PHPUnit PHP testing framework

CONCEPTS

A test is a public method named test* or marked #[Test] in a class extending TestCase whose file ends in Test.php.

Coverage needs a driver: Xdebug (xdebug.mode=coverage) or PCOV. Without one, coverage options only warn.

RUN AND SELECT

vendor/bin/phpunit
Run the configured tests

vendor/bin/phpunit *<path>*
Run one file or directory

vendor/bin/phpunit --filter *<pattern>*
Run tests whose name matches

--testsuite *<name>*
Run one suite from the config

--group *<name>*
Run tests in a group

EXECUTION

--stop-on-failure
Stop at the first failure

OUTPUT

--testdox
Show results as readable sentences

--display-deprecations
Show deprecation details

COVERAGE

--coverage-text
Print a coverage table

--coverage-html *<dir>*
Write an HTML report

XDEBUG_MODE=coverage vendor/bin/phpunit --coverage-html *<dir>*
Coverage with Xdebug

CONFIGURATION

--generate-configuration
Interactively create phpunit.xml

WRITING TESTS

A basic test

```
use PHPUnit\Framework\TestCase;

final class CartTest extends TestCase
{
    protected function setUp(): void
{ $this→cart = new Cart; }

    #[Test]
    public function adds_items(): void
    {
        $this→cart→add('apple');
        $this→assertCount(1, $this-
>cart);
    }
}
```

assertSame($expected, $actual)
Same type and value (===)

assertCount($n, $items)
Number of elements

$this->expectException(*<Class>***::class)**
Expect a thrown exception

ATTRIBUTES

#[Test]
Mark a method as a test

#[DataProvider('*<method>***')]**
Feed data sets from a static method

Data provider

```
#[DataProvider('sums')]
public function test_sum(int $a, int $b,
int $want): void
{
    $this→assertSame($want, $a + $b);
}

public static function sums(): array
{
    return ['zero' ⇒ [0, 0, 0], 'one'
⇒ [1, 0, 1]];
}
```