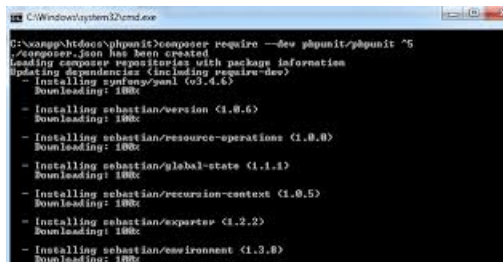


## Question 1

**Quelle est la commande pour installer PHPUnit via Composer ?**

- A. `composer require phpunit/phpunit`
- B. `composer install phpunit/phpunit`
- C. `composer require --dev phpunit/phpunit`
- D. `composer update phpunit/phpunit`

**Réponse :** C. composer require --dev phpunit/phpunit



```
C:\Windows\system32\cmd.exe

C:\xampp\htdocs\phpunit>composer require --dev phpunit/phpunit ^5
./composer.json has been created
Loading composer repositories with package information
Updating dependencies (including require-dev)
- Installing symfony/yaml (v3.4.6)
  Downloading: 100%
- Installing sebastian/version (1.0.6)
  Downloading: 100%
- Installing sebastian/resource-operations (1.0.0)
  Downloading: 100%
- Installing sebastian/global-state (1.1.1)
  Downloading: 100%
- Installing sebastian/recursion-context (1.0.5)
  Downloading: 100%
- Installing sebastian/exporter (1.2.2)
  Downloading: 100%
- Installing sebastian/environment (1.3.0)
  Downloading: 100%
```

## Question 2

**Quelle annotation est utilisée pour indiquer une méthode de test dans PHPUnit ?**

- A. @phpunit
- B. @testMethod
- C. @Test
- D. @test

## Réponse : D. @test

```
raytracer main ✓ ./tools/phpunit --testdox tests/unit/ColorTest.php
PHPUnit 11.0.0 by Sebastian Bergmann and contributors.

Runtime:      PHP 8.3.2
Configuration: /usr/local/src/raytracer/phpunit.xml

..... 7 / 7 (100%)

Time: 00:00.002, Memory: 14.79 MB

Color (SebastianBergmann\Raytracer\Color)
✓ Colors are (red, green, blue) tuples
✓ Float components can be represented as integers
✓ Another color can be added
✓ Another color can be subtracted
✓ Can be multiplied by a scalar
✓ The product of two colors can be calculated
✓ Can be compared to another color

OK (7 tests, 22 assertions)
```

## Question 3

**Quelle méthode PHPUnit est utilisée pour vérifier l'égalité de deux valeurs ?**

- A. `assertSame()`
- B. `assertEquals()`
- C. `assertEqual()`
- D. `assertTrue()`

## Réponse : B. assertEquals()

```
→ tests git:(feature/add-graphviz) ✗ grep -rin ">assert" . | awk '{print $2}' |  
  | awk -F"[" ">()" '{print $2}' | sort | uniq -c | sort -nr
```

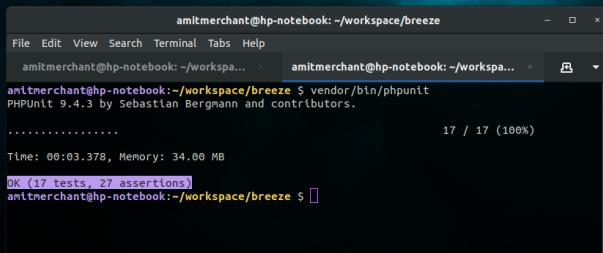
```
3145 assertEquals  
942 assertTrue  
889 assertInstanceOf  
874 assertSame  
563 assertFalse  
179 assertCount  
161 assertArrayHasKey  
158 assertNull  
112 assertInternalType  
89 assertEmpty  
81 assertContains  
66 assertNotEmpty  
26 assertNotEquals
```

## Question 4

**Quelle est la commande pour exécuter les tests PHPUnit ?**

- A. `phpunit tests`
- B. `php vendor/bin/phpunit`
- C. `phpunit run`
- D. `php tests/phpunit`

Réponse : B. php vendor/bin/phpunit



```
amitmerchant@hp-notebook: ~/workspace/breeze
File Edit View Search Terminal Tabs Help
amitmerchant@hp-notebook: ~/workspa... amitmerchant@hp-notebook: ~/workspa...
amitmerchant@hp-notebook:~/workspace/breeze $ vendor/bin/phpunit
PHPUnit 9.4.3 by Sebastian Bergmann and contributors.

..... 17 / 17 (100%)

Time: 00:03.378, Memory: 34.00 MB

OK (17 tests, 27 assertions)
amitmerchant@hp-notebook:~/workspace/breeze $
```



## Question 5

**Quelle méthode est appelée avant chaque méthode de test dans PHPUnit ?**

- A. `beforeTest()`
- B. `setup()`
- C. `setUpBeforeClass()`
- D. `setUp()`

**Réponse : D. setUp()**

### **Frequently used Methods**

1. assertEquals()
2. assertInstanceOf()
3. setUp() & tearDown()
4. setUpBeforeClass() & tearDownAfterClass()
5. @dataProvider
6. @expectedException & @expectedExceptionMessage

## Question 6

**Quelle méthode est appelée après chaque méthode de test dans PHPUnit ?**

- A. `tearDownTest()`
- B. `tearDown()`
- C. `tearDownAfterClass()`
- D. `tearDownEach()`

Réponse : B. tearDown()



ZF2 PHPUnit

## Fixtures

**setUp()** method – is called **before** a test method run

**tearDown()** method – is called **after** a test method run

setUp() and tearDown() will be called once for each test method run.



## Question 7

**Quelle assertion est utilisée pour vérifier si une variable est nulle ?**

- A. `assertNull()`
- B. `assertEmpty()`
- C. `assertIsNull()`
- D. `assertNotNull()`

Réponse : A. assertNull()

## Sharing fixture

```
class FooTest extends PHPUnit_Framework_TestCase {
    protected $Foo;

    protected function setUp() {
        $this->Foo = new Foo;
    }

    public function testFoo () {
        $this->assertNull($this->Foo->method());
    }

    public function testFool() {
        $this->assertNull($this->Foo->otherMethod());
    }
}
```

## Question 8

**Quelle assertion est utilisée pour vérifier qu'un tableau contient une clé spécifique ?**

- A. `assertArrayHasKey()`
- B. `assertContains()`
- C. `assertKeyExists()`
- D. `assertHasKey()`

Réponse : A. assertArrayHasKey()

When **assertions** evaluate to **FALSE** the test will emit a **FAILURE**

```
$this->assertTrue(false);  
$this->assertFalse(true);  
$this->assertEquals('foo', 'bar');  
$this->assertEmpty(array('foo'));  
  
$this->assertIsSame(new stdClass, new stdClass);  
  
$this->assertArrayHasKey('foo', array());  
  
$this->assertCount(2, array('apple'));
```



## Question 9

**Quelle méthode est utilisée pour définir des configurations de test communes pour tous les tests d'une classe ?**

- A. `setUp()`
- B. `setUpBeforeClass()`
- C. `init()`
- D. `startUp()`

Réponse : B. setUpBeforeClass()

**setUpBeforeClass() & tearDownAfterClass()**

*setUpBeforeClass()* method is executed only once per class, and even before object is constructed, and that is the reason why it is marked as static public function.

Similarly, *tearDownAfterClass()* method is called after all tests in class finish, and after last *tearDown()* method is called.

## Question 10

**Comment vérifier si une exception spécifique est lancée dans un test PHPUnit ?**

- A. `@expectedException`
- B. `@throwsException`
- C. `@expectException`
- D. `@exception`

## Réponse : C. @expectException

```
/**
 * @expectException Test_Exception
 */
public function testDivByZero()
{
    try {
        // Fyi you don't need to do an assert test here, as we are only testing the exception
        $result = $this->object->div(1,0);
    } catch (Exception $e) {
        if ('Exception' === get_class($e)) {
            throw new Test_Exception($e->getMessage(), $e->getCode());
        }
    }
}

// Test_Exception.php
class Test_Exception extends Exception
{
    public function __construct($message = null, $code = 0, Exception $previous = null)
    {
        parent::__construct($message, $code, $previous);
    }
}
```

## Question 11

**Quelle méthode est utilisée pour effectuer un nettoyage commun après tous les tests d'une classe ?**

- A. `tearDown()`
- B. `tearDownAfterClass()`
- C. `finish()`
- D. `cleanUp()`

## Réponse : B. tearDownAfterClass()

```
<?php
class LogTest extends TestCase
{
    protected static $Log;

    public static function setUpBeforeClass()
    {
        self::$Log = Log::Create( ... );
    }

    public static function tearDownAfterClass()
    {
        $Log = self::$Log;
        $Log->Delete();
    }
}
?>
```

[Practice.org](#)

## Question 12

**Quel attribut d'annotation peut être utilisé pour désactiver temporairement un test ?**

- A. @skip
- B. @disabled
- C. @ignore
- D. @skipTest

## Réponse : C. @ignore

```
/**  
 * @ignore Old test that needs rewriting  
 */  
class TestSkipped extends TestCase  
{  
}
```





## Question 13

**Quelle assertion est utilisée pour vérifier qu'une variable est de type chaîne de caractères ?**

- A. `assertIsString()`
- B. `assertString()`
- C. `assertIsType('string')`
- D. `assertType('string')`

## Réponse : A. assertIsString()

```
shaon@Shaon-Home:~/Projects/phpunittest$ php artisan test
PHP Warning: Module "intl" is already loaded in Unknown on line 0
PHP Warning: Module "intl" is already loaded in Unknown on line 0

PASS Tests\Unit\ExampleTest
✓ that true is true

WARN Tests\Feature\CarbonPackageTest                                0.05s
✓ today
✓ add days
✓ basic assertions
✓ array assertions
✓ equality assertions
✓ string assertions
✓ comparison assertions
✓ file assertions
✓ directory assertions
✓ exception assertions
- miscellaneous assertions → This test is not yet implemented.

PASS Tests\Feature\ExampleTest                                       0.02s
✓ the application returns a successful response

Tests: 1 incomplete, 12 passed (75 assertions)
Duration: 0.14s

shaon@Shaon-Home:~/Projects/phpunittest$
```

## Question 14

**Comment regrouper des tests dans PHPUnit pour les exécuter ensemble ?**

- A. @group
- B. @testGroup
- C. @groupTest
- D. @testSuite

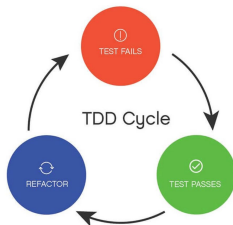
**Réponse : A. @group**

## PHPUnit - XML Configuration

- So, how can I group tests at will when using XML to set up the test suite?
- @group test annotation
- --group, --exclude-group command line options

# Utilisation des Connaissances Acquises

Pour utiliser PHPUnit, voici un exemple de code illustrant les concepts clés :



## Installation de PHPUnit via Composer

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

## Exemple de test PHPUnit

```
<?php
use PHPUnit\Framework\TestCase;

class CalculatorTest extends TestCase {
    protected $calculator;

    protected function setUp(): void {
        $this->calculator = new Calculator();
    }
}
```

```
public function testAdd() {  
    $result = $this->calculator->add(2, 3);  
    $this->assertEquals(5, $result);  
}
```

```
public function testSubtract() {  
    $result = $this->calculator->subtract(5, 3);  
    $this->assertEquals(2, $result);  
}
```

```
protected function tearDown(): void {  
    unset($this->calculator);  
}  
}
```

Pour exécuter les tests, utilisez la commande suivante :

```
php vendor/bin/phpunit
```