

Nombre mystère - nbaleatoire.html (deviner un nombre)

Consigne :

- L'ordi choisit un nombre (1–100)
- L'utilisateur tape une proposition dans un input et clique "Tester"
- Message : "plus grand / plus petit / gagné"
- Compteur d'essais affiché

Contraintes :

- `let nombreSecret, let essais`
- `if/else`
- DOM : input + message + compteur
- Pas de `prompt()`

Question.html (QCM)

Consigne :

- Afficher une question
- 4 boutons de réponse générés depuis un tableau
- Au clic : feedback vert/rouge + score

Contraintes :

- tableau (réponses)
- boucle pour créer les boutons
- événements click

Pierre, papier, ciseaux - pierrepapierciseaux.html

créer un mini-jeu interactif "Pierre / Papier / Ciseaux".

Interface attendue :

- 3 boutons : Pierre / Papier / Ciseaux
- Une zone "Choix de l'ordinateur"
- Une zone "Résultat"
- Une zone "Score" : Gagné / Égalité / Perdu

Comportement attendu :

- Quand l'utilisateur clique sur un choix :
 - l'ordinateur choisit aléatoirement (pierre/papier/ciseaux)
 - on affiche le choix de l'ordinateur

MODULE 8 — Exercices récapitulatifs

- on affiche le résultat (gagné / perdu / égalité)
- on met à jour le score

Contraintes :

- Utiliser un tableau ["pierre", "papier", "ciseaux"]
- Utiliser `Math.random()` pour le choix ordinateur
- Utiliser `if / else`
- Mettre à jour l'interface avec `textContent`

Bonus :

- Surbrillance du bouton sélectionné
- Bouton "Réinitialiser score"
- Mode contraste

Exercice — [pendu.html](#)

Créer une version simple du jeu du pendu en interface web.

Interface attendue

La page doit contenir :

- Un titre "Jeu du pendu"
- Une zone affichant le mot à deviner sous forme de tirets
Exemple : _ _ _ _ _
- Un champ pour entrer une lettre
- Un bouton "Tester"
- Une zone de message ("Bonne lettre !", "Raté...", "Déjà utilisée")
- Une zone affichant :
 - les lettres déjà proposées
 - le nombre d'erreurs

Données imposées

- Créer un tableau de mots :

```
const mots = ["WAVRE", "BOOTSTRAP", "JAVASCRIPT", "DESIGN"];
```

Choisir un mot au hasard au début.

- Le joueur propose une seule lettre
- Si la lettre est dans le mot :
 - a. le mot affiché se met à jour
- Si la lettre n'est pas dans le mot :
 - erreurs +1

MODULE 8 — Exercices récapitulatifs

- Si la lettre a déjà été proposée :
 - afficher “Lettre déjà utilisée”
 - ne pas augmenter erreurs
- Fin de partie :
 - gagné si toutes les lettres sont trouvées
 - perdu si erreurs = 6

Contraintes

- Utiliser au minimum :
 - `let` (état du jeu)
 - `if/else`
 - un tableau (mots + lettres proposées)
 - une boucle (pour afficher les tirets / vérifier victoire)
 - DOM (`textContent`, `addEventListener`)
- Ne pas utiliser `prompt()`.

Pour aller plus loin...

- Afficher une image du pendu (0 à 6) depuis `img/pendu0.png` ... `pendu6.png`
- Ajouter un clavier de lettres (boutons A-Z)