**Documentation Technique : Pierre-Feuille-Ciseaux en JavaScript**

*Version 1.0 · [Date]*

**1. Structure du Projet**

Copy

Download

pfc/

├── index.html # Structure HTML

├── script.js # Logique JavaScript

├── style.css # Styles CSS

└── assets/

└── docs/

└── reference-code.pdf (ce fichier)

**2. Fonctions Principales**

**\*\*choixOrdinateur()**

*Génère un choix aléatoire pour l'ordinateur.*

javascript

Copy

Download

function choixOrdinateur() {

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

const index = Math.floor(Math.random() \* 3);

return options[index];

}

* **Math.random()** : Génère un nombre aléatoire entre 0 (inclus) et 1 (exclus).
* **Math.floor()** : Arrondit à l'entier inférieur.
* **Tableau options** : Contient les 3 choix possibles.

**jouer(choixJoueur)**

*Compare les choix et détermine 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;

}

* **Logique de comparaison** :
  + Pierre écrase les ciseaux.
  + Feuille enveloppe la pierre.
  + Ciseaux coupent la feuille.
* **DOM Manipulation** : Met à jour l'élément HTML avec l'ID resultat.

**3. Gestion des Événements**

*Écoute 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"));

* **addEventListener** : Déclenche la fonction jouer() avec le choix correspondant.

**4. Diagramme de Logique**

Diagram

Code

Copy

Download

graph TD

A[Joueur clique] --> B{Choix Joueur}

B --> |Pierre| C[Comparer avec Ordi]

B --> |Feuille| C

B --> |Ciseaux| C

C --> D{Égalité ?}

D --> |Oui| E[Afficher "Égalité"]

D --> |Non| F{Joueur gagne ?}

F --> |Oui| G[Afficher "Gagné"]

F --> |Non| H[Afficher "Perdu"]

**5. Bonnes Pratiques & Optimisations**

1. **Ajouter un Score** :

javascript

Copy

Download

let score = { joueur: 0, ordi: 0 };

// Dans jouer() :

if (resultat.includes("Gagné")) score.joueur++;

else if (resultat.includes("Perdu")) score.ordi++;

1. **Utiliser des Emojis** :

javascript

Copy

Download

const emojis = { pierre: "✊", feuille: "✋", ciseaux: "✌️" };

1. **Version Simplifiée** : Remplacer les if par un switch ou un objet de règles :

javascript

Copy

Download

const regles = {

pierre: { ciseaux: "gagne", feuille: "perd" },

feuille: { pierre: "gagne", ciseaux: "perd" },

ciseaux: { feuille: "gagne", pierre: "perd" }

};

**6. FAQ Technique**

**Q1. Comment ajouter un bouton "Rejouer" ?**  
→ Ajoutez un bouton avec onclick="location.reload()".

**Q2. Pourquoi utiliser Math.floor et pas Math.round ?**  
→ Pour éviter un index 3 (hors limites du tableau).

**Q3. Comment étendre à "Pierre-Feuille-Ciseaux-Lézard-Spock" ?**  
→ Ajoutez des options et des règles supplémentaires ([exemple](https://fr.wikipedia.org/wiki/Pierre-feuille-ciseaux-l%25C3%25A9zard-spock)).

**7. Ressources Utiles**

* [MDN JavaScript](https://developer.mozilla.org/fr/docs/Web/JavaScript)
* [CodePen Exemple](https://codepen.io/pen?template=...)

**📌 À propos**  
Documentation générée pour [Nom de votre projet].  
Licence : MIT · Contribuer via [GitHub].