

## 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 ...

### 1 Génération Directe

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

Table de multiplication de 5

- $1 \times 5 = 5$
- $2 \times 5 = 10$
- ...

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` `<span class="num">5</span></p>` et une 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>`. On constate que ce texte HTML peut être décomposé en un début, des lignes et une fin.

**Récupérez** la page HTML `tablemulhtml.html` et écrivez les fonctions dans le fichier `tablemultiplication.js`. **Faites** charger le script `tablemultiplication.js` par la page `tablemulhtml.html`.

**Écrivez** la fonction `debutHTMLTableMultiplication(n)` qui renvoie le code HTML du début de la table de multiplication de `n` (vous pouvez utiliser une chaîne interpolée). **Testez** (appelez la fonction dans la console du navigateur). **Écrivez** la fonction `ligneHTMLTableMultiplication(n,numligne)` qui renvoie le code HTML de la ligne numéro `numligne` de la table de multiplication de `n`. **Testez**. **Écrivez** la fonction `finHTMLTableMultiplication(n)` qui renvoie le code HTML de la fin de la table de multiplication de `n`. **Testez**. **Écrivez** (sans utiliser la concaténation directe de chaînes avec `+` dans cette fonction : créez un tableau de chaînes à joindre ensuite avec `join`) la fonction `HTMLTableMultiplication(n,nblignes)` qui renvoie le code HTML complet de la table de multiplication de `n` contenant les lignes 1 à `nblignes` (valant 12 par défaut). **Testez**.

Vous pouvez regarder le fichier `tableinithtml.js` :

```
let htmlTable = HTMLTableMultiplication;
function ecrireTableMultiplication(n,nblignes) {
    document.getElementById("tablemult").innerHTML =
        htmlTable(n,nblignes);
}

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

La variable `htmlTable` pointe sur la fonction générant le code HTML de la table de multiplication de `n` contenant `nblignes` lignes que vous avez écrites et la fonction `ecrireTableMultiplication(n,nblignes)` insère le code HTML généré par `htmlTable` dans le `<div id="tablemult">` de la page HTML et `ecrireTableMultiplication(5,12)` est exécuté au chargement de la page. **Faites** charger ce script par la page HTML et **testez**. Vous pouvez dans la console appeler la fonction `ecrireTableMultiplication` pour afficher une autre table. **Testez**.

### 2 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."`

La ligne à trous stocke dans le tableau `_ligne` le découpage de la ligne de format suivant le caractère trou (dans l'exemple précédent le tableau vaudra `["Aujourd'hui ", " il fait un temps ", "."]`).

**Écrivez** dans le fichier `ligneatrou.js` la « classe » `LigneATrous` : le constructeur prend en paramètres une ligne de format et un caractère trou et calcule et stocke le découpage de la ligne `_ligne` correspondant, la méthode `ligneRemplie(contenus)` renvoie une chaîne contenant la ligne où les trous sont remplis avec les valeurs du tableau `contenus` (calculer le tableau contenant alternativement les chaînes de la ligne à trou et du remplissage et joindre à la fin).

**Copiez** `tablemulhtml.html` en `tablemultformat.html` et ne lui faites charger que `ligneatrou.js`. **Testez** : ajoutez dans `ligneatrou.js` une

procédure de test qui affiche sur la console quelques résultats de lignes à trous remplies et exécutez la dans la console du navigateur.

### Génération de table d'après un format

On veut générer différents types de tables de multiplications à partir de différents formats. **Écrivez** dans le fichier `tablemultiplicationformat.js` la « classe » `FormatTableMultiplication` : 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, le numéro de la table de multiplication `n` et le résultat de la multiplication, puis de la ligne de fin où le trou est remplacé par `n`. **Ajoutez-y** la fonction `HTMLTableListeFormat(n, nblignes)` qui génère la table de multiplication dans une liste `<ul>` en utilisant un objet qui représente un format de table à partir de lignes à trou.

**Copiez** le fichier `tableinithtml.js` en le fichier `tableinitformat.js`. **Faites pointer** `htmlTable` sur la fonction `HTMLTableListeFormat`. **Faites charger** `tableinitformat.js` par `tablemultformat.html` et **Testez**.