

Coding Dojo :

Coding Dojo : Exam

- a) Télécharger les fichiers de l'exam et modifier exam.js.
- b) Modifier `plusPetitElement()` pour que la fonction retourne le plus petit élément du tableau passé en paramètre. Toutes les valeurs sont inférieures à **1 000 000**.
- c) Modifier `plusGrandElement()` pour que la fonction retourne le plus grand élément du tableau passé en paramètre. Toutes les valeurs sont supérieures à **900 000**.
- d) Modifier `chercherElement()` pour que la fonction retourne les éléments du tableau `search` qui sont dans le tableau `elements`. Utiliser la fonction `ajouteSearched()` pour mettre les éléments trouvés.

```
ajouteSearched(42);  
// Ajoute 42 dans le tableau
```

- e) Modifier `ajouterElement()` pour que la fonction additionne les éléments du tableau passé en paramètre.
- f) Modifier `compterText()` pour que la fonction compte le nombre de "i" de la variable passée en paramètre.
- g) Avec SublimeText, pour ne pas avoir de barre de défilement horizontale, dans la barre d'outils faites "**View > Word Wrap**".
- h) Ne modifier aucune autre fonction et il n'est pas utilisé de créer de nouvelle fonction.

Coding Dojo : Démineur

a) Télécharger les fichiers du démineur et modifier uniquement minesweeper.js.

b) Démineur :

- But du jeu : Découvrir toutes les cases où il n'y a pas de mines.
- Comment Jouer : Chaque case contient soit une mine, soit le nombre de mine autour de cette case, soit du vide. Il est possible de mettre un drapeau sur une case si l'on sait qu'il y a une mine.
- Condition :
 - ▶ La variable minesweeper contient les informations de la grille. On accède à une case avec minesweeper[y][x].
 - ▶ Si la case est une mine, minesweeper[y][x] est égal à 'M' (pour Mine).
 - ▶ Si la case est vide, minesweeper[y][x] est égal à 'E' (pour Empty).
 - ▶ Si la case a une mine autour d'elle, minesweeper[y][x] est égal au nombre de mine autour d'elle. Au maximum il ne peut y avoir que 8 mines autour d'une case.
 - ▶ Il faut appuyer sur les boutons Facile, Moyen ou Difficile pour générer une nouvelle grille.
 - ▶ Le bouton "Compléter la grille" lance la fonction `completerGrille()`.
 - ▶ Le bouton "Dessiner la grille" lance la fonction `dessinerGrille()`.
 - ▶ Raccourcis clavier :
 - F : active/désactive le mode drapeau
 - D : active/désactive le mode debug

- c) Modifier `completerGrille()` pour que la fonction complète automatiquement la grille. C'est à dire que vous devrez mettre 'E' ou les chiffres de 1 à 8 dans toutes les cases de `minesweeper[y][x]`.
- Inutile d'appeler les fonctions `dessinerCaseVide(x, y)` et `dessinerNbMine(x, y)` dans cette fonction, `dessineGrille()` est appelé à la fin.
- d) Modifier `play(x, y)` pour que qu'a chaque fois qu'on clique sur une case, celle ci dessine soit du vide, soit le nombre de mine. Utiliser les fonctions `dessinerCaseVide(x, y)` et `dessinerNbMine(x, y)` pour dessiner l'information.
- L'exemple avec `dessinerMineExplose(x, y)`, affiche une case rouge si `minesweeper[y][x]` est égal à 'M', c'est à dire si on clique sur une mine.
 - Exemple : Si la case est vide, `minesweeper[y][x]` devra être égal 'E' et il faudra lancer la fonction `dessinerCaseVide(x, y)`

```
minesweeper[y][x] = 'E';  
dessinerCaseVide(x, y);
```

- e) Modifier `is_win()` pour que l'on gagne si on a découvert toutes les cases sans mine.

Coding Dojo : Démineur

f) Exemple :

1			1								1			2	
2	3	3	1			1	2	2	1		1	3		2	
1		1				1			3	1		1	1	1	
1	1	1				1	3			2					
	1	2	3	2	1		1	3		3	1	2	1	1	
2	4				1	1	1	2	1	2		2		3	2
			5	3	2	2		2		1	1	2	2		
3		3	2		2	3		2	1	1	1		1	3	
2	2	2	1	1	3		3	1	1		3	2	2	2	2
1		1			2		2		1	2			2		1

g) BONUS : Quand on clique sur une case vide ('E'), dévoiler automatiquement les cases vides voisines et les cases chiffres proches.

	2	1	1												
1	1		1												
		1	2												
	1	2													
	1														
	1														