

## Jour 2 – Job– 02 Test Unitaire

### Welcome to Unit Tests – Part 2

Code first or test first ?

## Vos Premiers Tests avec PHPUnit

---

### **Bienvenue dans le monde des tests unitaires !**

Aujourd'hui vous allez découvrir PHPUnit, l'outil le plus utilisé pour les tests unitaires en PHP.

L'objectif est de créer une fonction simple, écrire un test unitaire, et comprendre le fonctionnement de PHPUnit.



## Objectif :

Créer un petit projet PHP avec Composer, y intégrer PHPUnit, écrire une fonction simple et rédiger un test unitaire pour vérifier son bon fonctionnement.

- **Faites des captures d'écran pour illustrer un readme.md**
- **Les images seront intégré au readme**
- **Stocké dans un dossier image du projet**
- **Puis déposé sur github à chaque étape**
- **Avec des commits régulier**
- **Avec un nom et une description explicite et claire**

**(Pas de nom du genre "mon commit" ou "commit pierre", un nom pro c'est "modification de la connection à la BDD" par exemple, pareil pour les branches "Connection à la BDD" ou "formulaire de contact")**



- 1. Créer un dossier nommé `test-unitaire-php-debutant`.
- 2. Initialiser un projet Composer
- 3. Installer PHPUnit
- 4. Créer la structure des dossiers
  - `src/` pour les fichiers PHP5.
  - Créer un fichier de fonction
  - Dans `src/`, créer `Math.php` contenant une classe `Math` avec une méthode `addition(\$a, \$b)`.
- 6. Créer un fichier de test
  - Dans `tests/`, créer `MathTest.php` avec une classe de test qui vérifie `Math::addition`
- 7. Exemple de test
  - `$this->assertEquals(5, $math->addition(2, 3));`
  - `tests/` pour les fichiers de test
- 8. Lancer les tests
- 9. Tester les erreurs :
  - modifie la méthode pour faire échouer un test, puis corrige-la.
- 10. Crée un `README.md` avec des captures d'écran de chaque étape, commentées.



## Compétences visées

---

- Installer et configurer PHPUnit dans un projet PHP avec Composer
- Écrire des classes et méthodes simples en PHP
- Écrire des tests unitaires avec PHPUnit
- Comprendre les erreurs et les corriger
- Documenter son travail dans un README

## Rendu

---

- Faites des captures d'écran à chaque étape pour montrer la commande et le résultat même si c'est un résultat en erreur
- Rendre un "readme.md" avec ces captures d'écran à l'intérieur stockées sur votre github dans un dossier image en annotant chaque capture en expliquant le détail de chaque action et commande de manière structurée en Markdown dans un readme.md.
- partager votre projet github avec votre formateur
- Un dépôt GitHub nommé `test-unitaire-debutant` contenant :
  - Tous les fichiers du projet
  - Un fichier `README.md` avec captures d'écran et explication en Markdown
  - Des commits réguliers et nommés proprement



## Base de connaissances

---

- <https://phpunit.de/getting-started/phpunit-10.html>
- <https://www.php.net/manual/fr/>
- <https://getcomposer.org/>
- <https://github.com/>
- <https://www.markdownguide.org/basic-syntax/>