Comment supprimer les espaces dans une chaîne en python

1. Objectifs

- Utiliser des méthodes de chaîne et des expressions régulières en Python pour supprimer les esapaces dans une chaine

2. Présentation

- Pour supprimer les espaces blancs d'une chaîne. On pourrait principalement le classer en deux méthodes différentes, l'une est la méthode Python **str**, comme `str.split()` et `str.replace()`; l'autre est la méthode Python d'expression régulière.
- Dans ce tutoriel je vais vous montrer comment supprimer les espaces
  1. Au début d'une chaîne en Python
  2. A la fin d'une chaîne de caractères en Python
  3. Au début et à la fin d'une chaîne en Python
  4. D'une chaîne en Python
  5. Ne supprimer que les espaces dupliqués d'une chaîne en Python

3. Supprimer les espaces au début d'une chaîne en Python



1. **Méthode `str.lstrip()`**

   - **Syntaxe:**

     ```
     str.lstrip([chars])
     ```

   - **Paramètres**

- Renvoie une copie de la chaîne sans les caractères de début. L'argument **chars** est une chaîne spécifiant le jeu de caractères à supprimer.
- S'il est omis ou Aucun, l'argument **chars** supprime par défaut **les espaces**.

- **Exemple:**

```
str = '         Python est un langage très facile'
print(str.lstrip())
#output
Python est un langage très facile
```

2. **Méthode de regex Python (re.sub)**

- **Syntaxe:**

```
resultat = re.sub(pattern, repl, string, count=0, flags=0);
```

- **Paramètres**

  - **pattern** - spécifiez le pattern à remplacer
  - **repl** - en remplaçant les occurrences de pattern non superposées les plus à gauche dans la chaîne par le repl .
  - **count** - L'argument facultatif count est le nombre maximal d'occurrences de modèle à remplacer. Si omis ou nul, toutes les occurrences seront remplacées.
  - **pattern**= `r"^\s+"`
    - `r`: La notation de chaîne brute de Python pour les expressions régulières
    - `\s`: correspond aux caractères d'espacement.
    - `+`: Il correspondra à une ou plusieurs répétitions de caractères d'espacement.
    - `^(Caret.)` : Correspond au début de la chaîne

- **Exemple:**

```
import re
str = '         Python est un langage très facile'
```

apcpedagogie.com

```
resultat = re.sub(r"^\s+","",str)
print(resultat)
#output
#Python est un langage très facile
```

4. Supprimer les espaces à la fin d'une chaîne de caractères en Python



1. **Méthode `str.rstrip()`**

- **Définition et utilisation**
  - La méthode `rstrip()` supprime tous les caractères de fin (caractères à la fin d'une chaîne), l'espace est le caractère de fin par défaut à supprimer.
  - La méthode `rstrip()` renvoie une copie de la chaîne avec les caractères de fin supprimés (en fonction de l'argument de chaîne passé).
  - Le `rstrip()` supprime les caractères de la droite en fonction de l'argument (une chaîne spécifiant l'ensemble de caractères à supprimer).
  - Renvoie une copie de la chaîne en supprimant les espaces de fin.
- **Syntaxe:**

```
str.rstrip([caractères])
```

- **Paramètres**
  - **caractères** (facultatif) - une chaîne spécifiant l'ensemble de caractères à supprimer.
  - Si caractères n'est pas fourni, tous les espaces à droite sont supprimés de la chaîne.
  - Toutes les combinaisons de caractères dans caractères sont supprimés de la droite de la chaîne jusqu'à la première incompatibilité.

apcpedagogie.com

- **Exemple:**

```
str = 'Python est un langage très facile         '
print(str.rstrip())
#output
Python est un langage très facile
```

2. **Méthode de regex Python (re.sub)**

- **Syntaxe:**

```
resultat = re.sub(pattern...);
```

- **Paramètres**
  - **pattern** - spécifiez le pattern à remplacer
  - **pattern**= `r"\s+$"`
    - `r`: La notation de chaîne brute de Python pour les expressions régulières
    - `\s`: correspond aux caractères d'espacement.
    - `+`: Il correspondra à une ou plusieurs répétitions de caractères d'espacement.
    - `$`: Correspond à la fin de la chaîne

- **Exemple:**

```
import re
str = '         Python est un langage très facile'
resultat = re.sub(r"^\s+$","",str)
print(resultat)
#output
#Python est un langage très facile
```

apcpedagogie.com

5. Supprimer les espaces au début et à la fin d'une chaîne en Python



1. **Méthode `str.strip()`**

   - **Définition et utilisation**

     - La méthode `strip()` renvoie une copie de la chaîne en supprimant à la fois les caractères de début et de fin (en fonction de l'argument de chaîne passé).
     - La méthode `strip()` supprime les caractères de gauche et de droite en fonction de l'argument (une chaîne spécifiant l'ensemble de caractères à supprimer).
     - Cette méthode renvoie une copie de la chaîne avec les caractères de début et de fin supprimés.

   - **Syntaxe:**

     ```
     str.strip([caractères])
     ```

   - **Paramètres**

     - **caractères** (facultatif) - une chaîne spécifiant l'ensemble de caractères à supprimer des parties gauche et droite de la chaîne.
     - Si l'argument "caractères" n'est pas fourni, tous les espaces blancs de début et de fin sont supprimés de la chaîne.

   - **Exemple:**

     ```
     str = '         Python est un langage très facile         '
     print(str.strip())
     #output
     Python est un langage très facile
     ```

apcpedagogie.com

2. **Méthode de regex Python (re.sub)**

   - **Syntaxe:**

     ```
     resultat = re.sub(pattern...);
     ```

   - **Paramètres**

     - **pattern** - spécifiez le pattern à remplacer
     - **pattern**= `^\s+|\s+$`
       - **`^\s+`**: Correspond à un ou plusieurs espaces au début de la chaîne
       - **`\s+$`**: Correspond à un ou plusieurs espaces à la fin de la chaîne.
       - `|`: x | y correspond à x ou y

     ```
     import re
     str = '         Python est un langage très facile'
     resultat = re.sub(^\s+|\s+$,"",str)
     print(resultat)
     #output
     #Python est un langage très facile
     ```

6. Supprimer tous les espaces d'une chaîne en Python



1. **Méthode `str.replace()`**

   - **Définition et utilisation**

     - La méthode `replace()` renvoie une copie de la chaîne où toutes les occurrences d'une sous-chaîne sont remplacées par une autre sous-chaîne.

   - **Syntaxe:**

     ```
     str.replace(old, new [, count])
     ```

- **Paramètres**
  - **old** (obligatoire) - ancienne sous-chaîne que vous souhaitez remplacer.
  - **new** (obligatoire) - nouvelle sous-chaîne qui remplacera l'ancienne sous-chaîne.
  - **count** (facultatif) - le nombre de fois que vous souhaitez remplacer le vieille sous-chaîne avec le nouveau sous-chaîne.
  - si **count** n'est pas spécifié, la méthode `replace()` remplace toutes les occurrences de **vieille** sous-chaîne avec le **nouveau** sous-chaîne.

- **Exemple:**

```
str = '         Python est un langage très facile         '
print(str.replace(" ", ""))
#output
Pythonestunlangagetrèsfacile
```

2. **Méthode de regex Python (re.sub)**

- **Syntaxe:**

```
resultat = re.sub(pattern...);
```

- **Paramètres**
  - **pattern** - spécifiez le pattern à remplacer
  - **pattern**= `r"\s+"`
    - **^\s+**: Correspond à un ou plusieurs espaces

```
import re
str = '         Python est un langage très facile'
resultat = re.sub("\s+","",str)
print(resultat)
#output
#Pythonestunlangagetrèsfacile
```

apcpedagogie.com

7. ## Ne supprimer que les espaces dupliqués d'une chaîne en Python



1. **Méthode `str.split()`**

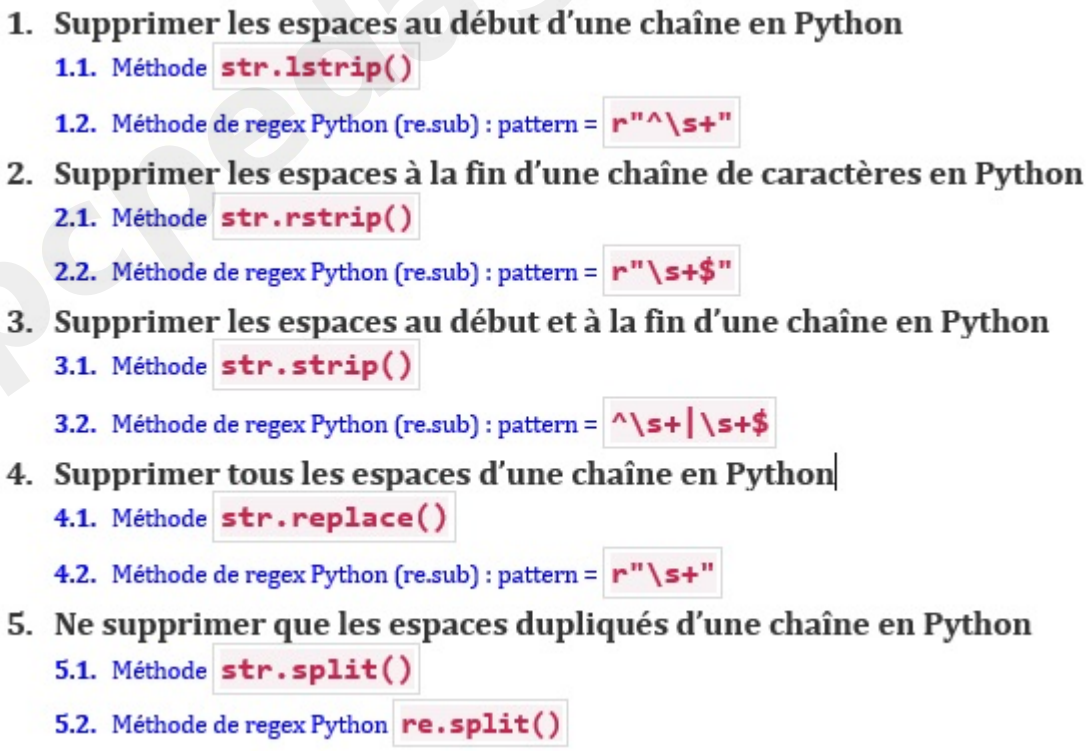
    - **La méthode split () en Python**

        - Lire notre article Utiliser la méthode split () en Python

2. **Méthode de regex Python (re.split)**

    - Lire notre article La méthode split () en Python

8. ## Résumé



9. Vous pouvez regarder la vidéo suivante:

apcpedagogie.com

Cours Python

Source : https://apcpedagogie.com/comment-supprimer-les-espaces-dans-une-chaine-en-python · téléchargé le 08/11/2026 · © APC Pédagogie · apcpedagogie.com

apcpedagogie.com