

1 Apprentissage de tables de multiplication

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

L'application est répartie dans trois fichiers : le fichier `tablemultview.js` contient la vue, c'est-à-dire les données et les méthodes permettant de manipuler l'aspect de la page ; le fichier `tablemultdata.js` contient les données, c'est-à-dire les données et les méthodes permettant de faire les calculs ; enfin le fichier `tablemultcontroller.js` contient le contrôleur, c'est-à-dire les données et les méthodes permettant de gérer les interactions avec l'utilisateur. **Écrivez** ces trois fichiers et faites les charger par la page. **Ajoutez** dans le module contrôleur la fonction `onLoad` qui servira à initialiser le module et faites en sorte qu'elle soit appelée juste après le chargement de la page, **ajoutez** la fonction `onUnload` qui servira à faire le ménage juste avant de quitter la page et faites en sorte qu'elle soit justement appelée juste avant de quitter la page.

2 Données

Les données contiennent le numéro `currentTableNumber` de la table de multiplication **Ajoutez** (dans le fichier `tablemultdata.js`) cette donnée (initialisée à 0). **Ajoutez** les fonctions : `tableNumber()` qui renvoie le numéro de la table, `tableValues()` qui renvoie le tableau des valeurs de la table de multiplication pour le numéro de la table et pour les lignes 1 à 10, `changeTableNumber(number)` qui change le numéro de la table de multiplication avec celui passé en paramètre, `valuesVerification(values)` qui prend en paramètre un tableau de valeurs et renvoie un tableau de booléens valant vrai si la valeur correspond à celle correspondante de la table de multiplication, faux sinon et enfin `numberOfGoodValues(values)` qui renvoie le nombre de valeurs du tableau `values` correspondants aux valeurs correspondantes de la table de multiplication. **Testez** dans la console.

3 Initialisation et mise à jour du numéro

Il faut mettre le numéro de la table dans tous les éléments de classe `num`. **Ajoutez** dans la vue (dans le fichier `tablemultview.js`) 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 la vue la fonction `selectedTableNumber()` qui renvoie le numéro sélectionné. **Testez**.

Ajoutez dans le module `Controller` la fonction `newTable()` qui fait que les données soient ré-initialisées avec le numéro de la table de multiplication sélectionné et que l'affichage de la vue soit mise à jour à partir de ce numéro. **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**.

4 Calcul et affichage des résultats

Il faut afficher les résultats dans les champs de saisie texte de classe `result`. **Ajoutez** dans la vue la fonction `displayResults(results)` qui affiche dans ces champs texte les valeurs du tableau `results` et les met en lecture seule. **Testez**. **Ajoutez** la fonction `eraseResults()` qui efface ces champs texte et les met 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 la 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**.

5 Affichage de messages

On veut pouvoir afficher un message général dans l'élément `<p id="resultmsg">` prévu à cet effet. **Ajoutez** dans la vue la 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**.

6 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. Il y a dans le tableau HTML des éléments de classe `check` dans lesquels on veut afficher les messages de vérification par rapport aux champs de saisie. **Ajoutez** dans la vue la fonction `displayChecks(msg)` qui remplace le contenu de chacun de ces éléments par les chaînes du tableau `msg`. **Ajoutez** la fonction `eraseChecks()` qui efface le contenu de ces éléments (il faut remplacer leur contenu par une chaîne vide). **Faites** en sorte que ces éléments soient effacés à chaque nouvelle table.

Ajoutez la fonction `answersValidities()` qui renvoie un objet dont la propriété `msgs` est un tableau de chaînes contenant pour chaque champ de saisie de texte "entrez une valeur" si le champ était vide, "entrez un nombre" si le champ ne contenait pas un nombre, "" si le champ contenait un nombre, et dont la propriété `valid` vaut vrai si tous les champs contenaient un nombre. **Ajoutez** la fonction `displayAnswersValidities(validities)` qui affiche les messages de vérification des champs texte et en plus affiche un dialogue avec le message données incorrectes s'il y a des données incorrectes.

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. Il faut récupérer les valeurs des réponses pour ensuite pouvoir les vérifier. **Ajoutez** dans la vue la fonction `userAnswers()` qui renvoie le tableau des valeurs saisies par l'utilisateur dans les champs texte. **Ajoutez** dans la vue la fonction `displayAnswersCorrectnesses(results)` qui affiche les messages de vérification (dans les éléments de classe `check`) correspondant aux tableau de booléens des résultats des réponses `results` : si le résultat est bon, il faut afficher "bonne réponse", sinon il faut afficher "réponse erronée" ; la fonction doit aussi si toutes les réponses sont bonnes 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**.