

Programmation web : TP

JavaScript

Exercice 1:

Écrivez un script JavaScript qui demande à l'utilisateur de saisir deux nombres, puis affiche la somme, la différence, le produit et le quotient de ces deux nombres dans la console.

Exercice 2 :

Faites un programme qui permet à l'utilisateur de deviner un nombre entre 1 et 10 choisi par l'ordinateur au hasard en indiquant à chaque proposition (lue avec la fonction `prompt`) si le nombre à deviner est plus grand ou plus petit que la proposition. Lorsque le nombre est trouvé vous afficherez le « score » c'est à dire le nombre de tentatives nécessaires pour arriver à la solution.

Exercice 3 : Création d'un quiz interactif en JavaScript

Votre tâche consiste à développer une application JavaScript qui fonctionne comme un quiz interactif. L'application posera une série de questions à l'utilisateur, recueillera ses réponses, et enfin, affichera le score de l'utilisateur, c'est-à-dire le nombre de réponses correctes comparé au nombre total de questions.

1- Préparer les questions :

- Créez un tableau `QUESTIONS` où chaque entrée est un autre tableau contenant deux éléments : la question en première position et la réponse correcte en seconde position. Par exemple, ["Quelle est la capitale de la France ?", "Paris"]

2- Interroger l'utilisateur :

- Créez une fonction `lancerQuiz()` dans votre script JavaScript
- Utilisez une boucle `for` pour parcourir chaque paire question-réponse dans le tableau `QUESTIONS`
- Pour chaque question, utilisez la fonction `prompt()` pour demander à l'utilisateur de saisir sa réponse
- Stockez la réponse de l'utilisateur dans une variable.

3- Vérification des réponses :

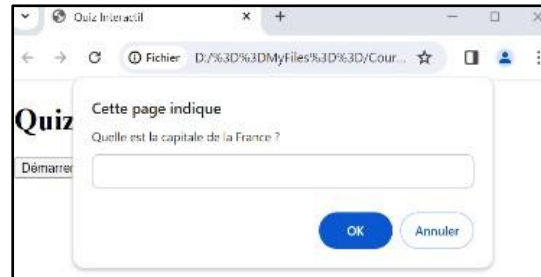
- Comparez la réponse saisie par l'utilisateur avec la réponse correcte stockée dans le tableau.
- Si la réponse est correcte, affichez "Réponse juste" et augmentez un compteur de réponses correctes.
- Sinon, affichez "Réponse fausse".

4- Résultats de l'évaluation :

- À la fin du quiz, affichez le nombre total de réponses correctes ainsi que le nombre de questions posées, par exemple : "Vous avez répondu correctement à 3 questions sur 5".

NB : La fonction JavaScript `lancerQuiz()` démarrera le quiz lorsque l'utilisateur cliquera sur le bouton `<button onclick="lancerQuiz()">Démarrer le Quiz</button>` inclus dans votre page HTML.

Exemple de résultat prévu :



Guide JavaScript

JavaScript est un langage de programmation qui permet de rendre les sites web interactifs. Il est exécuté principalement dans les navigateurs web mais est également utilisé sur les serveurs avec Node.js ou d'autres plateformes.

1. Déclaration de Variables

JavaScript offre plusieurs mots-clés pour la déclaration de variables :

- **var** : pour déclarer une variable avec une portée de fonction ou globale.
- **let** : pour déclarer une variable avec une portée limitée au bloc.
- **const** : pour déclarer une variable dont la valeur ne peut pas être modifiée après sa première affectation.

2. Types de Données

Les principaux types de données en JavaScript sont :

- Nombre : Utilisé pour les valeurs numériques.
- Chaîne de caractères : Pour le texte.
- Booléen : Vrai ou faux.
- Objet : Une collection de propriétés.
- Tableau : Une liste de valeurs.

3. Opérateurs

Les opérateurs permettent de réaliser des calculs ou de comparer des valeurs :

- Arithmétiques : +, -, *, /
- Comparaison : ==, ===, !=, !==
- Logiques : &&, ||, !

4. Structures de Contrôle et Boucles

Les instructions conditionnelles et les boucles contrôlent le flux d'exécution :

- **if...else** : Permet d'exécuter des blocs de code de manière conditionnelle.
- **switch...case** : Une structure de contrôle alternative à if...else quand il y a plusieurs conditions à comparer avec la même variable.
- **for** : Une boucle qui répète un bloc de code un nombre spécifié de fois.
- **while** : Une boucle qui continue de répéter un bloc de code tant que la condition spécifiée est vraie
- **do...while** : Similaire à la boucle while, mais garantit que le bloc de code sera exécuté au moins une fois avant que la condition ne soit évaluée, puis continue tant que la condition est vraie.

5. Fonctions

Les fonctions sont des blocs de code qui peuvent être appelés pour effectuer une tâche spécifique.

```
function Bonjour (name) {  
    return "Bonjour" + name + '!';  
}
```

6. Interaction avec l'Utilisateur

- **alert()** : affiche un message.
- **prompt()** : demande une entrée à l'utilisateur.
- **confirm()** : demande une confirmation.
- **console.log()** : Affiche des informations dans la console du navigateur. Cela peut être utile pour le débogage ou pour suivre le processus sans interrompre l'expérience utilisateur avec des pop-ups. Pour voir les messages en console, ouvrez l'outil de développement du navigateur (F12 ou clic droit puis "Inspecter") ou utilisez la console d'un IDE.