

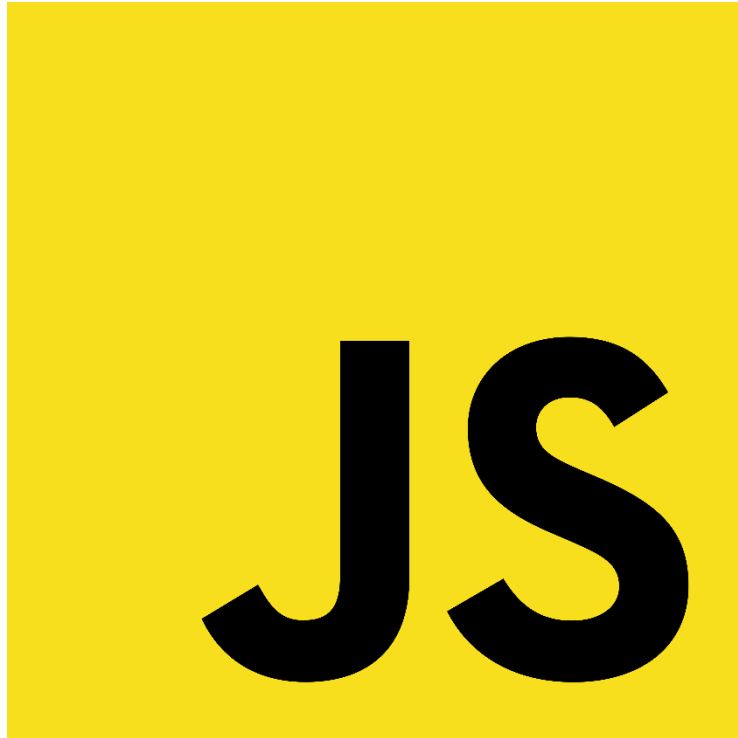
**Projet personnel réalisé en**  
**cours de formation :**

**Développement d'une  
application web ludique de  
géographie en JAVASCRIPT**

# Sommaire :

<b>I.</b>	<b>Introduction .....</b>	<b>3</b>
<b>II.</b>	<b>Analyse du besoin .....</b>	<b>4</b>
	2.1 Règles du jeu .....	4
	2.2 Fonctionnalités attendues .....	4
<b>III.</b>	<b>Développement JAVASCRIPT .....</b>	<b>5</b>
	3.1 Structure du code .....	5
	3.2 Interface utilisateur .....	10
<b>IV.</b>	<b>Conclusion .....</b>	<b>13</b>

# I. Introduction



Le projet **GéoGenius** est une application web interactive conçue pour tester les connaissances géographiques des utilisateurs. Ce projet permet d'illustrer les concepts fondamentaux du développement Front-End :

- Manipulation dynamique du DOM.
- Gestion des événements utilisateur.
- Logique de programmation en JavaScript.
- Design responsive.

**Objectif :** Développer une interface fluide permettant de choisir un continent, de répondre à une série de questions sur les capitales en les saisissant, avec un calcul de score en temps réel.

## II. Analyse du besoin

### 2.1 Règles du jeu

L'utilisateur arrive sur l'interface, Il sélectionne un continent via des boutons radio et ensuite un pays s'affiche de manière aléatoire. Il saisit le nom de la capitale et valide : si la réponse est bonne le score s'incrémente d'un point, sinon il ne bouge pas. Le jeu passe à la question suivante jusqu'à atteindre la fin de la série (10 questions), enfin un message de bilan et le score s'affiche selon la réussite de l'utilisateur.

### 2.2 Fonctionnalités attendues

L'application doit :

- Permettre la sélection dynamique d'une zone géographique.
- Vérifier la validité des réponses saisies.
- Mettre à jour le score dynamiquement.
- S'adapter aux types d'écrans.
- Réinitialiser la progression lors d'un changement de continent.

# III. Développement HTML,CSS,JS

## 3.1 Structure du code

On commence par créer la base du projet avec la structure html de la page web : les titres, boutons radios, zone de saisie, bouton de validation, bouton de réinitialisation de la partie et la zone d'affichage du score et du message.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>GéoGénius</title>
  <link rel="stylesheet" href="stylecapi.css">
  <script src="scriptsss.js" defer</script>
</head>

<!-- Début du corp-->
<body>
  <main>
    <div class="enTete">
      
      <h1>GéoGénius</h1>
    </div>
    <p> Choisissez une région du monde et tapez la bonne capitale en fonction du pays proposé </p><br>
    <div class="continents">
      <input type="radio" name="Continent" id="Afrique" value="1">
      <label for="Afrique">Afrique</label>
      <input type="radio" name="Continent" id="Asie" value="2">
      <label for="Asie">Asie</label>
      <input type="radio" name="Continent" id="Amérique" value="3">
      <label for="Amérique">Amérique</label>
      <input type="radio" name="Continent" id="Europe" value="4">
      <label for="Europe">Europe</label>
      <input type="radio" name="Continent" id="Océanie" value="5">
      <label for="Océanie">Océanie</label>
    </div>
    <div class="zonePropPays">
      Choisissez un continent !
    </div>

    <div class="zoneEntreCapi">
      <input type="text" name="textCapi" id="textCapi">
      <div class="btnValider"> <button name="btnValide" id="btnValide">Valider</button></div>
      <div class="btnRejouer"> <button name="btnRejouer" id="btnRejouer" onclick="location.reload()">Rejouer</button></div>
    </div>

    <div class="zoneScore">
      Votre score est de :&nbsp; <span>0</span> /10
    </div>

  </main>
  <footer>

  </footer>
</body>
<!-- Fin du corp-->

</html>
```

## On peut passer à la gestion de la logique du code et de son comportement !

Tout d'abord, on place nos capitales par continents dans un constante qui est un tableau de tableaux:

```
const continents = {
  Afrique: [
    { pays: "Maroc", capitale: "Rabat" },
    { pays: "Algérie", capitale: "Alger" },
    { pays: "Tunisie", capitale: "Tunis" },
    { pays: "Côte d'Ivoire", capitale: "Yamoussoukro" },
    { pays: "Nigeria", capitale: "Abuja" },
    { pays: "Mali", capitale: "Bamako" },
    { pays: "Ghana", capitale: "Accra" },
    { pays: "Kenya", capitale: "Nairobi" },
    { pays: "Sénégal", capitale: "Dakar" },
    { pays: "Égypte", capitale: "Le Caire" }
  ],
  Asie: [
    { pays: "Japon", capitale: "Tokyo" },
    { pays: "Chine", capitale: "Pékin" },
    { pays: "Inde", capitale: "New Delhi" },
    { pays: "Corée du Sud", capitale: "Séoul" },
    { pays: "Thaïlande", capitale: "Bangkok" },
    { pays: "Vietnam", capitale: "Hanoï" },
    { pays: "Indonésie", capitale: "Jakarta" },
    { pays: "Philippines", capitale: "Manille" },
    { pays: "Malaisie", capitale: "Kuala Lumpur" },
    { pays: "Arabie Saoudite", capitale: "Riyad" }
  ],
  Amérique: [
    { pays: "États-Unis", capitale: "Washington D.C." },
    { pays: "Canada", capitale: "Ottawa" },
    { pays: "Brésil", capitale: "Brasília" },
    { pays: "Mexique", capitale: "Mexico" },
    { pays: "Argentine", capitale: "Buenos Aires" },
    { pays: "Chili", capitale: "Santiago" },
    { pays: "Colombie", capitale: "Bogotá" },
    { pays: "Pérou", capitale: "Lima" },
    { pays: "Uruguay", capitale: "Montevideo" },
    { pays: "Cuba", capitale: "La Havane" }
  ],
}
```

```

Europe: [
  { pays: "France", capitale: "Paris" },
  { pays: "Allemagne", capitale: "Berlin" },
  { pays: "Espagne", capitale: "Madrid" },
  { pays: "Italie", capitale: "Rome" },
  { pays: "Royaume-Uni", capitale: "Londres" },
  { pays: "Portugal", capitale: "Lisbonne" },
  { pays: "Grèce", capitale: "Athènes" },
  { pays: "Suède", capitale: "Stockholm" },
  { pays: "Norvège", capitale: "Oslo" },
  { pays: "Suisse", capitale: "Berne" }
],
Océanie: [
  { pays: "Australie", capitale: "Canberra" },
  { pays: "Nouvelle-Zélande", capitale: "Wellington" },
  { pays: "Fidji", capitale: "Suva" },
  { pays: "Samoa", capitale: "Apia" },
  { pays: "Papouasie-Nouvelle-Guinée", capitale: "Port Moresby" },
  { pays: "Tonga", capitale: "Nuku'alofa" },
  { pays: "Vanuatu", capitale: "Port-Vila" },
  { pays: "Tuvalu", capitale: "Funafuti" },
  { pays: "Kiribati", capitale: "Tarawa" },
  { pays: "Salomon", capitale: "Honiara" }
]
];

```

Ensuite, on met en place nos variables de gestion et les éléments du DOM que l'on va manipuler :

```

let score = 0;
let currentIndex = 0;
let selectedContinent = "";
let propositions = [];

const btnValide = document.getElementById("btnValide");
const inputCapi = document.getElementById("textCapi");
const zonePays = document.querySelector(".zonePropPays");

```

Place alors à la gestion du choix du continent, en conséquence on propose les pays inhérents dans un ordre aléatoire :

```
document.querySelectorAll("input[name='Continent']").forEach(input => {
  input.addEventListener("change", function () {
    selectedContinent = this.id;
    // On copie et on mélange les pays pour plus de fun
    propositions = [...continents[selectedContinent]].sort(() => Math.random() - 0.5);

    currentIndex = 0;
    score = 0;
    btnValide.disabled = false; // On réactive le bouton
    mettreAJourScore();
    afficherPays();
  });
});
```

```
function afficherPays() {
  if (currentIndex < propositions.length) {
    zonePays.textContent = propositions[currentIndex].pays;
  } else {
    terminerJeu();
  }
}
```

On vérifie la validation du joueur, et on lui permet de valider avec la touche entrée ou en cliquant sur le bouton au choix :

```
btnValide.addEventListener("click", function () {
  const userInput = inputCapi.value.trim().toLowerCase();
  const bonneCapitale = propositions[currentIndex].capitale.toLowerCase();

  if (userInput === bonneCapitale) {
    score++;
  }

  currentIndex++;
  mettreAJourScore();
  inputCapi.value = "";
  afficherPays();
});
```

```
// Permettre de valider avec la touche Entrée
inputCapi.addEventListener("keypress", (e) => {
  if (e.key === "Enter") btnValide.click();
});
```

Et on met à jour l'affichage du score en conséquence :

```
function mettreAJourScore() {  
    document.querySelector(".zoneScore span").textContent = score;  
}
```

Enfin, on met fin au jeu et on affiche le message de fin selon le résultat obtenu par le joueur :

```
function terminerJeu() {  
    // On utilise le score global  
    if (score >= 7) {  
        zonePays.textContent = `BRAVO ! Score final : ${score}/10. Vous êtes un expert !`;  
    } else {  
        zonePays.textContent = `REESSAYEZ ! Score final : ${score}/10.`;  
    }  
    btnValide.disabled = true;  
}
```

### 3.2 Interface utilisateur

#### Observons maintenant le résultat !

Au lancement, la fenêtre s'ouvre avec un message de bienvenue, le joueur est invité à choisir le continent qu'il souhaite, puis ensuite un pays inhérent lui est proposé et il doit saisir la capitale correspondante.



Une fois son choix fait, la partie est lancée.



Le joueur saisi la capitale du pays proposé.



Ensuite il valide, si la réponse est bonne les score s'incrémente d'un point, sinon il ne bouge pas, et on passe au pays suivant.



Enfin, le score et le message de fin apparaissent à l'issue de la série de question, le bouton de validation est désactivé, et le message est différent selon le score obtenu par le joueur.



Une fois la partie terminée, le joueur peut soit relancer la partie en cliquant sur le bouton rejouer, soit fermer la fenêtre et mettre fin au jeu.

## IV. Conclusion

En conclusion, la réalisation du projet **GéoGenius** a marqué une étape essentielle dans mon apprentissage du développement Front-End. Bien plus qu'un simple quiz, ce projet a été un véritable laboratoire pour appliquer les mécaniques de l'interactivité web et structurer une logique applicative simple en JavaScript.

### Bilan technique

Le développement de cette application m'a permis de consolider ma maîtrise du **JavaScript** et mettre en œuvre des concepts clés tels que :

- **La manipulation dynamique du DOM** : Mise à jour en temps réel de l'interface sans rechargement de la page.
- **La gestion des événements** : Capture et traitement des saisies utilisateur pour une interactivité fluide.
- **La logique algorithmique** : Comparaison de chaînes de caractères (insensibilité à la casse/accents) et gestion du système de scoring par continent.

### Perspectives d'évolution

Bien que l'application soit pleinement opérationnelle, plusieurs pistes d'amélioration pourraient enrichir l'expérience :

- **Intégration d'une API REST** : Récupérer les données géographiques en temps réel via une API externe (comme *Rest Countries*) pour rendre la base de données inépuisable.
- **Gamification avancée** : Ajouter un compte à rebours pour augmenter la difficulté et intégrer des effets sonores.
- **Persistance des données** : Utiliser une base de données Firebase pour sauvegarder les meilleurs scores et permettre une progression sur le long terme.

En somme, **GéoGenius** démontre ma capacité à concevoir une interface web dynamique et performante, tout en respectant les standards actuels du développement Front-End.