Les Chaînes de Caractères en Python

💡 Définition et Création

Une **chaîne de caractères** (ou `string`) est une séquence ordonnée et **immuable** de caractères. On la reconnaît par les guillemets simples ' ' ou doubles " " qui l'entourent. Une chaîne de caractères peut contenir des lettres, des chiffres, des symboles et des espaces.

```
root@python:~$
```

```
# Création d'une chaîne de caractères vide
ma_chaine = ""
# Création d'une chaîne avec du texte
salut = "Bonjour le monde"
# Obtenir la longueur d'une chaîne avec len()
longueur = len(salut)
print(f"La chaîne 'salut' contient {longueur} caractères.")
# Affiche "La chaîne 'salutation' contient 16 caractères."
```

⚙️ Saisie et Traitement

Saisie avec `input()`

La fonction `input()` permet de demander à l'utilisateur de saisir une chaîne de caractères. Le programme s'arrête en attendant que l'utilisateur entre du texte et appuie sur Entrée. La valeur saisie est renvoyée sous forme de chaîne de caractères.

```
root@python:~$
```

```
nom = input("Entrez votre nom : ")
print(f"Bonjour, {nom} !")
# Si l'utilisateur tape "Alice"
# la console affichera : "Bonjour, Alice !"
```

Méthodes de modification de la casse

Les chaînes de caractères sont **immuables**, ce qui signifie que vous ne pouvez pas modifier un caractère existant directement. Les méthodes suivantes renvoient une **nouvelle** chaîne de caractères modifiée.

.upper() : renvoie une nouvelle chaîne avec tous les caractères en majuscules.

.lower() : renvoie une nouvelle chaîne avec tous les caractères en minuscules.

.capitalize() : renvoie une nouvelle chaîne avec la première lettre en majuscule.

```
root@python:~$
```

```
chaine = "Hello World"
print(chaine.upper())    # Affiche "HELLO WORLD"
print(chaine.lower())    # Affiche "hello world"
```

🤔 Immuabilité

⚠️ Une chaîne de caractères ne peut pas être modifiée après sa création. Toutes les opérations qui semblent "modifier" une chaîne, comme le remplacement d'un caractère, créent en réalité une **nouvelle** chaîne en mémoire.

```
root@python:~$
```

```
slogan = "J'adore Python"
# Cette instruction générera une erreur !
# slogan[8] = 'C'
# TypeError: 'str' object does not support item assignment
# Solution : créer une nouvelle chaîne
nouveau_slogan = slogan[:8] + "le Python"
print(nouveau_slogan) # Affiche "J'adore le Python"
```

🧭 Indexation et Accès aux éléments

Chaque caractère d'une chaîne possède une position, appelée index. En Python, ⚠️ **l'indexation commence à 0**.

On peut aussi utiliser des index négatifs pour accéder aux caractères à partir de la fin de la chaîne. L'index -1 correspond au dernier caractère, -2 à l'avant-dernier, et ainsi de suite.

```
root@python:~$
```

```
mot = "Python"
# Indexation positive
print(mot[0])   # Affiche "P"
# Indexation négative
print(mot[-1])  # Affiche "n"
# Le slicing : Extraire une partie de la chaîne [début:fin]
# ⚠️ La fin est exclue !
print(mot[1:4]) # Affiche "yth"
print(mot[2:])  # Affiche "thon"
```

📜 Parcourir une chaîne

La boucle for est l'outil principal pour parcourir une chaîne de caractères.

1. Par caractère :

C'est la méthode la plus simple et la plus courante.

```
root@python:~$
```

```
mot = "code"
for caractere in mot:
  print(caractere)
```

```
Affiche :
c
o
d
e
```

2. Par index :

Utile si vous avez besoin de la position du caractère en plus de sa valeur. On utilise la fonction range(len(chaine)).

```
root@python:~$
```

```
mot = "code"
for i in range(len(mot)):
  print(f"Caractère n°{i} : {mot[i]}")
```

```
Affiche :
Caractère n°0 : c
Caractère n°1 : o
Caractère n°2 : d
Caractère n°3 : e
```

#HACK AGAINST MACHISM

FOLLOW THE PYTHONIC WAY
#ACK AGAINST MACHISM