Utilisation en python

Une **string** en python peut être initialisée avec des simples guillemets `''` (ou simple quote) ou avec des double guillemets `""` (ou double quote).

Je vous conseille cependant d'utiliser les double quotes car en cas de présence d'une apostrophe dans votre string, vous aller devoir l'échapper avec un anti-slash `\` contrairement à une string initialisée avec des double quotes.

```
simple_quote = 'En simple quote, on doit échapper l\'apostrophe'
double_quote = "En double quote, l'apostrophe est conservée tel quel"
print(simple_quote)
```

Par ailleurs, d'autres langages de programmation plus bas niveau tel que C#, Java, C++ et C font la **distinction** entre les simples quotes destinées à un seul caractère et les double quotes destinées au string (2 caractères ou +). Autant prendre les bonnes habitudes dès le début.

```
String simpleQuote = 'Ça va pas compiler!';
String doubleQuote = "Ok, là j'accepte.";
```

Les chaînes de caractères multilignes

Elles permettent d'initialiser des longues strings et de conserver la mise en forme.

```
simple_quote = 'Avec une simple quote, je ne peux pas passer à la ligne \n sauf à
échapper n...'

double_quote = """Je suis une chaine
mais je peux m'étendre sur plusieurs lignes"""


simple_quote_triple = '''Même chose, et le truc cool
c'est que je n'ai plus besoin
d'échapper les apostrophes!'''

print(simple_quote)
print(double_quote)
print(simple_quote_triple)
```

Les caractères spéciaux

En incorporant des **caractères spéciaux** vous pouvez faire de la mise en forme pour votre string.

```
print("passement\nde ligne!")
```

Voici donc une liste non exhaustive des caractères spéciaux fréquemment utilisés en développement :

- \n passage à la ligne
- \t tabulation
- \b retour arrière

Échappement des caractères spéciaux dans une string

En Python, une chaîne brute (raw) est une chaîne préfixée par un « r » ou un « R ». Il est utilisé pour traiter les anti-slashs () comme des caractères littéraux et non comme des caractères d'échappement. Ceci est particulièrement utile lorsque vous travaillez avec des expressions régulières, des chemins de fichiers ou tout autre contexte dans lequel des barres obliques inverses sont utilisées.

Pour échapper les caractères spéciaux il faut créer une **"Raw String"** de la manière suivante :

```
raw_string = r"Je suis une raw string,\nsans passage à la ligne!"
print(raw_string)
```

⚠️ Anti-slashs ⚠️

Attention à la manipulation des chemins de fichier windows qui comportent une multitude.
Vous serez amenés à "échapper" le caractère d'échappement pour qu'il soit considéré comme caractère imprimable.

```
path = "C:\\User\\Bob"
pathraw = r"C:\User\Bob"
print(path)
print(pathraw)
```