

Exercice à développer pour la semaine du 11 septembre 2023 :

**Le jeu du devin**

On veut écrire un petit jeu qui consiste à tirer au sort un nombre Mystère compris entre 1 et 100. Le joueur devra deviner ce nombre. A chaque proposition suivant la circonstance, on affichera l'un des messages résultant suivants :

Trop Grand (en b bleu),

Trop Petit » (en bleu) ou

Gagné » (en vert).

On peut perdre si le nombre n'a pas été deviné lors de la 6ème tentative (6 coups au maximum par partie). Auquel, cas on affichera « Perdu » (en rouge) et la solution qui était attendue. Il sera toujours possible pour le joueur de recommencer une partie.

Le joueur a comme interface d'entrée une zone de saisie d'un nombre, une zone d'affichage pour l'affichage du résultat et un bouton initialement noté "**Jouer**". En cliquant dessus, le script d'exécution est invoqué pour analyser la valeur saisie, tenant compte des cas normaux et des cas d'erreurs :

La programmation à développer :

- Si le joueur saisi une chaine vide ou si la chaine ne représente pas un entier, un message d'erreur s'affiche (en noir) mais on ne comptera pas le coup comme valide.
- Si le joueur saisit un nombre entier hors de l'intervalle [1,100], alors tant pis pour lui : ça comptera pour une tentative infructueuse.
- Lorsque le joueur finit une partie, perdu ou gagné, on lui proposera de rejouer en cliquant sur l'unique bouton. Il faudra donc penser à changer temporairement l'intitulé du bouton en "**rejouer ?**"  
Rejouer signifie réinitialiser le compteur de coup tandis que le bouton reprendra son intitulé initial.

Le fichier **devin.html** contiendra la description HTML du jeu. Sous la forme de 3 balises : un input, un bouton et une zone d'affichage ! Rien de plus !

Le script **devin.js** associé à la page HTML contiendra la déclaration des événements, c'est pourquoi, aucun code JS n'est autorisé dans cette page (notamment via l'attribut `onclick`). Il faudra exploiter les méthodes comme

```
document.getElementById(),
```

```
window.addEventListener(),
```

```
window.removeEventListener().
```

Pour modifier/insérer du texte contenu dans une balise on peut utiliser la propriété `textContent` de l'objet DOM correspondant. Si par contre, le texte contient (entre autres) des balises HTML alors la propriété à exploiter est `innerHTML`. Les propriétés de styles d'un objet DOM sont référencées dans l'objet de l'objet via la propriété `style` de l'objet. La valeur de saisie d'une balise `<input>` est représentée dans l'objet DOM associé par la propriété `value` de cet objet. Pour information, la valeur défaut apparaissant dans la description HTM d'une balise `<input>` est stockée dans la propriété `default-value`.

Voici à quoi devrait ressembler certaines étapes du jeu (au nombre de tentatives près, à mettre à jour).

Affichage pour réaliser une nouvelle tentative :

### **Jeu du Devin : Deviner un nombre entier compris entre 1 et 100**

(On a le droit à 6 tentatives maximum)

Suite à une proposition de nombre, ici 50, on peut avoir cela :

### **Jeu du Devin : Deviner un nombre entier compris entre 1 et 100**

(On a le droit à 6 tentatives maximum)

50

[1] C'est plus

Suite à une proposition gagnante, on peut avoir cela :

### **Jeu du Devin : Deviner un nombre entier compris entre 1 et 100**

(On a le droit à 6 tentatives maximum)

87

[5] C'est gagné ! Le nombre Mystère était bien 87

Lors d'une 6<sup>ème</sup> tentative erronée, on peut avoir cela :

### **Jeu du Devin : Deviner un nombre entier compris entre 1 et 100**

(On a le droit à 6 tentatives maximum)

50

[6] C'est perdu ! Le nombre Mystère était 7

La saisie d'une chaîne contenant un caractère autre que numérique conduit à cet affichage :

### **Jeu du Devin : Deviner un nombre entier compris entre 1 et 100**

(On a le droit à 6 tentatives maximum)

Babar

[1] On a dit un nombre entier !!!

Si on rejoue à la fin d'une partie, on repart du début !