

Les chaines de caractères

Les chaînes

- Tous les langages de programmation prévoient une gestion des chaînes de caractères. JavaScript est particulièrement adapté pour le traitement des chaînes.
- L'objet String contient de nombreuses méthodes permettant de manipuler les chaînes.

La déclaration d'une chaîne de caractère

- Pour déclarer une chaîne de caractères, vous pouvez utiliser les séparateurs guillemets (") ou apostrophe (').

```
var chaine1="Bonjour";
```

```
var chaine2='Bonjour';
```

La concaténation de deux chaines

- L'opération de base est la concaténation de chaînes. Elle consiste à assembler deux chaînes en une seule. L'opérateur est le **+**, à ne pas confondre avec l'opérateur addition qui s'applique aux nombres.

```
var chaine1="Vive le ";  
var chaine2="JavaScript";  
var chaine=chaine1+chaine2;  
document.write(chaine);
```

La longueur d'une chaîne de caractère

- Une chaîne de caractères en JavaScript est un objet string sur lequel s'appliquent des propriétés et des méthodes.
- La propriété length indique le nombre de caractères de la chaîne.
- Pour une chaîne définie par :

```
var chaine="azerty";  
var l=chaine.length ;      //retourne le nombre de caractères, ici 6 caractères.
```

Récupérer un caractère de la chaîne

- Une chaîne de caractère peut être utilisée comme un tableau de caractères. On peut accéder à ses caractères à l'aide de ses indices entre []
- Les indices dans une chaîne commencent à 0
- Le dernier indice c'est : `longueur - 1`
- **Exemple :**

```
Var s= "Bonjour";  
Var l=s.length;  
  
alert( s[0]);    // 'B'  
alert( s[3]);    // 'j'  
alert( s[l-1]) ; // 'r'
```

Affectation d'un caractère dans une chaîne

- Dans javascript , c'est pas possible de faire l'affectation d'un caractère dans la chaîne.
- **Exemple:**

```
let S= "Bonjour" ;  
S[2]="G"  
alert(S)           //il affiche Bonjour
```

- Donc même si on fait une affectation à l'indice 2 , cette affectation n'est pas prise en compte.

Extraire une sous-chaîne

- Il est utile de pouvoir extraire un morceau d'une chaîne. Pour cela on utilise la méthode `substring()`
- La méthode `substring()` attend 2 paramètres :
 - l'indice du premier caractère (inclus),
 - l'indice du dernier caractère (exclus).

Si le deuxième paramètre est **omis**, la chaîne retournée **commence à l'indice indiqué et se termine à la fin de la chaîne.**

Extraire une sous-chaîne

- Exemple :

```
var str= "Bonjour les amis !";  
var str1= str.substring(0,2);  
var str2= str.substring(3,5);  
var str3 = str.substring(6,10);  
alert(str1);    // "Bo"  
alert(str2);    // "jo "  
alertr(str3);   // "r le"
```

Récupérer le n^{ième} caractère

- La méthode charAt(n) récupère le caractère qui se trouve à la position **n**.
- Attention, le premier caractère a comme indice **0**.
Pour une chaîne initialisée par :

```
var chaine="azerty";  
  
chaine.charAt(1) ; // retourne "z".
```

Retrouver une sous-chaîne dans une chaîne

- Deux méthodes permettent de retrouver une sous-chaîne d'une chaîne. Ces méthodes retrouvent la position d'une chaîne et retournent son indice, **indexOf()** et **lastIndexOf()**.

La méthode indexOf()

- **indexOf()** : va retourner la **première position** à laquelle un caractère (ou une séquence de caractères) donné a été retrouvé dans une chaîne (on parlera de première occurrence). Si la chaîne n'est pas trouvée elle retourne -1.

```
var text = "Hello world, welcome to the universe.";
var result = text.indexOf("welcome");    // result=13
```

La méthode lastIndexOf()

- La méthode lastIndexOf(souschaine) retourne l'indice de la dernière occurrence de sous-chaine.
Si la sous-chaine n'est pas trouvée, lastIndexOf() retourne -1.
- **Exemple** :

```
var domaine="JavaScript  
  
var x=domaine.lastIndexOf("a");  
  
document.write(x); //3
```

Transformer un texte en majuscule et en minuscule

- La méthode `toUpperCase()` transforme une chaîne en majuscule
- La méthode `toLowerCase()` transforme une chaîne en minuscule.
- Ces méthodes ne modifient pas dans la chaîne courante.

```
var chaine="Ceci est un TEXTE";  
var maj=chaine.toUpperCase();  
var min=chaine.toLowerCase();  
document.write(chaine);  
document.write(maj );  
document.write(min);
```

La méthode replace()

- va nous permettre de rechercher un caractère (ou une expression) dans une chaîne de caractères et de le(la) remplacer par un(e) autre.

```
var text = "Bonjour les amis";  
var txt= text.replace("Bonjour", "Bonsoir");
```

La méthode trim()

- Cette méthode permet de supprimer les espaces à gauche et à droite de la chaîne.
- Exemple:

```
Var ch=" Bonjour mes amis ";  
  
alert(ch.length) ; // 18  
  
var x=ch.trim();  
alert(x) ; // 16
```


Découper une chaine de caractères

- La méthode **split()** permet de découper une chaine de caractère selon un séparateur, et retourne un tableau de chaines.
- **Exemple:**

```
var S="Bonjour/les/amis" ;  
var T=S.split("/") ;  
  
alert(T[0]) ;      //affiche Bonjour  
alert(T[1]) ;      //affiche les  
alert(T[2]) ;      //affiche amis
```

Convertir un tableau de chaîne en une chaîne de caractères

- La méthode **join()** crée et renvoie une nouvelle chaîne de caractères en concaténant tous les éléments d'un tableau.
- La concaténation utilise la **virgule** ou une autre chaîne, fournie en argument, comme **séparateur**.
- Exemple :

```
var tab=new Array("Pommes", "Poires", "Ananas", "Cerises");  
Var chaine=tab.join("-");
```

- Résultat: "Pommes- Poires- Ananas- Cerises"

Convertir une chaine en tableau

- On peut utiliser la méthode `split()` pour faire cette conversion. Mais il existe une autre méthode:

`Array.from(chaine)`

- Exemple :

```
var ch="Bonjour";  
  
var t=Array.from(ch);  
  
alert( t[3] ) ; // 'j'
```