

Génération de tables de multiplication

On veut créer une page web affichant une table de multiplication pour pouvoir l'apprendre. On veut pouvoir faire cet affichage sous différentes formes : avec une liste, dans un tableau ...

Génération Directe

On veut générer une table de multiplication sous la forme d'une liste :

Table de multiplication de 5

- $1 * 5 = 5$
- $2 * 5 = 10$
- ...

On peut changer le contenu d'un élément HTML en copiant dans sa propriété `innerHTML` une chaîne de caractères contenant le contenu HTML écrit avec des balises. On veut ici créer la chaîne de caractères représentant la table de multiplication : un paragraphe contenant le titre `<p>Table de multiplication de` `5</p>` et une liste non numérotée `` contenant les lignes de la table de multiplication `1 * 5 = 5`. On constate que ce texte HTML peut être décomposé en un début, des lignes et une fin.

Récupérez la page HTML `tablemulthtml.html` et écrivez dans le fichier `tablemulthtml.js` le module `TableMultiplication` qui contient les fonctions suivantes.

Écrivez la fonction `tableMultiplicationHTMLDebut(n)` qui renvoie le code HTML du début de la table de multiplication de `n`. **Testez** (en utilisant la console). **Écrivez** la fonction `tableMultiplicationHTMLLigne(n,numligne)` qui renvoie le code HTML de la ligne numéro `numligne` de la table de multiplication de `n`. **Testez**. **Écrivez** la fonction `tableMultiplicationHTMLFin(n)` qui renvoie le code HTML de la fin de la table de multiplication de `n`. **Testez**. **Écrivez** la fonction `tableMultiplicationHTML(n,nbignes)` qui renvoie le code HTML complet de la table de multiplication de `n` contenant les lignes 1 à `nbignes` (sans utiliser la concaténation directe de chaînes avec `+` dans cette fonction). **Testez. Faites en sorte que** seule la fonction `tableMultiplicationHTML` soit exportée par le module.

Vous pouvez regarder le fichier `tableinithtml.js` :

```
var htmlTable = TableMultiplication.tableMultiplicationHTML;

function ecrireTableMultiplication(n,nbignes) {
    document.getElementById("tablemult").innerHTML =
        htmlTable(n,nbignes);
}

(function() {
    function init(event) { ecrireTableMultiplication(5,12); }
    window.addEventListener("load", init, false);
})();
```

La variable `htmlTable` pointe sur une fonction générant le code HTML de la table de multiplication de `n` contenant `nbignes` lignes et la fonction `ecrireTable(n,nbignes)` insère le code HTML généré par `htmlTable` dans le `<div id="tablemut">` de la page HTML et la fonction anonyme permet d'exécuter `ecrireTable(5,12)` au chargement de la page. Vous pouvez dans la console appeler la fonction `ecrireTable` pour afficher une autre table. **Testez. Faites** en sorte que la fonction `ecrireTable` utilise 12 lignes quand `nbignes` n'est pas donné. **Testez**.

Génération d'après un format

On veut redonner la main au développeur HTML pour représenter la table.

Ligne de format à trous

Le développeur HTML va donner des lignes de format à trous comme chaînes de format, c'est-à-dire une chaîne de caractères contenant des caractères à remplacer par du contenu. Par exemple, étant donnée la ligne de format `"Aujourd'hui % il fait un temps %."` et le caractère de trou `"%"`, si on donne un tableau de contenus de trous `["13 mars","dégagé"]`, on aura comme résultat la chaîne `"Aujourd'hui 13 mars il fait un temps dégagé."`

Pour faire cela en minimisant les concaténations successives de chaînes, la ligne à trous stockera dans un tableau `_ligne` le découpage de la ligne de format suivant le caractère trou avec des cases vides entre pour qu'elles puissent être remplies avec du contenu soit, avec l'exemple précédents le tableau `["Aujourd'hui "," " il fait un temps ","."].`

Écrivez dans le fichier `ligneatrou.js` le module `TxtUtils` qui exporte la « classe » `LigneATrous` : le constructeur prend en paramètres une ligne de format et un caractère trou et calcule le tableau correspondant, la méthode `ligneRemplie` renvoie une chaîne contenant la ligne où les trous sont remplis avec les valeurs du

tableau de contenus de trous passé en paramètre.

Testez : écrivez une procédure de test qui affiche sur la console quelques résultats de lignes à trous remplies et exécutez la, copiez `tablemulthtml.html` en `tablemultformat.html` et faites-lui charger `ligneatrou.js`.

Génération de table d'après un format

On veut générer différents types de tables de multiplications à partir de différents formats. Dans le fichier `tablemultformat.js` (à créer) rajoutez dans le module `TableMultiplication` la « classe » `TableMultiplicationFormat` : le constructeur prend en paramètre un objet avec des propriétés `debut`, `ligne`, `fin` et `trou` contenant les lignes de format pour le début, chaque ligne et la fin de la table de multiplication ainsi que le caractère indiquant les trous ; la table de multiplication gardera en mémoire les lignes à trous correspondantes (de début, de ligne et de fin) ; la méthode `HTMLTable(n,nblignes)` renvoie la chaîne représentant la table de multiplication de `n` (contenant les lignes 1 à `nblignes`) : elle est composée du début où le trou est remplacé par `n`, puis de chaque ligne de la table où les trous sont remplacés par le numéro de ligne, `n` et le résultat de la multiplication, puis de la ligne de fin où le trou est remplacé par `n`.

Copiez le fichier `tableinithtml.js` en le fichier `tableinitformat.js`, créez un objet `tableListeTrou` qui a une propriété `_table` qui représente un format de table qui génère la table de multiplication dans une liste `` en utilisant les lignes à trou pour donner le format et qui a une méthode `HTMLTable(n,nblignes)` qui renvoie le code HTML de la table. **Faites pointer** `htmlTable` sur une fonction qui renvoie le code HTML de la table en utilisant `tableListeTrou`. **Copiez** le fichier `tablemulthtml.html` en le fichier `tablemultformat.html` et modifiez les fichiers javascript utilisés. **Testez**.

Faites le même genre de chose pour générer la table de multiplication dans un tableau `<table>` en utilisant les lignes à trou pour donner le format : le titre de la table est dans la première ligne dans un `<th>`, puis pour chaque ligne chaque élément d'une ligne sera dans une case, le numéro de ligne, le `n` et le résultat étant respectivement de classes `lig`, `num` et `result`. **Testez**.