

**Filière : Informatique et intelligence artificielle****Semestre : S4 Année universitaire : 2024-2025****M244 : PROGRAMMATION WEB 2 : LANGAGE JAVASCRIPT****Chapitre 6 : TP3 - Créer un quiz interactif avec POO, AJAX et jQuery****Le TP sera composé de trois fichiers :**

index.html : Contient le formulaire du quiz.

styles.css : Contient les styles du quiz.

script.js : Gère la logique du quiz avec des fonctions.

**Exercice Pratique 1**

Vous allez créer un quiz interactif en JavaScript qui permet à l'utilisateur de répondre à 5 questions sur les bases de JavaScript. Le quiz doit comporter un formulaire avec 5 questions, un bouton pour valider les réponses, et un affichage du score final.

**ÉTAPES A REALISER :****1. Créer un formulaire (HTML) :**

Dans le fichier index.html, créez un formulaire contenant **5 questions avec 4 propositions de réponse** (radio buttons) pour chaque question. Une seule réponse est correcte pour chaque question.

**2. Créer une classe Quiz (JavaScript - POO) :**

Dans le fichier script.js, créez une classe Quiz pour structurer le code. La classe doit inclure les méthodes suivantes :

**Vérification des réponses** : Assurez-vous que toutes les questions ont une réponse.

**Calcul du score** : Comparez les réponses avec les bonnes réponses et calculez le score.

**Affichage du résultat** : Affichez un message indiquant le score ou une alerte si toutes les questions ne sont pas répondues.

**Méthodes à créer :**

**toutesRepondues()** : Vérifie si toutes les questions

ont été répondues.

**calculerScore()** : Calcule le score en comparant les réponses données avec les bonnes réponses.

**afficherResultat()** : Affiche le résultat final.

**3. Gestion des événements avec jQuery :**

Utilisez **jQuery** pour gérer l'événement **click** du bouton "Vérifier". Lorsque l'utilisateur clique sur ce bouton :

Vérifiez que toutes les questions ont été répondues.

Calculez le score.

Affichez le résultat dans un élément HTML.

**4. Vérification et affichage des résultats (AJAX - optionnel) :**

Pour aller plus loin, vous pouvez utiliser **AJAX** pour envoyer les résultats à un serveur ou API. Par exemple, après avoir calculé le score, envoyez-le à un serveur pour l'enregistrer.

**REDACTION D'UN COMPTE RENDU DU TP :**

Chaque étudiant doit produire un **rapport écrit structuré** contenant les éléments suivants :

**Introduction** : Présentez brièvement l'objectif du TP et les technologies utilisées.

**Code JavaScript** : Expliquez les grandes lignes de la logique du code JavaScript (classe Quiz, méthodes, gestion des événements, etc.).

**Réponses aux questions** : Répondez clairement et de manière structurée aux questions posées dans la fiche.

**Conclusion** : Résumez brièvement ce que vous avez appris et comment vous amélioreriez ce quiz à l'avenir.

**Le rapport doit être rédigé en français ou en anglais, de manière claire et concise, et ne doit pas dépasser 6 pages.**

**Bonne chance !**

*Pr. Farida BOUROUMANE*

## TP – Fiche Questions

1. Pourquoi avons-nous utilisé des boutons radio pour chaque question du quiz ?
2. Qu'est-ce que la programmation orientée objet (POO) et pourquoi l'avons-nous utilisée dans ce TP ?
3. Expliquer la différence entre == et === en JavaScript. Pourquoi est-il important d'utiliser === dans ce quiz ?
4. Expliquez le rôle de la méthode toutesRepondues() dans le code JavaScript.
5. Comment le score est-il calculé dans ce quiz ? Décrivez la logique du calcul du score.
6. Pourquoi avons-nous utilisé jQuery pour gérer l'événement de soumission du formulaire ? Quels avantages cela apporte-t-il par rapport à l'utilisation du JavaScript pur ?
7. Qu'est-ce qu'AJAX et pourquoi avez-vous choisi de l'ajouter comme fonctionnalité optionnelle dans ce TP ?
8. Qu'est-ce que le DOM en JavaScript et comment est-il utilisé dans ce TP pour manipuler les éléments de la page ?
9. Comment la méthode afficherResultat() fonctionne-t-elle pour afficher le score final dans le quiz ?
10. Quelle est l'importance des événements en JavaScript et pourquoi avons-nous utilisé des événements pour valider les réponses dans ce quiz ?
11. Expliquez le rôle de la méthode querySelectorAll() dans la récupération des éléments du DOM pour ce quiz.
12. Comment le code JavaScript gère-t-il les erreurs ou les cas où une question n'est pas sélectionnée ?
13. Pourquoi avons-nous choisi de n'afficher que le score final une fois que toutes les questions ont été répondues ?
14. Dans ce TP, pourquoi avons-nous utilisé une boucle forEach() pour parcourir les réponses des utilisateurs ?
15. Quelles sont les implications de l'utilisation d'un événement de type submit par rapport à un événement click dans ce quiz ?
16. Expliquez comment l'utilisation de la méthode setAttribute() peut améliorer la gestion du DOM dans ce type de projet.
17. Pourquoi est-il essentiel de valider les réponses de l'utilisateur avant de calculer et d'afficher le score ?
18. Quels sont les avantages d'utiliser un système de classe pour gérer un quiz au lieu d'utiliser simplement des fonctions globales ?
19. Quelles sont les différences de performances entre les méthodes DOM traditionnelles et l'utilisation de jQuery pour manipuler les éléments d'une page ?
20. Expliquez comment vous pourriez ajouter une fonctionnalité pour chronométrier le quiz et afficher le temps écoulé à la fin.
21. Expliquez la différence entre null et undefined en JavaScript. Pourquoi est-ce important de comprendre cette différence pour ce TP ?
22. Pourquoi est-il essentiel d'ajouter des commentaires dans le code pour expliquer la logique derrière certaines décisions, notamment dans des projets collaboratifs ?