

[Welcome home !](#)[Règlement intérieur](#)[Poster une issue](#)[JavaScript](#)[Environnement de dev](#)[👋 Semaine 1](#)[Terminal](#)[Variables](#)[Conditions](#)[Boucles](#)[🎥 Vidéos](#)[Concept et utilisation](#)[Cas pratique - String](#)[👋 Semaine 2](#)[Manipuler des chaînes de caractères](#)[Fonctions](#)[Tableaux](#)[👋 Semaine 3](#)[Déroulement d'un programme](#)[Les booléens](#)[Les valeurs truthy et falsy](#)[Notion de scope](#)[Conditions d'existence](#)[Conditions imbriquées](#)[👋 Semaine 4](#)[Objets](#)[Requêtes à API](#)[Backend & Strapi](#)[HTML / CSS](#)

Passer en revue les lettres d'une chaîne de caractères

Les boucles `for` sont très pratiques pour parcourir les chaînes de caractères.

✓ Souvenez-vous : dans une `string`, chaque lettre porte un `index` : la première a l'index 0, la seconde l'index 1, etc... jusqu'à la dernière lettre, qui possède comme `index` la longueur de la chaîne - 1.

Dans l'illustration suivante est représentée une "chaîne de caractères", comme une chaîne dont chaque maillon serait une lettre. Sur cette représentation, `str[0]` désigne la première lettre, `str[1]` la deuxième, etc... Il est donc considéré que la chaîne de caractère est stockée dans une variable appelée `str`.

Mettez donc à profit cette caractéristique : utilisez la variable `i` de la boucle `for` comme index !

```
const str = "JavaScript";

for (let i = 0; i < str.length; i++) {
  // la boucle démarre à 0, index de la pr
  // et termine à str.length - 1, index de
  console.log(str[i])
}
```

Voici ce qu'affiche le terminal :

```
J => au premier tour i = 0 donc str[i] = s
a => au premier tour i = 1 donc str[i] = s
v
```