

Projet pratique et révision

KELLY Abdoulaye

Mali_Code

12 février 2025

Plan Projet pratique et révision

- 1 Projet pratique
- 2 Révision des concepts clés
- 3 Conseils pour approfondir
- 4 Questions et réponses
- 5 Résumé du cours

- Création d'une application web simple.
- Intégration des concepts appris pendant le cours :
 - Manipulation du DOM.
 - Gestion des événements.
 - Requêtes HTTP avec 'fetch'.
 - Utilisation de 'async/await'.
- Travail en petits groupes ou individuellement.

Exemple de projet

```
<!-- Structure HTML -->
<div id="app">
  <h1>Générateur de citations</h1>
  <button id="bouton">Générer une citation</button>
  <p id="citation"></p>
</div>
```

```
// JavaScript
const bouton = document.getElementById("bouton");
const citation = document.getElementById("citation");
bouton.addEventListener("click", async () => {
  try {
    const reponse = await fetch("https://api.quotable.io/random");
    const data = await reponse.json();
    citation.textContent = data.content;
  } catch (erreur) {
    citation.textContent = "Erreur lors de la récupération de la citation."
  }
}
```

- **JavaScript de base** : variables, fonctions, structures de contrôle.
- **Manipulation du DOM** : sélection, modification, gestion des événements.
- **JavaScript moderne** : ES6+, promesses, 'async/await'.
- **Requêtes HTTP** : utilisation de 'fetch'.

Rappel sur les promesses

```
const maPromesse = new Promise((resolve, reject) => {
  setTimeout(() => {
    const succes = true;
    if (succes) {
      resolve("Succès !");
    } else {
      reject("Échec !");
    }
  }, 1000);
});

maPromesse
  .then(resultat => console.log(resultat))
  .catch(erreur => console.error(erreur));
```

Rappel sur async/await

```
async function fetchData() {  
  try {  
    const reponse = await fetch("https://api.example.com/data");  
    const data = await reponse.json();  
    console.log(data);  
  } catch (erreur) {  
    console.error("Erreur :", erreur);  
  }  
}  
  
fetchData();
```

Conseils pour approfondir

- Pratiquer régulièrement sur des plateformes comme :
 - **freeCodeCamp** : <https://www.freecodecamp.org/>
 - **LeetCode** : <https://leetcode.com/>
 - **Codewars** : <https://www.codewars.com/>
- Explorer des frameworks JavaScript comme React, Vue.js ou Angular.
- Contribuer à des projets open source sur GitHub.

Une séance de questions-réponses pour clarifier les derniers points.

- Introduction à JavaScript et à son écosystème.
- Manipulation du DOM et gestion des événements.
- Concepts modernes : ES6+, promesses, 'async/await'.
- Requêtes HTTP avec 'fetch'.
- Réalisation d'un projet pratique.

Merci !

Merci pour votre participation et bon courage pour la suite !