Chaînes de caractères : méthodes

Bases de Python

Résumé

Ce chapitre complète le chapitre sur les chaînes de caractères. Il présente les méthodes de parcours, de transformation et de manipulation des chaînes en Python. Il utilise la boucle `for`, présentée dans le chapitre sur les boucles. Si vous n'avez pas encore vu les boucles, vous pouvez revenir à la section « Parcourir une chaîne » plus tard.

1 Parcourir une chaîne

On peut parcourir les caractères d'une chaîne avec une boucle `for` :

```
mot = "NSI"
for lettre in mot:
    print(lettre)
```

Pour avoir à la fois l'indice et le caractère, on utilise `enumerate()` :

```
mot = "NSI"
for i, lettre in enumerate(mot):
    print(f"Indice {i} : {lettre}")
```

Exemple concret. Compter le nombre de voyelles dans un mot :

```
mot = "programmation"
compteur = 0
for lettre in mot:
    if lettre in "aeiouy":
        compteur += 1
print(f"'{mot}' contient {compteur} voyelles")
```

2 Méthodes de changement de casse

Les méthodes ne modifient pas la chaîne d'origine (rappel : les chaînes sont immuables). Elles renvoient une **nouvelle** chaîne.

```
mot = "Bonjour"
print(mot.upper())       # BONJOUR
print(mot.lower())       # bonjour
print(mot.capitalize())  # Bonjour (première lettre en majuscule)
print(mot)               # Bonjour (l'original n'a pas changé)
```

Application. Comparer deux chaînes sans tenir compte de la casse :

```python
saisie = "OUI"
if saisie.lower() == "oui":
    print("L'utilisateur a confirmé")
```

③ Méthodes de recherche et remplacement

```python
phrase = "Python est un langage puissant"
print(phrase.count("an"))          # 2
print(phrase.find("langage"))      # 14 (indice du début)
print(phrase.find("Java"))         # -1 (non trouvé)
print(phrase.replace("puissant", "élégant"))
```

La méthode `find()` renvoie l'indice de la première occurrence, ou `-1` si la sous-chaîne n'est pas trouvée. La méthode `replace()` remplace **toutes** les occurrences.

④ Méthodes de nettoyage

La méthode `strip()` supprime les espaces (et les retours à la ligne) en début et en fin de chaîne. C'est très utile pour nettoyer les saisies utilisateur ou les lignes lues dans un fichier :

```python
saisie = "   Alice   "
print(saisie.strip())     # "Alice" (supprime des deux côtés)
print(saisie.lstrip())    # "Alice   " (supprime à gauche)
print(saisie.rstrip())    # "   Alice" (supprime à droite)
```

⑤ Méthodes de vérification

Ces méthodes renvoient un booléen (`True` ou `False`) :

```python
print("42".isdigit())       # True  (ne contient que des chiffres)
print("abc".isalpha())      # True  (ne contient que des lettres)
print("abc123".isalnum())   # True  (lettres et chiffres uniquement)
print("HELLO".isupper())    # True
print("hello".islower())    # True
```

Application. Vérifier qu'une saisie est bien un nombre avant de la convertir :

```python
saisie = input("Entrez un nombre : ")
if saisie.isdigit():
    nombre = int(saisie)
    print(f"Le double est {nombre * 2}")
else:
    print("Ce n'est pas un nombre valide")
```

⑥ Exercices

Exercice 1 *Méthodes de chaînes* **(3 points)** Soit la chaîne `phrase = " Bonjour le Monde "`.

1. Que renvoie `phrase.strip()` ?

2. Que renvoie `phrase.strip().lower()` ?

3. Que renvoie `phrase.strip().replace("Monde", "Python")` ?

4. Que renvoie `phrase.strip().startswith("Bon")` ?

Exercice 2 *Parcours de chaîne* **(4 points)** Écrire un programme qui demande un mot à l'utilisateur et affiche :

1. le nombre de voyelles dans le mot ;
2. le mot écrit à l'envers.

Exemple : pour `"python"`, le programme affiche : « 2 voyelles » et `"nohtyp"`.

Exercice 3 *Validation de mot de passe* **(4 points)** Écrire un programme qui demande un mot de passe et vérifie s'il est « fort » selon les critères suivants :

— au moins huit caractères de long ;

— contient au moins une lettre ;

— contient au moins un chiffre.

Le programme affiche « Mot de passe fort » ou les critères non respectés.

Indication : utiliser les méthodes `.isalpha()` et `.isdigit()` sur chaque caractère.

Résumé des méthodes

Méthode	Rôle	Exemple
`s.upper()`	Tout en majuscules	`"nsi".upper()` → `"NSI"`
`s.lower()`	Tout en minuscules	`"NSI".lower()` → `"nsi"`
`s.capitalize()`	Première lettre en majuscule	`"nsi".capitalize()` → `"Nsi"`
`s.strip()`	Supprimer les espaces aux extrémités	`" ok ".strip()` → `"ok"`
`s.find(x)`	Indice de la première occurrence	`"abc".find("b")` → `1`
`s.count(x)`	Nombre d'occurrences	`"abab".count("ab")` → `2`
`s.replace(a, b)`	Remplacer toutes les occurrences	`"abc".replace("b", "X")` → `"aXc"`
`s.isdigit()`	Ne contient que des chiffres ?	`"42".isdigit()` → `True`
`s.isalpha()`	Ne contient que des lettres ?	`"abc".isalpha()` → `True`
`s.startswith(x)`	Commence par `x` ?	`"Python".startswith("Py")` → `True`
`s.endswith(x)`	Finit par `x` ?	`"fichier.csv".endswith(".csv")` → `True`