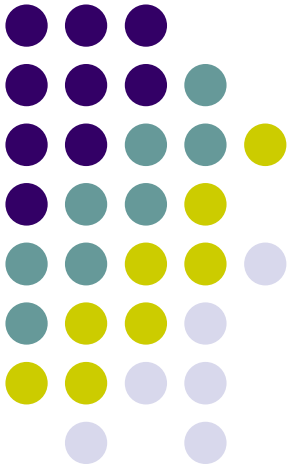
PHP

Tests unitaires



Franck LAMY – 2023/2024 – BTS SIO2

Objectifs

- Introduire la notion de tests unitaires
- Installer et utiliser PHPUnit

Tests unitaires

Un test unitaire permet de s'assurer du fonctionnement correct d'une partie déterminée d'une application ou d'une partie d'un programme.

Il a pour objectif d'isoler le comportement de la partie de code à tester de tout facteur extérieur et de vérifier qu'il est conforme à ce qui est attendu.

Tests unitaires

Un test unitaire permet de s'assurer du fonctionnement correct d'une partie déterminée d'une application ou d'une partie d'un programme.

Il a pour objectif d'isoler le comportement de la partie de code à tester de tout facteur extérieur et de vérifier qu'il est conforme à ce qui est attendu.

Tests unitaires

Le test unitaire va donc être écrit pour tester une toute petite partie du code source, indépendamment de l'environnement qui l'entoure.

Il doit être déterministe, c'est-à-dire qu'exécuté plusieurs fois, il devra toujours retourner le même résultat.

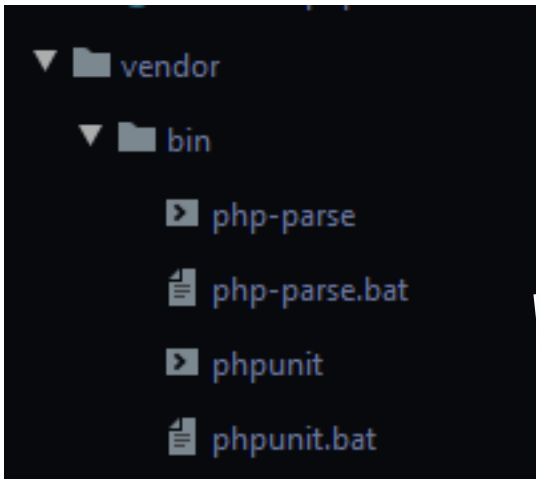
Tests unitaires

Framework de tests sous PHP : PHPUnit

Composer pour installer PHPUnit

```
composer require phpunit/phpunit --dev
```

Tests unitaires



Test si PHPUnit est bien installé

vendor/bin/phpunit

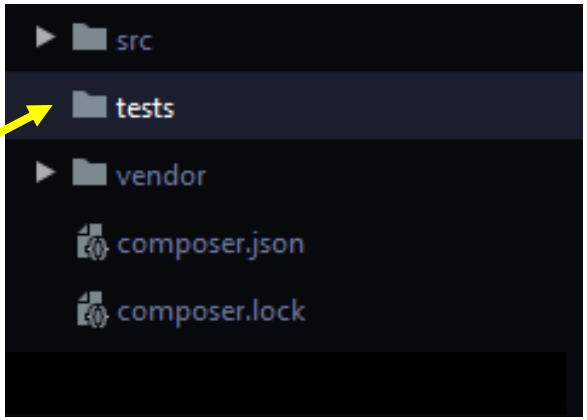
Tests unitaires

composer.js

```
{
    "require": {
        "phpunit/phpunit": "^9.4"
    },

    "autoload": {
        "psr-4": {
            "App\\" : "src/"
        }
    }
"autoload-dev": {
        "psr-4": {
            "App\\Tests\\": "tests/"
        }
    }
}
```

Déclaration d'un namespace App\Tests pour les classes de test



composer dumpautoload

Tests unitaires

ClasseA

```php
<?php
namespace App\Tests;

use PHPUnit\Framework\TestCase;

class ClasseATest extends TestCase
{
    /**
     * @test
     */
    public function doit_fonctionner() {
        $this->assertEquals(1,1);
    }
}
```

Méthode de test

Assertion

Tests unitaires


```
vendor/bin/phpunit tests --color=always
```

```
vendor/bin/phpunit tests --color=always --testdox
```

Utiliser le pattern AAA

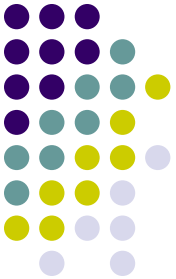
Chaque méthode de test peut-être structurée et formatée selon le pattern (modèle) AAA

AAA : Arrange Act Assert

Arrange : initialisation des objets nécessaires à l'exécution de la méthode à tester

Act : appel de la méthode à tester

Assert : vérification entre le comportement attendu et le comportement réeel de la méthode à tester



Utiliser le pattern AAA

Chaque méthode de test peut-être structurée et formatée selon le pattern (modèle) AAA

AAA : Arrange Act Assert

```php
/**
 * @test
 */
public function une_Methode_De_Test() {
    // Arrange

    // Act

    //Assert

}
```

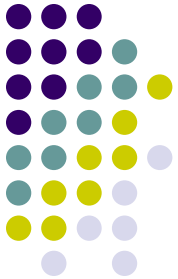
Nommer correctement une méthode de test

Les noms de méthodes de test devraient être descriptifs.

Un nom de méthode de test long et descriptif est souvent mieux qu'un nom court et obscur.

NomDeLaMethodeATester_Condition_ResultatAttendu()

Addition_AvecDesNombresPositifs_SommeCorrecte()



Assertions

La classe TestCase propose des méthodes permettant de formuler des assertions

```
$this->assertEquals(expected,actual);

$this->assertGreatherThan(expected,actual);

$this->assertTrue(condition);
```