

Introduction

Aujourd'hui, nous allons plonger dans le développement web en construisant **un jeu de mémoire interactif** !

Ce projet nous permettra de découvrir comment les fichiers HTML, CSS et JavaScript interagissent pour créer une application web dynamique.

L'objectif ? Comprendre les bases de la programmation web tout en s'amusant !

1 - Premier pas : Création de la structure HTML

1. **Créer un fichier HTML** (index.html)
2. **Supprimer les liaisons CSS et JS** et tester si la page s'affiche bien dans un navigateur.
3. **Explication** : Le HTML structure le contenu de notre page.

Code HTML :

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="UTF-8" />
    <title>Jeu de Mémoire</title>
    <link rel="stylesheet" href="src/style.css" />
  </head>
  <body>
    <h1 id="message">rien</h1>
    <div id="game-board" class="game-board"></div>
    <script src="src/script.js"></script>
  </body>
</html>
```

Question : Qu'observez-vous en ouvrant ce fichier dans un navigateur ?

2 - Ajout du CSS pour styliser notre page

1. **Créer un fichier CSS** (src/style.css)
2. **Lier le CSS à notre HTML**
3. **Explication** : Le CSS permet de définir l'apparence de notre page.

Code CSS :

```
body {  
  background-color: #f9f9f9;  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  height: 100vh;  
  margin: 0;  
}  
  
h1 {  
  color: #007acc;  
  font-family: sans-serif;  
}
```

Question : Comment le style de la page a-t-il changé ?

3 - Ajout de JavaScript pour dynamiser la page

1. **Créer un fichier JavaScript** (src/script.js)
2. **Modifier dynamiquement le contenu HTML**
3. **Explication** : Le JavaScript rend notre page interactive.

Code JavaScript :

```
document.getElementById('message').textContent = 'Hello World!';
```

Question : Que se passe-t-il lorsque vous ouvrez la page dans un navigateur ?

4 - Exercices interactifs : Création d'un jeu de mémoire

Étape 1 - Demander à ChatGPT de générer un jeu de mémoire

Prompt :

J'ai besoin d'un jeu de mémoire avec des cartes programmé en HTML, CSS et Javascript. J'aimerais que le code soit séparé entre les 3 fichiers. Mon fichier HTML est nommé index.html dans la racine, dans le dossier src se trouve mon fichier style.css et script.js.

Étape 2 - Tester le code

1. Copier-coller le code dans <https://playcode.io/>
2. Vérifier si le jeu fonctionne

Étape 3 - Debugging : Introduire une erreur et la corriger

1. Modifiez le code JavaScript comme ceci :

```
document.getElementaeazetById('message').textContent = 'Hello World!';
```

2. Observez le message d'erreur dans la console.
3. Utiliser ChatGPT pour identifier et corriger l'erreur.

Prompts :

1 - Je vais te donner mon jeu web : d'abord le HTML, ensuite le CSS et enfin le JS.

2 - Voici mon fichier HTML : (copier-coller le code HTML)

3 - Voici mon fichier CSS : (copier-coller le code CSS)

4 - Voici mon fichier JS : (copier-coller le code JavaScript)

Si ChatGPT ne repère pas l'erreur immédiatement, copier-coller le **message d'erreur de la console**.

Exemple de message d'erreur :

```
▼ TypeError: document.getElemenzaeazetById is not a function
  at <anonymous>:2:10
  at mn (<anonymous>:16:5455)
stack: "TypeError: document.getElemenzaeazetById is not a function\n at <anonymous>:2:10\n
message: "document.getElemenzaeazetById is not a function"
▼ [[Prototype]]: "TypeError\n"
  ► constructor: f TypeError()
  name: "TypeError"
  message: ""
  ► [[Prototype]]: {}
```

Étape 4 - Améliorations à l'aide de ChatGPT

Ajoutez progressivement les fonctionnalités suivantes :

Effets visuels et interactions

- **Animations CSS** : Ajoutez une transition fluide pour le retournement des cartes.
- **Messages dynamiques** : Affichez un message de bienvenue et un message de fin lorsqu'une partie est gagnée.

Mécanique du jeu

- **Système de score et de tentatives** : Ajoutez un compteur d'essais qui s'incrémente à chaque tentative.
- **Chronomètre** : Affichez un minuteur qui démarre au lancement de la partie.
- **Bouton de réinitialisation** : Permettez aux joueurs de recommencer une partie en cliquant sur un bouton.

Amélioration de l'expérience utilisateur

- **Effets sonores** : Ajoutez des sons pour le retournement des cartes, les paires réussies et les erreurs.
- **Sauvegarde des scores** : Utilisez `localStorage` pour enregistrer le meilleur score du joueur.

Extensions et personnalisation

- **Niveaux de difficulté** : Ajoutez plusieurs niveaux en variant le nombre de cartes.
 - **Thèmes personnalisables** : Proposez différents thèmes graphiques (animaux, formes, emojis, etc.).
 - **Accessibilité et responsive design** : Assurez-vous que le jeu fonctionne sur mobile et permet une navigation au clavier.
 - **Mode multijoueur** : Implémentez une version à deux joueurs où chacun joue à tour de rôle.
-

Challenge Final : Créez votre propre jeu !

Une fois que vous avez compris le fonctionnement du jeu de mémoire, essayez de créer **votre propre jeu web** avec de nouvelles règles et thématiques !

Amusez-vous et laissez parler votre créativité ! 🚀