	**Opérateurs de comparaisons renvoie True ou False**

Les opérations sur les chaines

Exemple :

```
>>> ch1="Centre"
>>> ch2=" CPGE"
>>> ch=ch1+ch2 #concaténation
>>> ch
'Centre CPGE'
>>> ch*4 #duplication
'Centre CPGECentre CPGECentre CPGECentre CPGE'
>>> 'A' in ch #test si 'A' présent dans la chaine ch
False
>>> 'G' in ch #test si 'A' présent dans la chaine ch
True
>>> ch1==ch2 #test si les deux chaines sont egaux
False
```

Parcourir d'une chaîne

Une chaîne de caractères est un **objet itérable** on peut utiliser **la structure répétitive for** pour **parcourir caractère par caractère une chaîne**.

Parcourir d'une chaîne

Example1: utilisation des indices avec la fonction range() et len()

```
>>> ch="Centre CPGE"
>>> #afficher tous les caractères de ch
>>> for i in range(taille):
        print(ch[i])


C
e
n
t
r
e

C
P
G
E
```

Parcourir d'une chaîne

Example2: utilisation des caractères au lieu des indices.

```
>>> ch="Centre CPGE"
>>> #afficher tous les caractères de ch
>>> for car in ch:
        print(car)


C
e
n
t
r
e

C
P
G
E
```

Parcourir d'une chaîne

Example3: Afficher des caractères avec leur indice par enumarate().

```
>>> ch="Centre CPGE"
>>> for i,car in enumerate(ch):
        print("indice :",i," caractère :",car)


indice : 0  caractère : C
indice : 1  caractère : e
indice : 2  caractère : n
indice : 3  caractère : t
indice : 4  caractère : r
indice : 5  caractère : e
indice : 6  caractère :
indice : 7  caractère : C
indice : 8  caractère : P
indice : 9  caractère : G
indice : 10  caractère : E
```

Les fonctions sur les chaines

❖La fonction len():

On utilise la fonction **len()** pour obtenir la longueur d'une chaîne.

Exemple :

```
>>> ch="Centre CPGE"
>>> taille=len(ch) #renvoie la taille de la chaine ch
>>> print("le nombre de caractères de la chaine est :",taille)
le nombre de caractères de la chaine est : 11
```

Les fonctions sur les chaines

❖La fonction ord(car):

On utilise la fonction **ord(car)** pour obtenir le code ascii d'un caractère (le code sera compris entre 0 et 255).

❖La fonction chr(ch):

On utilise la fonction **chr(n)** pour obtenir le caractère correspondant à un code ascii (**n** est un nombre compris entre 0 et 255) .

Les fonctions sur les chaines

Exemple :

```
>>> A=ord('A');a=ord('a');n=ord('9')

>>> print("code Ascii de A :",A,"\ncode Ascii de a :",a,"\ncode Ascii de 9 :",n)

code Ascii de A : 65
code Ascii de a : 97
code Ascii de 9 : 57
>>> A=chr(65);a=chr(97);n=chr(57)

>>> print("Caractère correspond:",A,"\nCaractère correspond:",a,"\nCaractère correspond",n)

Caractère correspond: A
Caractère correspond: a
Caractère correspond 9
```

Les méthodes sur les chaines

Sous Python, les chaines de caractères **(str)** sont des **objets** pour les quels on peut appliquer un certain nombre de méthodes (fonctions) particulièrement efficaces.

Exemple :

```
>>> ch="Centre CPGE"
>>> ch.upper() #convertir ch en majuscule
'CENTRE CPGE'
>>> ch.lower() #convertir ch eb miniscule
'centre cpge'
>>> ch.count('C')#nombre d'occurrence de 'C' dans ch
2
>>> ch.find('C') #renvoie True si 'C' presente dans ch
0
>>> ch.find('Cc') #renvoie True si 'C' presente dans ch -1 si non
-1
>>> ch.split(' ') #convertir ch en liste
['Centre', 'CPGE']
```

Remarque

Les fonctions et les méthodes ne modifient pas la chaine sur laquelle elles travaillent mais une créent une nouvelle chaine en cas de besoin.

Exemple :

```
>>> ch="Centre CPGE"
>>> ch.upper() #convertir ch en majuscule
'CENTRE CPGE'
>>> ch #la méthode upper() ne modifie pas ch
'Centre CPGE'
```

Les méthodes sur les chaînes

Fonctions	Description
chaine.upper()	Renvoie la chaîne en majuscule
chaine.lower()	Renvoie la chaîne en minuscule
chaine.count(ch)	Compte le nombre d'occurrence de "ch" dans la chaîne
chaine.find(ch)	Renvoie la position de "ch"dans la chaîne ou –1 si "ch"n'y est pas
chaine.replace(ch1,ch2)	Remplace chaque "ch1" par "ch2' dans la chaîne
chaine.capitalize()	Met la première lettre en majuscule
chaine.strip()	Supprime les espaces de début et de fin de la chaine

Les méthodes sur les chaînes

Fonctions	Description
chaine.isupper()	Renvoie True si la chaîne en majuscule
chaine.islower()	Renvoie True si la chaîne en minuscule
chaine.istitle()	Renvoie True si seule la 1er lettre de chaque mot de la chaîne est en majuscule
chaine.isalnum()	Renvoie True si la chaîne ne contient que des caractères alphanumériques
chaine.isalpha()	Renvoie True si la chaîne ne contient que des caractères alphabétique
chaine.isdigit()	Renvoie True si la chaîne ne contient que des caractères numériques
chaine.isspace()	Renvoie True si la chaîne ne contient que des espaces
chaine.startswith(prefix)	Renvoie True si la chaîne commence par "prefix"
chaine.endswith(suffix)	Renvoie True si la chaîne termine par "suffix"

Les conversions

- **Conversion d'un nombre en chaîne :**

```
>>> nb=123.56
>>> ch=str(nb)
>>> print(ch,"   ",type(ch))
123.56    <class 'str'>
```

- **Conversion d'une chaîne en nombre naturelle :**

```
>>> ch='126'
>>> nb=int(ch)#convertir chaine en nombre
>>> print(nb, " : ",type(nb))
126  :  <class 'int'>
```

- **Conversion d'une chaine en nombre réelle :**

```
>>> ch='126.89'
>>> nb=float(ch)#convertir chaine en nombre réelle
>>> print(nb, " : ",type(nb))
126.89  :  <class 'float'>
```

Les conversions

❖Conversion d'une chaine en liste :

```
>>> ch="Python"
>>> ch=list(ch)#convertir ch en liste
>>> print(ch, '   :',type(ch))
['P', 'y', 't', 'h', 'o', 'n']    : <class 'list'>
```

❖Conversion d'une chaîne en tuple :

```
>>> ch="Python"
>>> ch=tuple(ch)#convertir ch en tuple
>>> print(ch, '   :',type(ch))
('P', 'y', 't', 'h', 'o', 'n')    : <class 'tuple'>
```

Les caractères d'échappement

L'utilisation d'un antislash '\' dans une chaîne de caractères entraîne un comportement particulier de cette chaîne de caractères :

`\n`	**Provoque un retour à la ligne**
`\t`	**Provoque d'une tabulation**
`\r`	**Provoque une retour chariot**
`\a`	**Provoque un bip**
`\b`	**Provoque une retour arrière**
`\\`	**Permet d'écrire un seul antislash**

Exercice 01

Ecrire une fonction **palindrome(mot)** prenant en paramètre un **mot** et qui retourne **True** ou **False** selon que le mot est ou non un palindrome.

Un mot est un palindrome si il peut-être lu aussi bien de gauche à droite que de droite à gauche.

Exemple :

"**radar**", "**kayak**" et "**ressasser**" sont des mots palindromes.

Travaux pratiques