



## Les tables indexées

---

### 1. Qu'est-ce qu'une table indexée

Une table indexée (table indexée à une dimension ou vecteur) peut être comparée à des casiers numérotés : dans la case 1, il y a quelque chose, dans la 2, autre chose .... En javascript, elle est souvent utilisée car elle permet d'obtenir plusieurs variables en se basant sur un chiffre (voir plus loin).

### 2. L'organisation d'une table

Les tables indexées ont une organisation relativement simple. Tout d'abord, une table est une variable donc porte un nom, ici on prendra *var table*. Ensuite, il s'agit d'associer un tableau à cette variable, via l'instruction :

```
var table = Array();
```

A partir de maintenant, la variable *table* est une table indexée. Pour la suite, il suffit d'attribuer une valeur pour chaque case. Les valeurs sont des variables, donc peuvent être de même type qu'une variable standard (string, number, boolean) et plusieurs types de variable peuvent être attribués dans la même table. Pour déclarer un casier d'une table, on utilise le nom de la variable suivie de [*valeur\_numérique*] comme exemple :

```
table[0] = "première variable";  
table[1] = 156;  
table[2] = false;
```

Et ainsi de suite, les tables n'ont pas de limite. A noter également qu'en JavaScript, la numérotation ne commence pas à partir de 1, mais de 0.

Une chose intéressante, et c'est ce qui fait un atout majeur des tables, on peut remplacer le chiffre entre [ ] par une variable, ainsi :

```
table[0] = "première variable";
```

Et :

```
var test = 0;  
table[test] = "première variable";
```

Reviennent exactement au même.

Une autre manière de déclarer le contenu d'une table est d'inscrire chaque case lors de la déclaration entre ( ) et de séparer les champs par une virgule. Je m'explique, on a la table suivante :

```
var newtable = Array();
newtable[0] = "première variable";
newtable[1] = "seconde variable";
newtable[2] = "troisième variable";
newtable[3] = "quatrième variable";
newtable[4] = "cinquième variable";
```

Si maintenant on écrit :

```
var newtable = Array("première variable", "seconde variable", "troisième variable",
"quatrième variable", "cinquième variable");
```

Nous aurions le même résultat, la case `newtable[0]` aura la même valeur "première variable" dans les deux cas.

### 3. Appel à une table

Maintenant que la table est créée, il suffit de l'utiliser, en prenant l'exemple de la table suivante :

```
var newtable = Array();
newtable[0] = "première variable";
newtable[1] = "seconde variable";
newtable[2] = "troisième variable";
newtable[3] = "quatrième variable";
newtable[4] = "cinquième variable";
```

Maintenant, si nous voulons utiliser la case numéro 3, et que nous voulons afficher son contenu sur la page Web, il suffira d'écrire :

```
document.write(newtable[3]);
```

Et à, il s'affichera *quatrième variable* sur la page Web.

Maintenant, nous pouvons réutiliser le contenu du tableau dans une variable. Je m'explique : prenons un nombre *y* généré au hasard entre 0 et 4 :

```
var y = Math.floor(Math.random()*5);
```

Il n'est pas forcément utile de connaître le fonctionnement de ce script. Sachez simplement qu'il génère un nombre aléatoire entre 0 et 4. Il suffit maintenant d'utiliser une variable *table2* pour utiliser le contenu du tableau. Il suffit alors de déclarer :

```
var table2 = newtable[y];
```

Sur le fond, il est vrai que cela n'est pas forcément utile de procéder ainsi, mais cela évite de gérer deux variable (*newtable* et *y*) et de n'en gérer qu'une seule à la place (*table2*). Si nous voulons écrire le contenu de la variable :

```
document.write(table2);
```

Cela affichera aléatoirement une des cases du tableau.

#### 4. Visualisation et tri du contenu d'une table

Maintenant que la table est créée, il vous est peut-être nécessaire de visualiser toute votre table. Pour cela, il existe une instruction particulièrement adaptée (elle a même du être créée uniquement pour cela) : l'instruction `.join()`. Cette instruction transforme la table en une chaîne de caractères. Reprenons notre table

```
var newtable = Array();
newtable[0] = "première variable";
newtable[1] = "seconde variable";
newtable[2] = "troisième variable";
newtable[3] = "quatrième variable";
newtable[4] = "cinquième variable";
```

Si on utilise l'instruction et que l'on affiche le résultat par exemple:

```
document.write(newtable.join());
```

On obtient :

première variable,seconde variable,troisième variable,quatrième variable,cinquième variable

Le caractère par défaut pour séparer les cases de la table est la virgule ,. Mais il est possible de changer ce caractère en l'inscrivant dans les parenthèses de l'instruction comme ceci :

```
document.write(newtable.join("<br>"));
```

On obtiendrait alors :

première variable  
seconde variable  
troisième variable  
quatrième variable  
cinquième variable

Une autre opération de visualisation intéressante serait de pouvoir trier le contenu de la table par ordre alphabétique. Eh bien rassurez vous, cette instruction existe, il s'agit de l'instruction `.sort()`. Attention : l'opération est irréversible : une fois l'instruction appliquée, il est impossible de retourner en arrière, il est alors conseillé de faire une copie de la table en utilisant une autre variable. Reprenons notre instruction :

```
newtable.sort();
```

Maintenant que la table est classée, si on l'affiche

```
document.write(newtable.join("<br>"));
```

On obtient

cinquième variable  
première variable  
quatrième variable  
seconde variable  
troisième variable