

# Secret Number

- les **conditions**
- les **boucles logiques**
- la **gestion d'état** (nombre d'essais, nombres déjà joués)

## Objectif

Développer la **logique du jeu du nombre mystère** en JavaScript :

L'ordinateur choisit un nombre aléatoire, et l'utilisateur doit le deviner en un nombre limité d'essais.

---

## Matériel fourni

- Fichier HTML ( `index.html` ) avec une zone d'affichage et un champ pour entrer un nombre
  - Fichier CSS ( `style.css` ) pour le style
  - Vous programmez uniquement la **logique du jeu**
- 

## Fonctionnalités à développer en JavaScript

### 3.1 Initialisation

- Générer un nombre aléatoire entre **1 et 10**
  - Définir le nombre maximum d'essais (5 par exemple)
  - Initialiser un tableau pour stocker les propositions déjà faites
- 

### 3.2 Gestion des propositions

Pour chaque nombre proposé par l'utilisateur :

1. Si la proposition est correcte :
  - Afficher un message de **victoire**

- Si la proposition est incorrect, la case contenant le chiffre devient "Rouge" et n'est plus sélectionnable :

### 3.3 Gestion de la fin de partie

- Victoire** : le joueur trouve le nombre avant d'avoir épuisé ses essais
- Défaite** : le joueur a utilisé tous ses essais sans trouver → afficher « *Perdu ! Le nombre était ...* »

## Étapes suggérées pour les élèves

- Initialiser le nombre mystère et le compteur d'essais
  - Écrire la fonction de traitement d'une proposition
  - Mettre à jour l'affichage après chaque tentative
  - Gérer la victoire et la défaite
  - Tester avec différents nombres
-