

# Les tableaux en JavaScript (objet Array)

L'objet Array permet de lister une série de valeurs dans une seule variable. Les différents éléments sont identifiés par un nombre entier représentant leur position. Ce nombre est appelé un index. L'objet Array permet ainsi de créer des variables dites indicées ou indexées.

L'index commence (comme souvent en JavaScript) à la valeur 0.

## Exemple

semaine	
0	Lundi
1	Mardi
2	Mercredi
3	Jeudi
4	Vendredi
5	Samedi
6	Dimanche

La variable indexée `semaine` contient comme valeurs les différents jours de la semaine.

Ces variables indexées peuvent être identifiées en JavaScript (comme dans d'autres langages de programmation) sous le vocable de tableau.



Les tableaux en JavaScript n'ont rien en commun avec les tableaux du langage HTML. À ne pas confondre !

## 1. La définition d'un tableau

La création d'un objet de type Array s'opère à l'aide de l'opérateur `new` suivi de `Array()`.

```
var tableau = new Array();
```

Commentaires :

- En JavaScript, il faut d'abord définir le tableau avant de pouvoir en préciser le contenu.
- Il n'est pas obligatoire de définir la taille du tableau dans sa déclaration (par exemple `var tableau = new Array(7)`). La taille des tableaux est prise en charge de façon dynamique par JavaScript.
- Il est possible, à tout moment, de connaître la taille d'un tableau, en invoquant la propriété `length` de l'objet correspondant à ce tableau.

Notre tableau avec les jours de la semaine peut se définir comme suit :

```
var semaine = new Array();
```

## 2. L'initialisation d'un tableau

Pour initialiser un tableau, la syntaxe suivante est utilisée :

`tableau[i] = élément;` où *i* est un nombre compris entre 0 et la longueur du tableau -1.

Soit pour le tableau semaine :

```
var semaine = new Array();
semaine[0] = "Lundi";
semaine[1] = "Mardi";
semaine[2] = "Mercredi";
semaine[3] = "Jeudi";
semaine[4] = "Vendredi";
semaine[5] = "Samedi";
semaine[6] = "Dimanche";
```

La définition et l'initialisation peuvent se faire lors de la même instruction sous la forme suivante :

```
var semaine = new Array("Lundi" , "Mardi" , "Mercredi" , "Jeudi" ,
"Vendredi" , "Samedi" , "Dimanche");
```

### 3. L'accès aux données du tableau

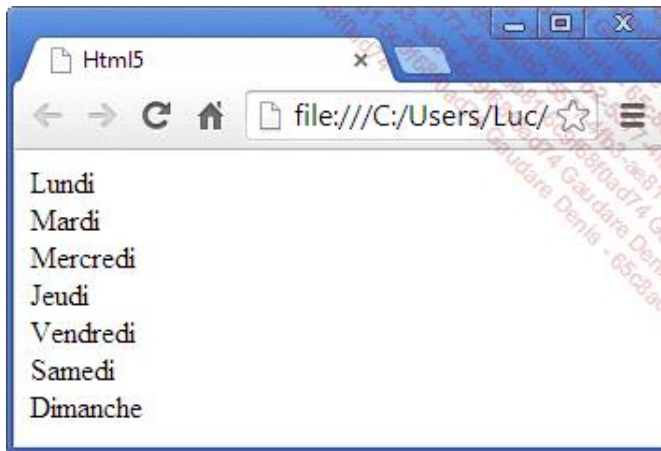
On accède à un élément du tableau par le nom du tableau et le numéro de l'indice.

Soit, dans notre exemple `semaine[0]` pour "Lundi".

#### Exemple

*Afficher les éléments du tableau semaine.*

```
<!DOCTYPE html>
<html lang="fr">
<head>
<title>JavaScript</title>
<meta charset="UTF-8">
<script>
var semaine = new Array();
semaine[0] = "Lundi";
semaine[1] = "Mardi";
semaine[2] = "Mercredi";
semaine[3] = "Jeudi";
semaine[4] = "Vendredi";
semaine[5] = "Samedi";
semaine[6] = "Dimanche";
for (i=0; i<7; i++) {
document.write(semaine[i] + "<br>");
}
</script>
</head>
<body>
</body>
</html>
```



L'affichage des jours de la semaine est réalisé par une boucle `for` (cf. ce chapitre - Les conditions et les boucles - La boucle `for`) avec `i` initialisé à 0 et comme condition `i` inférieur à 7.

➤ Lorsque la longueur du tableau est inconnue, on peut utiliser la propriété `length`. Ainsi, l'instruction `for` de l'exemple deviendrait : `for (i=0; i<semaine.length; i++);`

## 4. Les tableaux associatifs

Les tableaux associatifs sont des tableaux dont l'indice est une chaîne de caractères (et non plus un nombre). La chaîne est considérée comme l'index pour l'accès aux éléments du tableau.

```
var moteur = new Array("Google","Yahoo","Bing");
moteur["Google"]="http://www.google.fr";
moteur["Yahoo"]="http://www.yahoo.fr";
moteur["Bing"]="http://www.bing.com";
```

Notez que les guillemets entre les crochets sont de rigueur en cas d'utilisation de chaînes de caractères.

Pour accéder aux données d'un tableau associatif :

```
moteur["Google"] ou moteur[1];
```

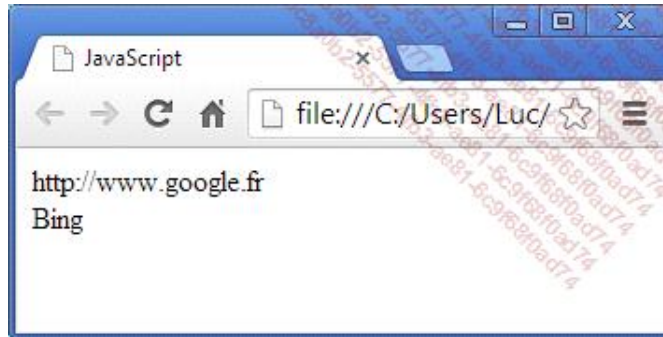
### Exemple

```
<!DOCTYPE html>
<html lang="fr">
<head>
<title>JavaScript</title>
<meta charset="UTF-8">
<script>
var moteur = new Array("Google","Yahoo","Bing");
moteur["Google"] = "http://www.google.fr";
moteur["Yahoo"]="http://www.yahoo.fr";
```

```

moteur["Bing"]="http://www.bing.com";
document.write(moteur["Google"] + "<br>");
document.write(moteur[2]);
</script>
</head>
<body>
</body>
</html>

```



## 5. Les méthodes spécifiques aux tableaux

L'objet `Array()` présente de nombreuses méthodes. En voici quelques-unes.

Élément	Description
<code>concat()</code>	Assemble deux ou plusieurs tableaux pour n'en faire qu'un seul. Cette méthode réclame en argument le tableau (ou plusieurs) qui est rajouté à la fin du tableau référencé.
<code>join()</code>	Regroupe tous les éléments du tableau dans une seule chaîne de caractères. Les différents éléments sont séparés par un caractère séparateur spécifié en argument. Par défaut, ce séparateur est une virgule.
<code>pop()</code>	Supprime le dernier élément du tableau et retourne sa valeur.
<code>reverse()</code>	Inverse l'ordre des éléments (mais ne les trie pas).
<code>shift()</code>	Supprime le premier élément du tableau et retourne sa valeur.
<code>slice()</code>	Crée un nouveau tableau à partir d'une partie d'un tableau existant. S'écrit <code>slice[début,fin]</code> .
<code>sort()</code>	Trie les éléments par ordre alphabétique (à condition qu'ils soient de même nature).

### a. Tri alphabétique d'un tableau

On applique la méthode `sort()` au tableau `semaine`.

```

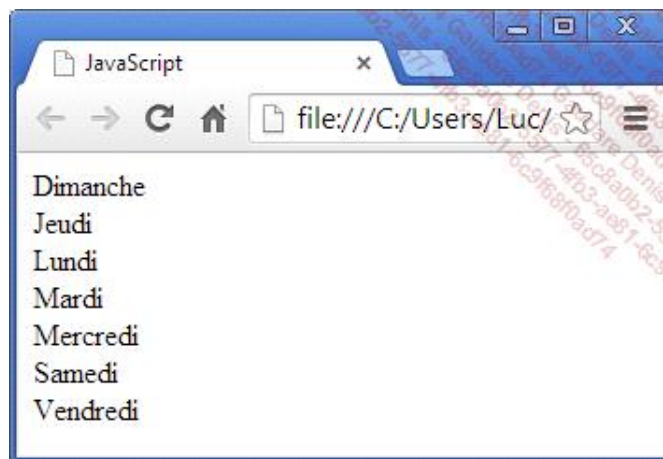
<!DOCTYPE html>
<html lang="fr">
<head>
<title>JavaScript</title>
<meta charset="UTF-8">
<script>
var semaine = new Array();
semaine[0] = "Lundi";

```

```

semaine[1] = "Mardi";
semaine[2] = "Mercredi";
semaine[3] = "Jeudi";
semaine[4] = "Vendredi";
semaine[5] = "Samedi";
semaine[6] = "Dimanche";
semaine.sort();
for (i=0; i<7; i++) {
document.write(semaine[i] + "<br>");
}
</script>
</head>
<body>
</body>
</html>

```



## b. Assemblage de tableaux

Soit deux tableaux : `semestre1` et `semestre2`. Ces deux tableaux sont regroupés dans un nouveau tableau `annee` par la méthode `concat()`.

```

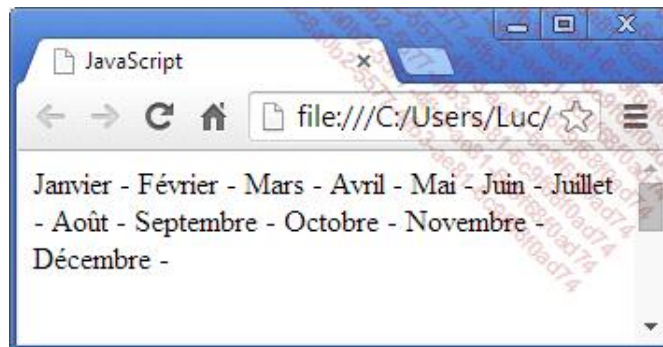
<!DOCTYPE html>
<html lang="fr">
<head>
<title>JavaScript</title>
<meta charset="UTF-8">
</head>
<body>
<script>
var semestre1 = new Array();
semestre1[0] = "Janvier";
semestre1[1] = "Février";
semestre1[2] = "Mars";
semestre1[3] = "Avril";
semestre1[4] = "Mai";
semestre1[5] = "Juin";
var semestre2 = new Array();
semestre2[0] = "Juillet";

```

```

semestre2[1] = "Août";
semestre2[2] = "Septembre";
semestre2[3] = "Octobre";
semestre2[4] = "Novembre";
semestre2[5] = "Décembre";
var annee= new Array();
annee = semestre1.concat(semestre2);
for (i=0; i<annee.length; i++){
document.write(annee[i] + " - ");
}
</script>
</body>
</html>

```



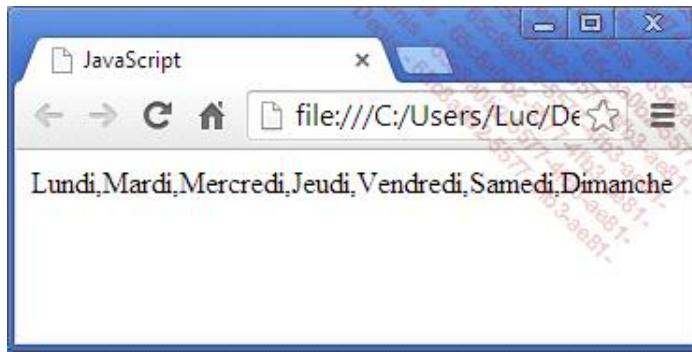
### c. Regrouper en une chaîne

Avec la méthode `join()`, regroupons les éléments du tableau `semaine` dans une chaîne de caractères.

```

<!DOCTYPE html>
<html lang="fr">
<head>
<title>JavaScript</title>
<meta charset="UTF-8">
<script type="text/JavaScript">
var semaine = new Array();
semaine[0] = "Lundi";
semaine[1] = "Mardi";
semaine[2] = "Mercredi";
semaine[3] = "Jeudi";
semaine[4] = "Vendredi";
semaine[5] = "Samedi";
semaine[6] = "Dimanche";
var join = semaine.join();
document.write(join);
</script>
</head>
<body>
</body>
</html>

```



Par défaut, le séparateur est une virgule. Un autre séparateur peut être spécifié en argument de la méthode `join()`. Soit, par exemple, `join("-")` pour un tiret de séparation.

#### d. Partie d'un tableau

Utilisation de la méthode `slice()` pour créer un nouveau tableau comportant les mois du printemps.

```
<!DOCTYPE html>
<html lang="fr">
<head>
<title>JavaScript</title>
<meta charset="UTF-8">
</head>
<body>
<script>
var annee = new Array();
annee[0] = "Janvier";
annee[1] = "Février";
annee[2] = "Mars";
annee[3] = "Avril";
annee[4] = "Mai";
annee[5] = "Juin";
annee[6] = "Juillet";
annee[7] = "Août";
annee[8] = "Septembre";
annee[9] = "Octobre";
annee[10] = "Novembre";
annee[11] = "Décembre";
var printemps = new Array();
printemps = annee.slice(3,6);
for (i=0; i<printemps.length; i++){
document.write(printemps[i] + "<br> ");
}
</script>
</body>
</html>
```

