**Guide : Coder un jeu Pierre-Feuille-Ciseaux en JavaScript**

*Niveau : Débutant*

**1. Structure HTML de base**

Commencez par créer un fichier index.html avec une interface simple :

html

Copy

Download

Run

<!DOCTYPE html>

<html>

<head>

<title>Pierre-Feuille-Ciseaux</title>

<style>

body { font-family: Arial; text-align: center; margin-top: 50px; }

button { padding: 10px 20px; margin: 10px; cursor: pointer; }

</style>

</head>

<body>

<h1>Pierre-Feuille-Ciseaux</h1>

<div>

<button id="pierre">Pierre</button>

<button id="feuille">Feuille</button>

<button id="ciseaux">Ciseaux</button>

</div>

<p id="resultat">Choisissez une option !</p>

<script src="script.js"></script>

</body>

</html>

**2. Logique du jeu en JavaScript**

Créez un fichier script.js avec le code suivant :

**Étape 1 : Générer le choix de l'ordinateur**

javascript

Copy

Download

function choixOrdinateur() {

const options = ["pierre", "feuille", "ciseaux"];

const index = Math.floor(Math.random() \* 3); // Génère un nombre entre 0 et 2

return options[index];

}

**Étape 2 : Comparer les choix et déterminer le gagnant**

javascript

Copy

Download

function jouer(choixJoueur) {

const choixOrdi = choixOrdinateur();

let resultat;

if (choixJoueur === choixOrdi) {

resultat = "Égalité !";

} else if (

(choixJoueur === "pierre" && choixOrdi === "ciseaux") ||

(choixJoueur === "feuille" && choixOrdi === "pierre") ||

(choixJoueur === "ciseaux" && choixOrdi === "feuille")

) {

resultat = `Gagné ! L'ordinateur a choisi ${choixOrdi}.`;

} else {

resultat = `Perdu... L'ordinateur a choisi ${choixOrdi}.`;

}

document.getElementById("resultat").textContent = resultat;

}

**Étape 3 : Écouter les clics sur les boutons**

javascript

Copy

Download

document.getElementById("pierre").addEventListener("click", () => jouer("pierre"));

document.getElementById("feuille").addEventListener("click", () => jouer("feuille"));

document.getElementById("ciseaux").addEventListener("click", () => jouer("ciseaux"));

**3. Explications des concepts clés**

* **Math.random()** : Génère un nombre aléatoire entre 0 et 1. Multiplié par 3 et arrondi avec Math.floor(), on obtient un index valide pour le tableau options.
* **Comparaisons** : La logique utilise des conditions pour vérifier qui gagne selon les règles classiques :
  + Pierre bat ciseaux.
  + Feuille bat pierre.
  + Ciseaux battent feuille.
* **Événements** : Les boutons déclenchent la fonction jouer() avec le choix correspondant.

**4. Améliorations possibles**

Pour aller plus loin :

1. **Ajouter un score** :

javascript

Copy

Download

let scoreJoueur = 0, scoreOrdi = 0;

// Dans la fonction jouer() :

if (resultat.includes("Gagné")) scoreJoueur++;

else if (resultat.includes("Perdu")) scoreOrdi++;

console.log(`Score: Vous ${scoreJoueur} - ${scoreOrdi} Ordinateur`);

1. **Interface graphique** : Utilisez des emojis (✊, ✋, ✌️) ou des images.
2. **Mode "best-of"** : Limiter le nombre de manches et déclarer un vainqueur final.

**5. Code complet sur CodePen/JSFiddle**

Pour tester rapidement, vous pouvez utiliser :

* [CodePen Template](https://codepen.io/pen?template=...) *(à personnaliser)*
* Ou copier-coller les extraits ci-dessus dans un projet local.

**Résumé**

Avec moins de 30 lignes de JavaScript, vous avez codé un jeu fonctionnel ! Ce projet est idéal pour comprendre :

* Les fonctions et conditions.
* La manipulation du DOM (affichage des résultats).
* Les événements en JavaScript.

**À vous de jouer** 🎮 : Essayez d'ajouter des fonctionnalités ou de personnaliser le style !

assets/docs/

├── drp206-guide-installation-v1.pdf

├── drp206-reference-code-v1.pdf

└── drp206-cheat-sheet-v1.pdf