

Création d'une table en utilisant exclusivement le DOM

Création d'une table

Rappel

Tout d'abord, il faut savoir qu'une table contrairement aux pratiques trop répandues sur l'Internet ne sert pas à la présentation (-> c'est le rôle du CSS), mais plutôt à l'organisation de données de manière cohérente. Une table est composée de différentes parties chacune désignée par des balises différentes :

- la balise principale d'une table est la balise `<table>`;
- Une table a besoin d'une légende, c'est le rôle de la balise `<caption>`;
- une table en XHTML est un ensemble de lignes qui sont reprises sous la balise `<tr>`;
- Chaque ligne d'une table est composée de cases représentées généralement par la balise `<td>`. Mais lorsque cette case contient un titre on lui préférera la balise `<th>`;
- Enfin pour montrer la différence entre les données d'une table et ses entêtes, on utilise les balises `<thead>`, `<tfoot>` pour l'entête et le bas de table, et `<tbody>` pour le contenu à proprement parler de la `<table>`;

ceci est un tableau statique

COLONNE 1	COLONNE 2	COLONNE 3	COLONNE 4
row 1, cols 1	row 1, cols 2	row 1, cols 3	row 1, cols 4
row 2, cols 1	row 2, cols 2	row 2, cols 3	row 2, cols 4
row 3, cols 1	row 3, cols 2	row 3, cols 3	row 3, cols 4
row 4, cols 1	row 4, cols 2	row 4, cols 3	row 4, cols 4
COLONNE 1	COLONNE 2	COLONNE 3	COLONNE 4

Comme vous le constater, la création d'une table avec le DOM demandera la création ordonnée de plusieurs balises et leur intégration progressive dans la balise générale `<table>`.

Création d'une table

C'est pourquoi j'ai mis sur pied une fonction `createTable` en javascript qui accepte 2 valeurs, `nbrows` correspondant au nombre de lignes et `nbccls` correspondant au nombre de colonnes que l'on veut voir dans notre table. Voici la méthode de création de cette fonction.

```

1 function createTable(nbrows,nbccls){
2   //1) setting tbody node
3   var tbody = document.createElement( "TBODY" );
4   for ( var i = 0; i < nbrows; i++){

```

[Top](#)

```

5   var tr = document.createElement( "TR" );
6   for ( var j = 0; j < nbcols; j++){
7       var td = document.createElement( "TD" );
8       var td_text = "rows " +eval(i+1)+ ", cols " +eval(j+1);
9       td_text = document.createTextNode(td_text);
10      td.appendChild(td_text);
11      tr.appendChild(td);
12  }
13  tbody.appendChild(tr);
14  }
15
16
17  //2) setting thead and tfoot
18  var thead = document.createElement( "THEAD" );
19  var tfoot = document.createElement( "TFEOT" );
20  var trTitle = document.createElement( "TR" );
21  for ( var j = 0; j < nbcols; j++){
22      var th = document.createElement( "TH" );
23      th.appendChild(document.createTextNode( "colonne " + j));
24      trTitle.appendChild(th);
25  }
26  thead.appendChild(trTitle.cloneNode( true ));
27  tfoot.appendChild(trTitle.cloneNode( true ));
28
29
30  //3) setting caption
31  var caption = document.createElement( "CAPTION" );
32  var caption_text = "ceci est un tableau de " +nbrows "
33  caption_text += " lignes et de " +nbcols + " colonnes ";
34  caption_text += " publié à l'aide du DOM ";
35  caption_text = document.createTextNode(caption_text);
36  caption.appendChild(caption_text);
37
38  //4) setting table
39  var table = document.createElement( "TABLE" );
40  table.appendChild(caption);
41  table.appendChild(thead);
42  table.appendChild(tfoot);
43  table.appendChild(tbody);
44
45
46  //5) appending table to the document tree
47  document.getElementById( "content").appendChild(table);
48  }

```

1. En utilisant `createTextNode`, `createElement`, `appendChild` ainsi que 2 boucles `for` de javascript nous créons successivement un noeud `tbody` ainsi que les `nbrows <tr>` et les `ncols <td>` de la partie `<tbody>` de notre future table.

Top

2. En utilisant la même méthode nous créons l'unique `<tr>` qui doit contenir les `nbrows` `<th>` de la table. Comme le contenu de `<thead>`, `<tfoot>` et identique nous utilisons `cloneNode` pour nous faciliter le travail.
3. Enfin nous créons la partie `caption` de la table, en utilisant `createTextNode` et `createElement`.
4. Maintenant que toutes les parties de la table sont créées il faut créer un noeud `<table>` et d'agencer nos différents noeuds soit avec `insertBefore`, soit avec `appendChild`, cela dépend de votre humeur...
5. Finalement vous pouvez publier votre table dans le document XHTML là où vous le désirez !!!!

L'utilisation de cette fonction me permet de créer des tables aux caractéristiques suivantes par exemple :

- 4 lignes et 4 colonnes
- 5 lignes et 2 colonnes
- 2 lignes et 5 colonnes

Effacer une table

Mais comme vous le remarquer tout de suite, nous ne faisons que créer des tables et rapidement elle encombre votre navigateur il faudrait donc mettre en place un moyen pour les effacer, c'est le but de la fonction `removeTable` qui va tout simplement enlever aller enlever le noeud ajouter si celui-ci est un noeud de type balise et de nom table.

```

1  /* detect lastChild of #content and delete it if it is a table
2  function removeTable() {
3      var content = document.getElementById( "content" );
4      if (content.lastChild.nodeName == "TABLE" )
5          content.removeChild( content.lastChild );
6  }
```

En utilisant cette fonction nous pouvons donc effacer une table. Mais, le plus simple ne serait-il pas de contrôler la présence ou non d'une table avant d'en créer une autre ainsi le contrôle serait accompli par notre script et plus par nous...

Affinement de nos script

Une manière de le faire serait d'utiliser l'`id` de la table créée, comme `createNewTable` qui nous permet de créer plus "intelligemment" les tables suivantes :

```

1  /* merge the 2 previous functions to allow the presence of
2  /* a maximum one table at a time
3  function createNewTable(rows,cols){
4      var content = document.getElementById( "content" );
5      if (content.lastChild.id == "newTable" ) removeTable();
6  }
```

Top

```
7 createTable(rows,cols);  
8 if (content.lastChild.nodeName == "TABLE" )  
9 content.lastChild.setAttribute( "id" , "newTable" );  
10 }
```

- 4 lignes et 4 colonnes
- 5 lignes et 2 colonnes
- 2 lignes et 5 colonnes

Sommaire du Tutorial

1. DOM : Introduction
2. Qu'est-ce que le DOM ?
3. Parcourir un document
4. Propriétés d'un node
5. getElementById
6. getElementsByName
7. getElementsByTagName
8. Les méthodes relationnelles
9. spécificité des navigateurs actuels
10. Conclusion Transitoire
11. Modifier un document
12. les attributs d'une balise
13. Modifier un noeud texte
14. DOM et Cascading StyleSheet
15. Création d'un élément
16. Modifier l'arbre d'un document
17. Création d'une table
18. DOM : Conclusion