

# Qualité & tests

## Partie 1

### Exercice 1 : Découverte des tests unitaires avec Jest

**Objectif :** Installer et configurer Jest, puis écrire des tests unitaires sur des fonctions simples.

1. Installe Jest dans un projet Node.js.
2. Crée un fichier `utils.js` contenant trois fonctions :
  - o `somme(a, b)` : retourne la somme de `a` et `b`.
  - o `estPair(n)` : retourne `true` si `n` est pair, sinon `false`.
  - o `factorielle(n)` : retourne `n!` ( $n! = n \times (n-1) \times \dots \times 1$ ).
3. Écris des **tests unitaires** avec Jest pour vérifier ces fonctions.

### Exercice 2 : Détection des erreurs et mocks

**Objectif :** Manipuler les **mocks** et tester les cas extrêmes.

1. Ajoute une gestion d'erreur pour `factorielle()` (interdiction des nombres négatifs).
2. Ajoute un **mock** pour vérifier qu'une fonction est appelée.

### Exercice 3 : Mise en place du TDD

**Objectif :** Écrire les **tests avant d'écrire le code** (TDD).

1. Implémente une fonction `inverse(chaine)` qui inverse une chaîne.
2. Suis la méthodologie TDD :
  - o **Écris les tests en premier.**
  - o Implémente la fonction pour que les tests passent.

### Exercice 4 : Premier test e2e avec Playwright

**Objectif :** Tester un **formulaire de connexion**.

1. Installe Playwright et configure-le.
2. Teste la présence des champs `email` et `mot de passe`.
3. Simule la saisie et la connexion.