Génération de tables de multiplication

Quand on change de table de multiplication, on veut maintenant essayer de mettre la page à jour plutôt que de recalculer entièrement la page.

Génération directe et mise à jour du DOM

Génération directe du DOM

Récupérez le fichier HTML `tablemultdom.html`.

On veut générer le fragment de document qui contient le DOM de la table de multiplication sous la forme d'une liste : le titre `<p>Table de multiplication de <span class="num">5</span></p>` puis la liste non numérotée `<ul>` contenant les lignes de la table de multiplication
`<li><span class="lig">1</span> * <span class="num">5</span> = <span class="result">5</span></li>`.

Pour faciliter l'écriture du code, **écrivez** dans le fichier `domutils.js` le module `DOMUtils` qui exporte la fonction `removeChildren(n)` qui enlève les nœuds enfants de l'élément `n` et la fonction `createSpan(className, content)` qui crée un élément `<span>` de classe `className` et contenant le texte `content`.

Dans le fichier `tablemultiplicationdom.js` **créez** le module `TableMultiplication`. **Écrivez** la fonction `DOMLigneTableMultiplicationListe(n,numligne)` qui crée le DOM de la ligne `numligne` de la table de multiplication de `n` et renvoie l'élément `<li>` correspondant.

Écrivez et exportez la fonction `DOMTableMultiplicationListe(n, nblignes)` qui crée le DOM de la table de multiplication de `n` contenant les `nblignes` premières lignes et renvoie un fragment de document contenant les éléments `<p>` et `<ul>` correspondants.

Écrivez dans le fichier `tablemultiplicationdominit.js` le code suivant :

```
(function() {
function init(event) { ecrireTable(5); }
window.addEventListener("load", init, false);
})();
```

Ajoutez-y la fonction `ecrireTable(n)` qui remplace le contenu de l'élément `<div id="tablemult">` par le DOM de la table de multiplication de `n`. **Testez**.

On veut maintenant générer le fragment de document qui contient le DOM de la table de multiplication sous la forme d'une table `<table>` : le titre de la table dans la première ligne `<tr><th colspan="5">Table de multiplication de <span class="num">n</span></th></tr>`, puis les lignes de la table
`<tr><td class="num">n</td><td>*</td>`
`<td class="lig">numligne</td><td>=</td>`
`<td class="result">n*numligne</td></tr>`.

Ajoutez la fonction `DOMLigneMultiplicationTable(n,numligne)` qui crée le DOM de la ligne `numligne` de la table de multiplication de `n` et renvoie l'élément `<tr>` correspondant.

Ajoutez et exportez la fonction `DOMTableMultiplicationTable(n, nblignes)` (dans l'espace de noms `MULT`) qui crée le DOM de la table de multiplication de `n` avec les `nblignes` premières lignes et renvoie un fragment de document contenant l'élément `<table>` correspondant.

Pour pouvoir facilement changer entre la liste et le tableau, faites pointer une variable `DOMTableMultiplication` sur `DOMTableMultiplicationListe` et modifiez `ecrireTable` en conséquence (pour qu'elle utilise `DOMTableMultiplication`). **Testez**. Faites maintenant pointer `tableMultiplicationDOM` sur `DOMTableMultiplicationTable` et **testez**.

Mise à jour du DOM

Plutôt que de recréer à chaque fois entièrement le DOM de la table de multiplication, on voudrait juste faire les mises à jour des contenus qui changent quand on change le numéro `n` de la table : les éléments de classe `num` qui contiennent `n`, les éléments de classe `lig` qui contiennent le numéro de ligne et les éléments de classe `result` qui contiennent le résultat de la multiplication.

Ajoutez et exportez dans le module `TableMultiplication` la fonction `MajDOMMultiplication(etable,n)` qui fait ces mises à jour dans l'élément `etable` qui contient déjà une table de multiplication.

Ajoutez dans `tablemultiplicationdominit.js` la fonction `majTable` qui met à jour la table de multiplication contenue dans l'élément `<div id="tablemult">`. **Testez**.

Il faut maintenant faire en sorte que `ecrireTable` crée le DOM de la table la première fois qu'on l'appelle, puis fasse la mise à jour les fois suivantes. **Renommez** `ecrireTable` en `creerTable`. Les fonctions sont des objets comme les autres et peuvent donc avoir des propriétés. **Écrivez** la fonction `ecrireTable(n)` qui teste sa propriété `_first` pour savoir si c'est la première fois qu'elle est appelée et agit en conséquence. **Ajoutez** et initialisez cette propriété. **Testez**.

C'est un peu dommage de devoir tout le temps tester si c'est la première fois ou pas. On veut faire en sorte que `ecrireTable` exécute `creerTable` la première fois qu'elle est appelée, puis pointe ensuite directement sur `updateTable`. **Ré-écrivez** la fonction `ecrireTable` pour qu'elle fasse justement ça. **Testez**.

Génération du DOM d'après un format

On veut créer la table de multiplication à partir d'un format comme dans le TD précédent, mais on veut laisser le concepteur de la page HTML indiquer le format à utiliser. Pour cela le concepteur écrit chaque donnée du format (début, ligne, fin et caractère de remplacement) dans des commentaires mis dans l'élément `<div id="tablemult">`. On ré-utilise pour cela les « classes » `LigneATrou` et `TableMultiplicationFormat` (écrites dans les fichiers `ligneatrou.js` et `tableformat.js`). **Ajoutez** et exportez dans le module `DOMUtils` la fonction `comments(e)` qui renvoie un tableau contenant les nœuds commentaires enfants du nœud `e`. **Rajoutez** dans `tableformat.js` la fonction `formatAPartirCom(e)` qui renvoie un objet format (c.-à-d. avec les propriétés `debut`, `ligne`, `fin` et `trou`) à partir des commentaires contenus dans le nœud `e`. **Copiez** `tableinitdom.js` en `tableinitformatdom.js`. **Modifiez** `creerTable` pour qu'elle crée la table de multiplication d'après le format écrit dans les commentaires de l'élément `<div id="tablemult">`. **Copiez** `tablemultdom.html` en `tablemultformatdom.html`, adaptez les fichiers javascript inclus et mettez le format de la table dans les commentaires. **Testez**.