

Apprentissage de tables multiplication

On veut faire une petite application permettant aux écoliers de réviser leurs tables de multiplications et de vérifier s'ils les connaissent. Récupérez le fichier `tablemult.html` qui contient la page HTML correspondante.

L'application va être répartie dans trois fichiers : le fichier `tablemultview.js` définira un module `View` contenant la vue, c'est-à-dire les données et les méthodes permettant de manipuler l'aspect de la page ; le fichier `tablemultdata.js` définira un module `Data` contenant les données, c'est-à-dire les données et les méthodes permettant de faire les calculs ; enfin le fichier `tablemultcontroller.js` définira un module `Controller` contenant le contrôleur, c'est-à-dire les données et les méthodes permettant de gérer les interactions avec l'utilisateur. **Écrivez** ces trois fichiers : chaque module doit contenir une fonction `onLoad` qui servira à initialiser le module et faire en sorte qu'elle soit appelée juste après le chargement de la page et doit contenir une fonction `unload` qui servira à faire le ménage juste avant de quitter la page et faire en sorte qu'elle soit justement appelée juste avant de quitter la page.

Initialisation et mise à jour du numéro

Il faut mettre le numéro de la table dans tous les éléments de classe `num`. **Ajoutez** dans le module `View` une variable `numClassElements`, tableau qui doit contenir tous les éléments de classe `num` et initialisez la dans la fonction `onLoad()`. **Ajoutez** et exportez la fonction `writeTableNumber(num)` qui écrit le numéro `num` dans tous ces éléments. **Testez** en appelant `writeTableNumber` dans la console.

L'utilisateur peut choisir le numéro de la table. Pour cela il y a une sélection `<select id="selectnum">` contenant les numéros comme options. **Ajoutez** dans le module `View` une variable `selectNumberElement` pointant dessus (à initialiser dans `onLoad`) et une méthode `selectedTableNumber()` exportée qui renvoie le numéro sélectionné. **Ajoutez** dans le module `Data` une variable `tableNumber` contenant le numéro de la table de multiplication actuelle ainsi qu'une fonction exportée `changeTableNumber(num)` qui met à jour ce numéro.

Ajoutez dans le module `Controller` la fonction `nouvelleTable()` qui fait que la vue et les données soit ré-initialisées avec la table de multiplication du numéro sélectionné, c'est-à-dire que ce numéro soit affiché dans la vue et soit celui de la table actuelle dans les données **Modifiez** la fonction `onLoad` du module

`Controller` pour que la table manipulée soit celle du numéro sélectionné initialement dans la page. **Testez** en changeant l'attribut `selected="selected"` dans des `<options>` différentes.

Ajoutez dans le module `Controller` le gestionnaire d'événements `onChangeTableNumber` qui sera appelé quand l'utilisateur choisira un nouveau numéro et qui ré-initialise la table de multiplication avec ce numéro. Faites en sorte que cette fonction soit attachée à l'élément sélection au chargement de la page et détachée quand on la quitte. **Testez**.

Calcul et affichage des résultats

Pour faire les calculs de multiplication de l'interface, **ajoutez** et exportez dans le module `Data` la « classe » `MultiplicationTable` représentant une table de multiplication : son constructeur prend en paramètres le numéro `tableNumber` de la table ainsi que le nombre de résultats `nbresults` à calculer (10 par défaut) ; l'objet stocke ces données (`_nbresults` et `_tableNumber`) ainsi que le tableau `_values` contenant les résultats des multiplications du numéro par la ligne correspondante (pour éviter de les recalculer) ; la classe a une méthode `number()` qui renvoie le numéro de la table, une méthode `value(i)` qui renvoie la valeur numéro `i` et une méthode `values()` qui renvoie un tableau des valeurs. **Testez**.

Remplacez dans les données la variable `tableNumber` qui stockait le numéro de la table de multiplication par la variable `currentMultiplicationTable` qui stocke maintenant la table de multiplication et **modifiez** la fonction `changeTableNumber` en conséquence. **Ajoutez** et exportez les fonctions `tableNumber()` qui renvoie le numéro de la table, `tableValue(i)` qui renvoie la valeur numéro `i` et la fonction `tableValues()` qui renvoie le tableau des valeurs.

Il faut afficher les résultats dans les champs texte de classe `inputresult`. **Ajoutez** dans la vue une variable `inputresultClassElements`, tableau qui contient ces éléments et doit être correctement initialisé au chargement de la page, et **ajoutez** et exportez une fonction `displayResults(results)` qui affiche en lecture seule dans ces champs texte les valeurs du tableau `results`. **Testez**. **Ajoutez** et exportez une méthode `eraseResult()` qui efface ces champs texte et les remet en écriture. **Faites** en sorte que les champs texte soient effacés quand on utilise une nouvelle table. **Testez**.

On a mis un bouton `Résultats` d'identifiant `bresult` pour afficher les résultats quand on le clique. **Ajoutez** dans le contrôleur une fonction `displayResult()` qui affiche les résultats de la table courante dans les champs texte mis en lecture seule. **Écrivez** le gestionnaire d'événements `onClick` à appeler quand on clique sur un contrôle, associez-le au corps du document et faites en sorte

que les résultats de la table soient affichés quand on clique sur le bouton Résultats. **Testez.**

Affichage de messages

On veut pouvoir afficher un message général dans l'élément `<p id="resultmsg">` prévu à cet effet. **Ajoutez** et exportez dans la vue une fonction `displayMessage(msg)` qui affiche le message `msg` dans ce paragraphe. **Testez.** **Modifiez** dans le contrôleur `displayResult()` pour qu'elle affiche le message Apprenez bien ces valeurs. **Testez.**

Saisie et vérification des valeurs

On veut permettre à l'élève de rentrer les résultats des multiplications et de les vérifier. **Faites** en sorte que le message Rentrez les valeurs des multiplications soit affiché quand on utilise une nouvelle table. **Testez.**

Il faut vérifier que l'élève a bien rentré toutes les valeurs et que ce sont des nombres avant de vérifier ces valeurs. On a dans le tableau des éléments de classe `check` correspondants aux champs texte et dans lesquels on veut afficher les messages d'erreur. **Ajoutez** dans la vue une variable `checkClassElements` contenant les éléments de classe `check` à initialiser au chargement de la page. **Ajoutez** et exportez la fonction `eraseCheck()` qui efface le contenu de ces éléments. Faites en sorte que ces éléments soient effacés à chaque nouvelle table. **Ajoutez** et exportez la fonction `checkAnswersValidity()` qui vérifie la validité des champs texte : si un champ texte est vide il faut afficher entrez une valeur, si un champ texte ne contient pas un nombre, il faut afficher entrez un nombre, sinon il faut effacer l'ancien message ; s'il y a des champs textes invalides, il faudra en plus afficher un dialogue avec le message `données incorrectes` et renvoyer faux, sinon il faut renvoyer vrai.

On a mis un bouton Vérifier d'identifiant `bverify` pour vérifier les réponses. **Ajoutez** dans le contrôleur la fonction `verifyAnswers` qui pour l'instant vérifie la validité des réponses dans les champs textes. **Modifiez** le gestionnaire d'événements `onClick` du contrôleur pour que la validité des champs texte soit vérifiée quand l'utilisateur clique sur ce bouton. **Testez.**

Si les champs textes sont correctement remplis, il faut alors vérifier les réponses. **Ajoutez** dans la vue la fonction `checkAnswersCorrectness(results)` qui vérifie si les champs textes contiennent les valeurs du tableau `results` : si un champ texte contient la bonne valeur il faut afficher bonne réponse, sinon afficher réponse erronée ; si toutes les valeurs sont bonnes, il faudra afficher le message général de félicitations Bravo tu as trouvé toutes les bonnes

réponses, sinon afficher un message général indiquant le nombre de bonnes réponses. **Modifiez** la fonction `verifyAnswers` du contrôleur pour qu'elle vérifie, si les réponses sont valides, qu'elles correspondent aux valeurs de la table de multiplication. **Testez.**

Vous pouvez continuer à améliorer cette application