

Projet #1 — Calculatrice en Console (Node.js)

Objectif Pédagogique :

- Comprendre et utiliser les **variables**, **fonctions**, **conditions**, **boucles** en JavaScript.
 - Savoir interagir avec l'utilisateur dans un programme **console** avec **Node.js**.
-

Concepts techniques à maîtriser :

Concept	Explication
Le module readline	C'est un module natif de Node.js qui permet de récupérer des entrées clavier de l'utilisateur dans un terminal. On l'utilise pour demander un input puis lancer une fonction avec cet input.
Les fonctions	Chaque opération (addition, soustraction...) devra être écrite sous forme de fonction qui prend des paramètres et retourne un résultat.
Le switch	Structure très pratique pour exécuter un bloc de code différent selon la valeur choisie par l'utilisateur (utile pour le menu).
La boucle while ou do...while	Nécessaire pour répéter le menu tant que l'utilisateur ne choisit pas de quitter.
parseFloat()	Permet de convertir une chaîne de caractères (entrée utilisateur) en nombre décimal utilisable dans des calculs.

Exemple de Code :

```
const readline = require('readline');

// Interface pour lire l'entrée utilisateur
const rl = readline.createInterface({
  input: process.stdin,
  output: process.stdout
});

// Exemple d'utilisation de readline
rl.question('Tape quelque chose : ', function(answer) {
  console.log('Tu as tapé :', answer);
  rl.close();
});
```

Cet exemple montre uniquement **comment poser une question à l'utilisateur** et récupérer la réponse.

C'est à toi d'ajouter un **menu avec switch**, des **fonctions pour les calculs** et une **boucle** pour recommencer.

L'Exercice à Réaliser :

Créer une **calculatrice console** qui permet :

1. D'afficher un menu avec les opérations de base (+ - * /).
2. De demander à l'utilisateur deux nombres.
3. De réaliser le calcul en fonction de son choix (avec un **switch**).
4. D'afficher le résultat.
5. De recommencer ou quitter (avec une boucle).

Contraintes :

- Tu dois créer une **fonction par opération** (addition, soustraction...).
- Ne jamais répéter le même code inutilement : utilise des **fonctions**.
- Bien gérer les erreurs si l'utilisateur entre des valeurs non numériques.