|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **BAREME DE NOTATION – Noms et prénoms : ........................................................................................................................** | | | | | | |
| **Évaluation en classe (sur 10 points)** | | | | | | |
| **Attitude**  *Concentration, implication, sérieux, ...* |  | | | **SOS professeur** | Image associÃ©e   |  |  |  |  | | --- | --- | --- | --- | | *> 3 appels* | *2 appels* | *1*  *appel* | *Aucun appel* | | |
| **Évaluation du compte rendu (sur 10 points)** | | | | | | |
| **Rédaction**  *Lisibilité, rigueur, justification, ...* | Image associÃ©e | **Résolution** |  | | | **NOTE :** |

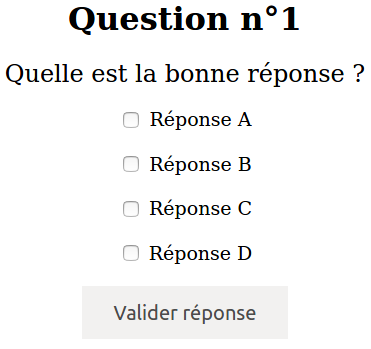
### Introduction

Dans ce projet, vous allez créer des questions de QCM à partir de pages HTML et de scripts JavaScript. L'objectif est que vous créiez au moins une question de QCM chacun. Pour faire ce projet, vous devez vous rappeler des bases de l'HTML (vous ne re-coderez pas dans ce langage mais vous devrez comprendre le code fourni), et vous pouvez/devez vous aider du cours de JavaScript. Des aides sont données après certaines questions, lisez bien les questions jusqu'au bout avant de commencer à y répondre !

Pour se lancer, vous devez allez chercher les 3 fichiers suivant dans le commun NSI et les copier dans un dossier QCM sur votre Z : *qcm.html*, *script.js*, *style.css*.

### Question QCM

Définissez une question de QCM pouvant être posée à l'examen du baccalauréat sur le programme de NSI de première. La question doit avoir 4 réponses possibles dont au moins une est bonne. Plusieurs bonnes réponses sont possibles. Le but de ce qui suit est de modifier les fichiers fournis afin que ce soit votre question qui s'affiche à l'ouverture du fichier HTML dans un navigateur, qu'un utilisateur puisse essayer d'y répondre et qu'une réponse lui soit donnée.



**N.B.** : le jour du bac, une seule réponse sera possible. Il est interdit de prendre une question déjà posée lors d'un QCM précédent.

### Partie HTML

Dans un premier temps, ouvrez le fichier HTML dans un navigateur et dans un éditeur de texte, et répondez aux questions suivantes sous forme de commentaires dans le code. Pour commenter en HTML on utilise la balise suivante :

CTRL+C pour copier, CTRL+V pour coller

1

<!-- Texte commenté -->

1. Pour chaque élément de la liste ci-dessous, écrivez son **numéro** ainsi que son **identifiant** en commentaires de la ligne où il est créé :

* 1 : le titre de la question de QCM
* 2 : le texte de la première réponse possible
* 3 : le bouton de validation de la réponse de l'utilisateur
* 4 : l'énoncé de la question de QCM
* 5 : la case à cocher de la dernière réponse possible

2. Repérez la/les fonction·s JavaScript qui apparaissent dans ce code, en commentant leur nom à la fin de la/des ligne·s où elle·s est/sont appelée·s.

#### 3. Où sont contenus les éléments de style de cette page HTML ? Écrivez la réponse en commentaire à la fin de LA ligne concernée.

4. Repérez chaque élément permettant à l'utilisateur d’interagir avec la page par le commentaire "**\***" à la fin de la ligne où il est créé.

### Partie JavaScript

5. Incluez le fichier JavaScript au fichier HTML.

**ATTENTION : Faire un rendu du fichier HTML que vous nommerez *qcm\_etape1.html*.**

À partir de maintenant, vous ne modifierez plus le fichier HTML.

6. Dans le fichier *script.js*, repérez la ligne qui permet d'afficher le titre au chargement de la page par un commentaire "affiche énoncé" et repérez les lignes qui permettent d'afficher les réponses possibles au chargement de la page par des commentaires "affiche réponses".

Aide : Un commentaire en JavaScript s'écrit :

CTRL+C pour copier, CTRL+V pour coller

1

/\* Texte commenté \*/

7. Modifiez des constantes dans le script pour que, au chargement de la page HTML, l'énoncé de la question et les réponses affichés soient ceux que vous avez choisis dans la partie 2 : *Question QCM*.

8. Trouvez la fonction qui est réalisée à l'appui sur le bouton et complétez-la afin que le titre de la question devienne "*Mets des paillettes dans mon titre Kevin*" lorsqu'on appuie sur le bouton.

9. Changez également la couleur du titre à l'appui sur le bouton (couleur au choix).

**ATTENTION : Faire un rendu du fichier JavaScript que vous nommerez *script\_etape2.js*.**

10. À présent, vous devez modifier le titre à l'appui sur le bouton uniquement si la première case est cochée.

Aide : Retrouvez l'élément "boîte à cocher" dans l'HTML et son identifiant. En JavaScript, lorsque qu'une case à cocher d'identifiant *ma\_case* est effectivement cochée, on a :

CTRL+C pour copier, CTRL+V pour coller

1

document.getElementById(id="ma\_case").checked == true

Si elle n'est pas cochée, on a :

CTRL+C pour copier, CTRL+V pour coller

1

document.getElementById(id="ma\_case").checked == false

Sachez que les booléens *true* et *false* sont toujours équivalents et remplaçables par les valeurs 1 et 0 respectivement.

11. Créez une constante nommée "*a\_comparer*" dans le script. Cette constante peut valoir 0 ou 1. Choisissez, au choix, une des deux valeurs.

Maintenant, le titre doit être modifié **si et seulement si** la première case est dans le même état que la constante.

**ATTENTION : Faire un rendu du fichier JavaScript que vous nommerez *script\_etape3.js*.**

### Partie JavaScript avancé

On souhaite à présent qu'à l'appui sur le bouton, le titre et la couleur des réponses soit modifiés en fonction de l'état de toutes les cases à cocher. Les questions suivantes vont vous guider pour parvenir à ce résultat.

12. Modifiez la liste "*bonne\_reponse*" pour qu'elle contienne la réponse réelle à votre question.

Aide : Un 0 ou un 1 correspond à une des réponses possibles :

* 0 : "ne doit pas être cochée",
* 1 : "doit être cochée",

Voir question 4.7.

12. A présent, à l'appui sur le bouton, le titre doit être modifié en fonction de toutes les cases cochées :

* Si les cases cochées correspondent au contenu de "*bonne\_reponse*", le titre doit être modifié par "*Bonne réponse :)*", en vert.
* Sinon, il doit être modifié par "*Mauvaise réponse :(*" en rouge.

Aide : Vous pouvez vous aider de cette page: <https://www.w3schools.com/js/js_comparisons.asp> pour formuler vos conditions (*Si...Alors...*), notamment pour mettre plusieurs conditions dans un *Si*... .

Grâce aux deux questions suivantes, vous allez à présent modifier les textes des réponses possibles en fonction des cases cochées par l'utilisateur, à l'appui sur le bouton :

* Les réponses possibles correspondant à des cases devant être cochées, doivent toujours être **colorisées en vert**.
* Les réponses possibles correspondant à des cases ne devant pas être cochées mais ayant été cochées doivent être **colorisées en rouge**.
* Les autres réponses possibles doivent **rester noires**.

13. Écrivez, en commentaire, l'algorithme sous forme de [**pseudo-code**](https://fr.wikipedia.org/wiki/Pseudo-code) et avec vos propres mots (*Si...Alors...*)

14. Traduisez cet algorithme en JavaScript sous le commentaire précédent.

15. Modifiez le code pour que, une fois la réponse donnée, le bouton permette de **rejouer la question** :

1. Au premier appui sur le bouton le texte du bouton doit changer et indiquer "*Rejouer*".
2. Au second appui sur le bouton, la page doit revenir à l'état initial.

Aide :

* Le texte d'un bouton d'identifiant "*mon\_bouton*" est contenu dans :

CTRL+C pour copier, CTRL+V pour coller

1

document.getElementById("mon\_bouton").innerHTML

* Créez une variable qui pourra valoir 0 ou 1 pour indiquer l'état de la page :
  + 0 : "en attente de réponse" ;
  + 1 : "réponse donnée, en attente de rejouer".

Cette variable devra être modifiée à chaque appui sur le bouton pour passer à l'état suivant.

**ATTENTION : Faire un rendu du fichier JavaScript que vous nommerez *script\_etape4.js*.**

### Bonus : Faire une deuxième question de QCM

À partir de ce qui a été vu précédemment, créez une deuxième question de QCM.

Comme précédemment, la page doit se charger avec l'énoncé de la question 1. Lorsqu'on appuie pour la première fois sur le bouton, la réponse de la question 1 doit s'afficher. Si on clique une deuxième fois sur le bouton, on doit passer à la question 2. Au quatrième clic, la réponse de la question 2 doit s'afficher. Enfin, au dernier clic la page doit revenir à l'état initial.

Lorsqu'on clique sur le bouton, il y a donc 4 possibilités d'action.

**ATTENTION : Faire un rendu du fichier JavaScript que vous nommerez *script\_bonus.js*.**