

05-cours_python_chaines_car

January 13, 2020

1 Manipulations de base sur les chaînes de caractères

1.1 Définition

Une chaîne peut se définir approximativement comme une suite de caractères. On utilise comme délimiteurs les caractères " ou '.

```
msg1 = "Bonjour"
msg2 = 'Bonjour'
msg3 = "J'aime"
msg4 = 'J'aime' # Erreur
```

1.2 Banaliser les caractères

Exercice : “banaliser” un caractère dans une chaîne

Afficher les phrases suivantes :

- Python c’est facile
- “facile” selon le prof !
- Python c’est “facile”

```
[1]: msg = "Python c'est facile"
      print(msg)
      msg = '"facile" selon le prof !'
      print(msg)
      msg = "Python c'est \"facile\""
      print(msg)
```

```
Python c'est facile
"facile" selon le prof !
Python c'est "facile"
```

1.3 Chaînes sur plusieurs lignes

```
print("""Bonjour,
J'espère que vous allez bien !
""")
```

Exercice : utilisation de variables

En utilisant deux variables (une pour le caractère '#', une pour le message) afficher à l’écran :

```
#####
Bienvenue
Degemer Mat
Willkommen
#####
```

```
[1]: cadre = "#####"
message = ""
Bienvenue
Degemer Mat
Willkommen
""
print(f"{cadre}{message}{cadre}") # ou bien : print(cadre + message + cadre)
```

```
#####
Bienvenue
Degemer Mat
Willkommen
#####
```

1.4 Autres “bizarreries”

Que fait le code suivant ?

```
msg = "#" * 80
print(msg)
```

```
[5]: msg = "#" * 80
print(msg)
```

```
#####
```

1.5 Indicage, extraction, longueur

Assignons la chaîne de caractères ISEN Bretagne à une variable :

```
isen_bretagne = "ISEN Bretagne"
```


1.5.1 Indicage

Une chaîne de caractères étant une séquence (ou suite) de caractères, il est possible d’accéder à un caractère particulier, connaissant son indice (sa position dans la chaîne). Sachant que l’indexation commençant à 0, on a donc :

```
[3]: isen_bretagne = "ISEN Bretagne"
print(isen_bretagne[0])
print(isen_bretagne[4]) # Oui, l'espace est un caractère
print(isen_bretagne[5])
print(isen_bretagne[20])
```

I

B

```
↳  -----  
↳ IndexError                                Traceback (most recent call↳  
↳ last)  
  
    <ipython-input-3-f8e34f9c1c5b> in <module>  
        3 print(isen_bretagne[4]) # Oui l'espace est un caractère  
        4 print(isen_bretagne[5])  
----> 5 print(isen_bretagne[20])  
  
IndexError: string index out of range
```

1.6 Extraction d'une sous-chaîne

Pour extraire des morceaux de la chaîne de caractères, on procède de la façon suivante :

```
sous_chaine1 = chaine[m:n]
```

Attention ! : le morceau de chaîne contiendra les caractères de la chaîne originale depuis l'index *m* **inclus** jusqu'à l'index *n* **exclu** (en d'autres termes, de l'index *m* à l'index *n-1* tous deux inclus).

```
[6]: chaine = "ISEN Bretagne"  
     print(chaine[0:4])  
     print(chaine[6:8])
```

```
ISEN  
re
```

A noter qu'on peut omettre les index de début ou de fin. Effectuer quelques tests du type :

```
[4]: chaine = "ISEN Bretagne"  
     sous_chaine1 = chaine[:4]  
     sous_chaine2 = chaine[5:]  
     print(f"{sous_chaine1} {sous_chaine2}")
```

```
ISEN Bretagne
```

1.6.1 Longueur d'une chaîne

Il est possible de connaître la longueur de la chaîne de caractères en utilisant la fonction *len()* :

```
longueur_chaine = len(string_isen)
print(longueur_chaine)
```

1.7 Exercice : extraire des morceaux d’une chaîne de caractères

Extraire les suites de caractères *ISEN* et *retag* à partir de la chaîne de caractères “ISEN Bretagne”. Placer le morceaux de chaînes dans deux variables *chaine1* et *chaine2* puis afficher ces deux variables.

```
[5]: chaine_isen = "ISEN Bretagne"
chaine1 = chaine[0:4]
chaine2 = chaine[6:11]
print(chaine1)
print(chaine2)
```

```
ISEN retag
```

1.8 Exercice : concaténation de sous-chaînes

Extraire les deux mots *ISEN* et *Bretagne* à partir de la chaîne d’origine “ISEN Bretagne”. En utilisant la concaténation de chaînes de caractères, reconstituer ensuite la chaîne ISEN Bretagne.

```
[6]: # 1ère possibilité
chaine = "ISEN Bretagne"
mot1 = chaine[:4]
mot2 = chaine[5:]
# 1ère possibilité
chaine2 = mot1 + " " + mot2
print(chaine2)
# 2ème possibilité (mieux)
chaine2 = f"{mot1} {mot2}"
print(chaine2)
```

```
ISEN Bretagne
ISEN Bretagne
```

1.9 Exercice : extraire les voyelles d’une chaîne

- Demander un mot à l’écran (en minuscules)
- Parcourir ce mot
- A la fin, afficher la liste des voyelles

```
[15]: # 1ère version
mot = "python"
i = 0
voyelles = ""
while i < len(mot):
    if mot[i] == 'a' or mot[i] == 'e' or mot[i] == 'i' or mot[i] == 'o' or
    ↪ mot[i] == 'u' or mot[i] == 'y':
```

```

        voyelles += mot[i]
    i+=1
print(voyelles)

```

yo

```

[16]: # 2ème version : plus élégante
mot = "python"
i = 0
voyelles = ""
while i < len(mot):
    if mot[i] in "aeiouy":
        voyelles += mot[i]
    i+=1
print(voyelles)

```

yo

1.10 Quelques opérations possibles sur les chaînes de caractères

Il existe de nombreuses possibilités pour effectuer des opérations sur des chaînes de caractères :

- Mise en minuscules / majuscules
- Suppression des espaces au début et à la fin
- Recherche, remplacement...

Les chaînes de caractères disposent de méthodes (~fonction) qui vont permettre ces opérations.

Voici quelques exemples :

```

[1]: #
# Mise en majuscules
#
my_str = "sacré toto"
# upper() est une méthode (~fonction) de l'objet str
my_str_upper = my_str.upper()
print(my_str) # my_str n'est pas modifiée
print(my_str_upper)
# On n'est (évidemment) pas obligé de passer par une autre variable
my_str = my_str_upper

#
# Suppression des espaces
#
my_str2 = "    fait pas beau    "
print(f"***{my_str2}***")
my_str2 = my_str2.strip()
print(f"***{my_str2}***")

```

```
sacré toto
SACRÉ TOTO
***    fait pas beau    ***
***fait pas beau***
```

1.10.1 Exercice : trouver deux façons de tester si une chaîne de caractères est vide

```
[ ]: # 1ère façon
str_value1 = input("Saisissez une chaîne de caractères ")
if str_value1.strip() == '':
    print("Chaîne vide")
else:
    print("Chaîne NON vide")

# 2ème façon
str_value2 = input("Saisissez une chaîne de caractères ")
if len(str_value2.strip()) == 0:
    print("Chaîne vide")
else:
    print("Chaîne NON vide")
```

1.10.2 Exercice : demander à l'utilisateur de saisir une réponse :

- S'il répond "oui" afficher "On continue !"
- Sinon afficher "Stop !"

On ne doit pas tenir compte des majuscules / minuscules dans la réponse (on dit qu'on "ignore la casse").

```
[8]: reponse = input("Réponse (oui / non) : ")
reponse = reponse.strip() # Suppression des espaces éventuels avant et après
reponse = reponse.lower() # On met tout en minuscules
# On aurait pu faire aussi une seule opération : reponse = reponse.strip().
# → lower()

if reponse == "oui":
    print("On continue !")
else:
    print("Stop !")
```

```
Réponse (oui / non) : Oui
On continue !
```

```
[ ]: ## Exercice : chaîne de caractères et liste
Quelle est la différence entre ces deux codes ?
python
liste_car = ['b', 'o', 'n', 'j', 'o', 'u', 'r', ' ', 'I', 'S', 'E', 'N']
for c in liste_car:
```

```

    print(c)
~ ~ ~

et :

~ ~ ~python
message = "bonjour ISEN"
for c in message:
    print(c)
~ ~ ~

Que se passe t-il si on effectue :

~ ~ ~python
liste_car[7] = '-'
message[7] = '-'
~ ~ ~

```

1.11 Exercice : former une chaîne de caractères à partir d'une liste

- Reformier une chaîne de caractères à partir de la liste ci-dessous en utilisant la fonction *join* :

```

# ["toto", "est", "beau", "quand", "il", "le", "faut"]
# Résultat attendu : "toto est beau quand il le faut"

```

```

[ ]: my_list = ["toto", "est", "beau", "quand", "il", "le", "faut"]
my_str = " ".join(my_list)
print(my_str)

```

2 Exercices de révision

2.1 Concaténation de chaînes de caractères

Afin de mettre en place un système de caisses automatiques, le directeur du cinéma *Les Studios* vous demande de réaliser un programme demandant au client son âge, s'il est abonné et lui annonçant en retour le tarif appliqué :

- Moins de 18 ans : 4 €
- Abonné : 5,50 €
- Sinon : 8 €

Afficher selon les valeurs saisies par le client, par exemple :

- Vous avez 12 ans, le montant à régler est de 4 €
- Vous êtes abonné, le montant à régler est de 5,50 €

```
[7]: age = int(input("Quel est votre âge ? "))
    abonne = 'n'
    if age < 18:
        prix = 4.00
    else:
        abonne = input("Etes-vous abonné ? (o/n) ")
        if abonne.lower() == 'o':
            prix = 5.50
        else:
            prix = 8.00

    if abonne.lower() == 'o':
        print(f"Vous êtes abonné, le montant à régler est de {prix} €")
    else:
        print(f"Vous avez {age} ans, le montant à régler est de {prix} €")
```

Quel est votre âge ? 12

Vous avez 12 ans, le montant à régler est de 4.0 €

2.2 Chaînes de caractères et conditions

Le cinéma “Liberté” étant proche de la faillite, à cause notamment de la forte concurrence des Studios, les dirigeants décident de mener une politique agressive sur les tarifs. Après avoir réfléchi toute la nuit ils vous communiquent la façon dont la nouvelle version du logiciel doit calculer les prix :

- Demander à l'utilisateur son âge
- Si il a entre 0 et 4 ans ou plus de 80 ans, c'est gratuit
- Si il a plus de 100 ans on lui donne 5 €
- Si il a entre 5 et 17 ans il paye 3 € et en plus on lui offre un seau de Popcorn
- Si il a 18 ans ou plus et qu'il habite Brest il paye 7 €. S'il n'habite pas Brest c'est 8 €
- Ah au fait, quelque soit son âge, s'il habite Quimper c'est 50 €
- Le programme doit fonctionner de la même façon s'il y a des majuscules / minuscules dans le nom de la ville.

```
[1]: # Solution 1

ville = input("Où habitez-vous ? ")
ville = ville.strip().lower()

if ville == 'quimper':
    print("Euh... c'est 50 €")
else:
    age = int(input("Votre âge ? "))
    if age < 5:
```



```

    print("C'est gratuit !")
elif age > 100:
    print(f"On vous rembourse : 5 Euro")
elif age >= 5 and age <= 17:
    print("Tiens un seau de popcorn ! Mais tu dois régler 3 €")
else: # L'utilisateur a 18 ans ou plus (mais <= 100 ans)
    if ville == 'brest':
        print("Vous devez payer 7 €")
    else:
        print("Vous devez payer 8 €")

```

Où habitez-vous ? brest

Votre âge ? 12

Tiens un seau de popcorn ! Mais tu dois régler 3 Euro

[2]: # Solution 2

```

ville = input("Où habitez-vous ? ")
ville = ville.strip().lower()

if ville == 'quimper':
    tarif = 50.0
else:
    tarif = 0.0
    age = int(input("Votre âge ? "))
    if not ((age > 0 and age <=4) or (age > 80)):
        # L'utilisateur n'a pas entre 0 et 4 ans et n'a pas plus de 80 ans
        if age > 100:
            tarif = -5.0
        elif age >= 5 and age <= 17:
            print("Tiens un seau de popcorn !")
            tarif = 3.0
        else:
            # L'utilisateur a plus de 17 ans (mais n'a pas plus de 80 ans)
            if ville == 'brest':
                tarif = 7.0
            else:
                tarif = 8.0
        # Si la condition not ((age > 0 and age <=4) or (age > 80)) n'est pas
        ↳ respectée
        # alors tarif = 0.0 (valeur mise au départ)
    if tarif == 0.0:
        print("C'est gratuit !")
    elif tarif > 0:
        print(f"Vous devez régler : {tarif} Euro")
    else:
        print(f"On vous rembourse : {tarif} Euro")

```

Où habitez-vous ? brest
Votre âge ? 12
Tiens un seau de popcorn !
Vous devez régler : 3.0 Euro