

# PHPUnit korean

PHPUnit

# Contents

PHPUnitManual:1	1
PHPUnitManual:2	3
PHPUnitManual:3	4
PHPUnitManual:4	8
PHPUnitManual:4.1	9
PHPUnitManual:4.2	11
PHPUnitManual:4.3	11
PHPUnitManual:4.4	8
PHPUnitManual:4.5	2
PHPUnitManual:4.6	5
PHPUnitManual:4.7	8
PHPUnitManual:4.8	6
PHPUnitManual:5	6
PHPUnitManual:5.1	3
PHPUnitManual:6	6
PHPUnitManual:6.1	9
PHPUnitManual:6.2	9
PHPUnitManual:6.3	9
PHPUnitManual:6.4	7
PHPUnitManual:7	7
PHPUnitManual:7.1	7
PHPUnitManual:7.2	4
PHPUnitManual:8	5
PHPUnitManual:8.1	5
PHPUnitManual:8.2	6
PHPUnitManual:8.3	7
PHPUnitManual:8.4	8
PHPUnitManual:8.5	8
PHPUnitManual:8.6	4
PHPUnitManual:8.7	5
PHPUnitManual:8.8	8
PHPUnitManual:9	8
PHPUnitManual:9.1	9
PHPUnitManual:9.2	9
PHPUnitManual:9.3	9
PHPUnitManual:10	3



PHPUnitManual:19.5	1
PHPUnitManual:A	5
PHPUnitManual:B	4
PHPUnitManual:C	5
PHPUnitManual:C.1	6
PHPUnitManual:C.2	9
PHPUnitManual:C.3	6
PHPUnitManual:C.4	7
PHPUnitManual:C.5	6
PHPUnitManual:C.6	7
PHPUnitManual:C.7	6
PHPUnitManual:C.8	8
PHPUnitManual:D	6
PHPUnitManual:E	9
PHPUnitManual:F	7

## References

Article Sources and Contributors	1
Image Sources, Licenses and Contributors	9
	9
	3

# PHPUnitManual:1

1

가

PHPUnit

(testing)

(performing a battery of tests)

가

가

(unit)

가

unit test

print

가 PHP

array

count() 가

, count()

0

가 가

count() 1

Example 1.1

Example 1.1

```
<?php
$fixture = array();
// $fixture is expected to be empty.
$fixture[] = 'element';
// $fixture is expected to contain one element.
?>
```

가

가

가

count()

( 1.2 print

).

가 0 1

array count() 가

Example 1.2 print

```
<?php
$fixture = array();
print count($fixture) . "\n";
$fixture[] = 'element';
print count($fixture) . "\n";
?>
```

0

1

( )

1.3

ok

not ok

가 가 .

Example 1.3

```
<?php
$fixture = array();
print count($fixture) == 0 ? "ok\n": "not ok\n";
$fixture[] = 'element';
print count($fixture) == 0 ? "ok\n": "not ok\n";
?>
```

ok  
ok

가 , 2가  
( 1.4 (assertion) ). 가

Example 1.4 Assertion

```
<?php
$fixture = array();
assertTrue(count($fixture) == 0);
$fixture[] = 'element';
assertTrue(count($fixture) == 1);
function assertTrue($condition)
{
    if (!$condition){
        throw new Exception('Assertion failed.');
```

가 , (refactoring) 가  
(Cross-Team Test), 가

# PHPUnitManual:2

---

## 2 PHPUnit

### PHPUnit

1                    array    count()                    . PHP 가                    array\_\*()

                  가                    ,                    .

                  . PHPUnit                    .

PHPUnit                    가                    .                    가

- .
- ,                    .
- ,                    .
- .
- .
- ,                    가                    .
- ,                    .
- .
- .
- 가                    .                    가                    ,
- 가                    2                    .
- /                    .
- ,                    가                    ,
- /                    .
- 가                    ,                    가                    ,
- ( :                    )

PHPUnit                    PHP                    .                    PHP                    ,

                  PHP                    .

, 가 .  
 PHPUnit 가 가 .  
 가 .  
 PHPUnit 가 가 .  
 PHPUnit 가 가 .  
 PHPUnit 가 가 .  
 가 ,PHPUnit  
 PHPUnit public interface . public

# PHPUnitManual:3

---

## 3 PHPUnit

PHPUnit 3 가 . PEAR [1] Composer [2] PHPUnit  
 (dependencies) ,PHPUnit PHP Archive (PHAR) [3]  
 . PHAR PHPUnit ( )  
 Composer PHP Archive (PHAR) PHPUnit 3.7 ( )  
 PHPUnit 3.7 ). PHPUnit  
 PHPUnit 3.7 PHP 5.3.3 , PHP 5.4.7  
 PHPUnit Code Coverage PHP\_CodeCoverage  
 PHP\_CodeCoverage Xdebug 2.0.5 . Xdebug 2.2.0



## PEAR

2 , (root ) PHPUnit PEAR

```
localhost # pear config-set auto_discover 1
localhost # pear install pear.phpunit.de/PHPUnit
```

OS PHP , PEAR 가  
 PEAR , sudo pear upgrade PEAR . PEAR  
 PEAR [4]

## COMPOSER

PHPUnit ,  
 composer.json phpunit/phpunit  
 composer.json , PHPUnit 3.7

```
{
  "require-dev": {
    "phpunit/phpunit": "3.7.*"
  }
}
```

Composer phpunit  
 composer.json  
 stand-alone composer.json

```
{
  "require-dev": {
    "phpunit/phpunit": "3.7.*"
  },
  "config": {
    "bin-dir": "/usr/local/bin/"
  }
}
```

## PHP Archive (PHAR)

PHPUnit PHP Archive (PHAR) . PHAR PHPUnit  
 ( ) :

```
localhost # wget http://pear.phpunit.de/get/phpunit.phar
localhost # chmod +x phpunit.phar
```

가 .

<b>DbUnit</b>	DbUnit PHP/PHPUnit . 가 . PEAR . localhost # <i>pear install phpunit/DbUnit</i> composer "require-dev" 가 . "phpunit/dbunit": ">=1.2"
<b>PHP_Invoker</b>	callable timeout . timeout . localhost # <i>pear install phpunit/PHP_Invoker</i> composer "require-dev" 가 . "phpunit/php-invoker": "*"
<b>PHPUnit_Selenium</b>	PHPUnit Selenium RC integration. localhost # <i>pear install phpunit/PHPUnit_Selenium</i> composer "require-dev" 가 . "phpunit/php-selenium": ">=1.2"
<b>PHPUnit_Story</b>	PHPUnit Behavior-Driven Development Story-based test runner. localhost # <i>pear install phpunit/PHPUnit_Story</i> composer "require-dev" 가 . "phpunit/phpunit-story": "*"
<b>PHPUnit_SkeletonGenerator</b>	localhost # <i>pear install phpunit/PHPUnit_SkeletonGenerator</i>
<b>PHPUnit_TestListener_DBUS</b>	DBUS listener. localhost # <i>pear install phpunit/PHPUnit_TestListener_DBUS</i>
<b>PHPUnit_TestListener_XHProf</b>	XHProf listener. localhost # <i>pear install phpunit/PHPUnit_TestListener_XHProf</i>

<b>PHPUnit_TicketListener_Fogbugz</b>	Fogbugz (issue) API ticket listener. localhost # pear install phpunit/PHPUnit_TicketListener_Fogbugz
<b>PHPUnit_TicketListener_GitHub</b>	GitHub (issue) API ticket listener. localhost # pear install phpunit/PHPUnit_TicketListener_GitHub
<b>PHPUnit_TicketListener_GoogleCode</b>	Google Code (issue) API ticket listener. localhost # pear install phpunit/PHPUnit_TicketListener_GoogleCode
<b>PHPUnit_TicketListener_Trac</b>	Trac (issue) API ticket listener. localhost # pear install phpunit/PHPUnit_TicketListener_Trac

PHPUnit 3.6      PHPUnit 3.7      가

<b>OutputTestCase</b> 가	PHPUnit_Extensions_OutputTestCase 가 . PHPUnit 3.6 notice " "
가	(cwd) , PHPUnit Code Coverage 가 . 3.7 가
<b>listener</b> 가 <b>autoload</b>	" listener" listener , PHPUnit listener 가 listener 가 , 3.7 autoload listener 가 , aotoload . Autoloader 가 listener 가 listener , bootstrap.php
<b>mock object</b> 가	mock , 가 가 가 method 가 . (swich back) . 10.14 , 10.15
<b>addUncoveredFilesFromWhitelist</b> 가 , <b>processUncoveredFilesFromWhitelist</b>	Code Coverage <whitelistaddUncoveredFilesFromWhitelist="true"> , PHPUnit (include) . , whitelist , PHPUnit 3.7 가 , , PHPUnit 3.6 3.7 , <whitelist processUncoveredFilesFromWhitelist=="true"> 가
<b>cacheTokens</b> Default value <b>false</b>	PHPUnit 3.7.2 (cache) , whitelist , code base 가 가 , phpunit.xml cacheTokens="true" 가 "PHPUnit"

Notes

# PHPUnitManual:4

---

4 PHPUnit

4.1 "PHPUnit" PHP PHPUnit

1. Class ClassTest .
2. ClassTest ( ) PHPUnit\_Framework\_TestCase .
3. test\* public methods .
4. Docblock @test 가 .
4. assertEquals() (assertion) . (Assertions )

4.1 PHPUnit

```

<?php
class StackTest extends PHPUnit_Framework_TestCase
{
    public function testPushAndPop()
    {
        $stack = array();
        $this->assertEquals(0, count($stack));
        array_push($stack, 'foo');
        $this->assertEquals('foo', $stack[count($stack)-1]);
        $this->assertEquals(1, count($stack));
        $this->assertEquals('foo', array_pop($stack));
        $this->assertEquals(0, count($stack));
    }
}
?>
```

Whenever you are tempted to type something into a print statement or a debugger expression, write it as a test instead.

print 가 , - Martin Fowler

# PHPUnitManual:4.1

## 4.1

Unit Tests are primarily written as a good practice to help developers identify and fix bugs, to refactor code and to serve as documentation for a unit of software under test. To achieve these benefits, unit tests ideally should cover all the possible paths in a program. One unit test usually covers one specific path in one function or method. However a test method is not necessary an encapsulated, independent entity. Often there are implicit dependencies between test methods, hidden in the implementation scenario of a test.

(unit test) . , , , , 가 , (route) . 가 , , 가 . - Adrian Kuhn et. al.

PHPUnit 가 (producer) 가 fixture fixture (return) , (consumer) 가 .

- , 가
- , 가

### 4.2 @depends

```
<?php
class StackTest extends PHPUnit_Framework_TestCase
{
    public function testEmpty()
    {
        $stack = array();
        $this->assertEmpty($stack);
        return $stack;
    }
    /**
     * @depends testEmpty
     */
    public function testPush(array $stack)
    {
        array_push($stack, 'foo');
        $this->assertEquals('foo', $stack[count($stack)-1]);
        $this->assertNotEmpty($stack);
        return $stack;
    }
    /**
     * @depends testPush
     */
```

```

public function testPop(array $stack)
{
    $this->assertEquals('foo', array_pop($stack));
    $this->assertEmpty($stack);
}
}
?>

```

```

, testEmpty()
fixture . testPush() testEmpty()
. testPop() testPush()
, PHPUnit
가 , 4.3 "
"
4.3

```

```

<?php
class DependencyFailureTest extends PHPUnit_Framework_TestCase
{
    public function testOne()
    {
        $this->assertTrue(FALSE);
    }
    /**
     * @depends testOne
     */
    public function testTwo()
    {
    }
}
?>

```

```

PHPUnit 3.7.0 by Sebastian Bergmann.
FS
Time: 0 seconds, Memory: 5.00Mb
There was 1 failure:
1) DependencyFailureTest::testOne
Failed asserting that false is true.
/home/sb/DependencyFailureTest.php:6
There was 1 skipped test:
1) DependencyFailureTest::testTwo
This test depends on "DependencyFailureTest::testOne" to pass.
FAILURES!
Tests: 1, Assertions: 1, Failures: 1, Skipped: 1.phpunit --verbose
DependencyFailureTest
PHPUnit 3.7.0 by Sebastian Bergmann.
FS
Time: 0 seconds, Memory: 5.00Mb

```

```

There was 1 failure:
1) DependencyFailureTest::testOne
Failed asserting that false is true.
/home/sb/DependencyFailureTest.php:6
There was 1 skipped test:
1) DependencyFailureTest::testTwo
This test depends on "DependencyFailureTest::testOne" to pass.
FAILURES!
Tests: 1, Assertions: 1, Failures: 1, Skipped: 1.
    
```

```

        @depends
        ,      가
        .PHPUnit      가
    
```

## PHPUnitManual:4.2

---

4.2

### (Providers)

```

        " provider() )
        @dataProvider
        , public
        (Iterator 가
        , type , Object
        ) 가
        가
    
```

4.4

```

<?php
class DataTest extends PHPUnit_Framework_TestCase
{
    /**
     * @dataProvider provider
     */
    public function testAdd($a, $b, $c)
    {
        $this->assertEquals($c, $a + $b);
    }
    public function provider()
    {
        return array(
            array(0, 0, 0),
            array(0, 1, 1),
            array(1, 0, 1),
            array(1, 1, 3)
        );
    }
}
    
```

```

}
?>

PHPUnit 3.7.0 by Sebastian Bergmann.
...F
Time: 0 seconds, Memory: 5.75Mb
There was 1 failure:
1) DataTest::testAdd with data set #3 (1, 1, 3)
Failed asserting that 2 matches expected 3.
/home/sb/DataTest.php:9
FAILURES!
Tests: 4, Assertions: 4, Failures: 1.phpunit DataTest
PHPUnit 3.7.0 by Sebastian Bergmann.
...F
Time: 0 seconds, Memory: 5.75Mb
There was 1 failure:
1) DataTest::testAdd with data set #3 (1, 1, 3)
Failed asserting that 2 matches expected 3.
/home/sb/DataTest.php:9
FAILURES!
Tests: 4, Assertions: 4, Failures: 1.

```

#### 4.5 Iterator Object

```

<?php
require 'CsvFileIterator.php';
class DataTest extends PHPUnit_Framework_TestCase
{
    /**
     * @dataProvider provider
     */
    public function testAdd($a, $b, $c)
    {
        $this->assertEquals($c, $a + $b);
    }
    public function provider()
    {
        return new CsvFileIterator('data.csv');
    }
}
?>

PHPUnit 3.7.0 by Sebastian Bergmann.
...F
Time: 0 seconds, Memory: 5.75Mb
There was 1 failure:
1) DataTest::testAdd with data set #3 ('1', '1', '3')
Failed asserting that 2 matches expected '3'.

```



```

/home/sb/DataTest.php:11
FAILURES!
Tests: 4, Assertions: 4, Failures: 1.phpunit DataTest
PHPUnit 3.7.0 by Sebastian Bergmann.
...F
Time: 0 seconds, Memory: 5.75Mb
There was 1 failure:
1) DataTest::testAdd with data set #3 ('1', '1', '3')
Failed asserting that 2 matches expected '3'.
/home/sb/DataTest.php:11
FAILURES!
Tests: 4, Assertions: 4, Failures: 1.

```

#### 4.6 CsvFileIterator

```

<?php
class CsvFileIterator implements Iterator {
    protected $file;
    protected $key = 0;
    protected $current;
    public function __construct($file) {
        $this->file = fopen($file, 'r');
    }
    public function __destruct() {
        fclose($this->file);
    }
    public function rewind() {
        rewind($this->file);
        $this->current = fgetcsv($this->file);
        $this->key = 0;
    }
    public function valid() {
        return !feof($this->file);
    }
    public function key() {
        return $this->key;
    }
    public function current() {
        return $this->current;
    }
    public function next() {
        $this->current = fgetcsv($this->file);
        $this->key++;
    }
}
?>

```

```

        @dataProvider
        ,
        @depends
        가
        .
        ""
        가
        data set
        가
        ,
        .
        ,
        setUpBeforeClass
        setUp
        가
        ,
        ,
        , PHPUnit 가
    
```

**Notes**

# PHPUnitManual:4.3

---

4.3

4.7 "@expectedException" , 가 throw  
 @expectedException  
 4.7 "@expectedException"

```

<?php
class ExceptionTest extends PHPUnit_Framework_TestCase
{
    /**
     * @expectedException InvalidArgumentException
     */
    public function testException()
    {
    }
}
?>
    
```

```

PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 4.75Mb
There was 1 failure:
1) ExceptionTest::testException
Expected exception InvalidArgumentException
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.phpunit ExceptionTest
PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 4.75Mb
    
```

```
There was 1 failure:
1) ExceptionTest::testException
Expected exception InvalidArgumentException
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.
```

```
, @expectedExceptionMessage @expectedExceptionCode @expectedException
4.8 "@expectedExceptionMessage @expectedExceptionCode
"
4.8 @expectedExceptionMessage @expectedExceptionCode
```

```
<?php
class ExceptionTest extends PHPUnit_Framework_TestCase
{
    /**
     * @expectedException
     * InvalidArgumentException
     * @expectedExceptionMessage Right Message
     */
    public function testExceptionHasRightMessage()
    {
        throw new InvalidArgumentException('Some Message', 10);
    }
    /**
     * @expectedException
     * InvalidArgumentException
     * @expectedExceptionCode 20
     */
    public function testExceptionHasRightCode()
    {
        throw new InvalidArgumentException('Some Message', 10);
    }
}
?>
```

```
PHPUnit 3.7.0 by Sebastian Bergmann.
FF
Time: 0 seconds, Memory: 3.00Mb
There were 2 failures:
1) ExceptionTest::testExceptionHasRightMessage
Failed asserting that exception message 'Some Message' contains 'Right
Message'.
2) ExceptionTest::testExceptionHasRightCode
Failed asserting that expected exception code 20 is equal to 10.
FAILURES!
Tests: 2, Assertions: 4, Failures: 2.phpunit ExceptionTest
PHPUnit 3.7.0 by Sebastian Bergmann.
```

```
FF
Time: 0 seconds, Memory: 3.00Mb
There were 2 failures:
1) ExceptionTest::testExceptionHasRightMessage
Failed asserting that exception message 'Some Message' contains 'Right
Message'.
2) ExceptionTest::testExceptionHasRightCode
Failed asserting that expected exception code 20 is equal to 10.
FAILURES!
Tests: 2, Assertions: 4, Failures: 2.
```

```
@expectedExceptionMessage @expectedExceptionCode 가 "@expectedExceptionMessage"
"@expectedExceptionCode"
, setExpectedException()
4.9 " "
4.9
```

```
<?php
class ExceptionTest extends PHPUnit_Framework_TestCase
{
    public function testException()
    {
        $this->setExpectedException('InvalidArgumentException');
    }
    public function testExceptionHasRightMessage()
    {
        $this->setExpectedException(
            'InvalidArgumentException', 'Right Message'
        );
        throw new InvalidArgumentException('Some Message', 10);
    }
    public function testExceptionHasRightCode()
    {
        $this->setExpectedException(
            'InvalidArgumentException', 'Right Message', 20
        );
        throw new InvalidArgumentException('The Right Message', 10);
    }
}
}??>
```

```
PHPUnit 3.7.0 by Sebastian Bergmann.
FFF
Time: 0 seconds, Memory: 3.00Mb
There were 3 failures:
1) ExceptionTest::testException
Expected exception InvalidArgumentException
2) ExceptionTest::testExceptionHasRightMessage
```

```

Failed asserting that exception message 'Some Message' contains 'Right
Message'.
3) ExceptionTest::testExceptionHasRightCode
Failed asserting that expected exception code 20 is equal to 10.
FAILURES!
Tests: 3, Assertions: 6, Failures: 3.phpunit ExceptionTest
PHPUnit 3.7.0 by Sebastian Bergmann.
FFF
Time: 0 seconds, Memory: 3.00Mb
There were 3 failures:
1) ExceptionTest::testException
Expected exception InvalidArgumentException
2) ExceptionTest::testExceptionHasRightMessage
Failed asserting that exception message 'Some Message' contains 'Right
Message'.
3) ExceptionTest::testExceptionHasRightCode
Failed asserting that expected exception code 20 is equal to 10.
FAILURES!
Tests: 3, Assertions: 6, Failures: 3.

```

4.1 "

"

.

void setExpectedException(string \$exceptionName[, string \$exceptionMessage = , integer \$exceptionCode = NULL])	\$exceptionName, \$exceptionMessage \$exceptionCode
String getExpectedException()	
4.1	

, 4.10 "

"

.

4.10

```

<?php
class ExceptionTest extends PHPUnit_Framework_TestCase {
    public function testException() {
        try {
            // ...
        }
        catch (InvalidArgumentException $expected) {
            return;
        }
        $this->fail('가 가 .');
    }
}
?>

```

가

4.10 "

"

가

,

fail()

가

가

.

가

, catch

가

## Notes

# PHPUnitManual:4.4

---

### 4.4 PHP

## PHP

PHPUnit 3.7.0 by Sebastian Bergmann.  
 .  
 Time: 0 seconds  
 OK (1 test, 1 assertion)phpunit ExpectedErrorTest  
 PHPUnit 3.7.0 by Sebastian Bergmann.  
 .  
 Time: 0 seconds  
 OK (1 test, 1 assertion)

```

<?php
class ExpectedErrorTest extends PHPUnit_Framework_TestCase
{
    /**
     * @expectedException PHPUnit_Framework_Error
     */
    public function testFailingInclude()
    {
        include 'not_existing_file.php';
    }
}
?>

```

```

PHPUnit 3.7.0 by Sebastian Bergmann.
.
Time: 0 seconds
OK (1 test, 1 assertion)phpunit ExpectedErrorTest
PHPUnit 3.7.0 by Sebastian Bergmann.
.
Time: 0 seconds
OK (1 test, 1 assertion)

```

PHPUnit\_Framework\_Error\_Notice PHPUnit\_Framework\_Error\_Warning , PHP

가  
 @expectedException  
 setExpectedException() Exception  
 PHP ( fopen )  
 notice  
 PHPUnit\_Framework\_Error\_Notice  
 4.12 PHP 가

```

<?php
class ErrorSuppressionTest extends PHPUnit_Framework_TestCase
{
    public function testFileWriting() {
        $writer = new FileWriter;
        $this->assertFalse(@$writer->write('/is-not-writeable/file',
'stuff'));
    }
}
class FileWriter
{
    public function write($file, $content) {
        $file = fopen($file, 'w');
        if($file == false) {
            return false;
        }
        // ...
    }
}
?>

```

```

PHPUnit 3.7.0 by Sebastian Bergmann.
.
Time: 1 seconds, Memory: 5.25Mb
OK (1 test, 1 assertion)phpunit ErrorSuppressionTest
PHPUnit 3.7.0 by Sebastian Bergmann.
.
Time: 1 seconds, Memory: 5.25Mb
OK (1 test, 1 assertion)

```

file or directory 가 , fopen(/is-not-writeable/file): failed to open stream: No such

## Notes

# PHPUnitManual:4.5

---

4.5

```

        , (echo print )
        . PHPUnit_Framework_TestCase PHP [1]
    
```

4.13 " " expectedOutputString()

4.13

```

<?php
class OutputTest extends PHPUnit_Framework_TestCase
{
    public function testExpectFooActualFoo()
    {
        $this->expectOutputString('foo');
        print 'foo';
    }
    public function testExpectBarActualBaz()
    {
        $this->expectOutputString('bar');
        print 'baz';
    }
}
?>
    
```

```

PHPUnit 3.7.0 by Sebastian Bergmann.
.F
Time: 0 seconds, Memory: 5.75Mb
There was 1 failure:
1) OutputTest::testExpectBarActualBaz
Failed asserting that two strings are equal.
--- Expected
+++ Actual
@@ @@
-'bar'
+'baz'
FAILURES!
Tests: 2, Assertions: 2, Failures: 1.phpunit OutputTest
PHPUnit 3.7.0 by Sebastian Bergmann.
.F
Time: 0 seconds, Memory: 5.75Mb
There was 1 failure:
1) OutputTest::testExpectBarActualBaz
Failed asserting that two strings are equal.
    
```



```

--- Expected
+++ Actual
@@ @@
-'bar'
+'baz'
FAILURES!
Tests: 2, Assertions: 2, Failures: 1.
    
```

4.2 "

void expectOutputRegex(string \$regularExpression)	\$regularExpression	.
void expectOutputString(string \$expectedString)	\$expectedString	.
bool setOutputCallback(callable \$callback)		callback
4.2		

## Notes

# PHPUnitManual:4.6

---

4.6

### (assertion)

가

### assertArrayHasKey()

assertArrayHasKey(mixed \$key, array \$array[, string \$message = ])

\$array \$key 가 , \$message .

assertArrayNotHasKey()

4.14 assertArrayHasKey()

```

<?php
class ArrayHasKeyTest extends PHPUnit_Framework_TestCase
{
    public function testFailure()
    {
        $this->assertArrayHasKey('foo', array('bar' => 'baz'));
    }
}
?>
    
```

PHPUnit 3.7.0 by Sebastian Bergmann.

F

Time: 0 seconds, Memory: 5.00Mb

```

There was 1 failure:
1) ArrayHasKeyTest::testFailure
Failed asserting that an array has the key 'foo'.
/home/sb/ArrayHasKeyTest.php:6
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.phpunit ArrayHasKeyTest
PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.00Mb
There was 1 failure:
1) ArrayHasKeyTest::testFailure
Failed asserting that an array has the key 'foo'.
/home/sb/ArrayHasKeyTest.php:6
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.
    
```

### assertClassHasAttribute()

assertClassHasAttribute(string \$attributeName, string \$className[, string \$message = ])

\$className::attributeName, \$message .

assertClassNotHasAttribute(), .

#### 4.15 assertClassHasAttribute()

```

<?php
class ClassHasAttributeTest extends PHPUnit_Framework_TestCase
{
    public function testFailure()
    {
        $this->assertClassHasAttribute('foo', 'stdClass');
    }
}
?>
    
```

```

PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 4.75Mb
There was 1 failure:
1) ClassHasAttributeTest::testFailure
Failed asserting that class "stdClass" has attribute "foo".
/home/sb/ClassHasAttributeTest.php:6
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.phpunit ClassHasAttributeTest
PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 4.75Mb
There was 1 failure:
1) ClassHasAttributeTest::testFailure
    
```

```
Failed asserting that class "stdClass" has attribute "foo".
/home/sb/ClassHasAttributeTest.php:6
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.
```

### assertClassHasStaticAttribute()

assertClassHasStaticAttribute(string \$attributeName, string \$className[, string \$message = ])

\$className::attributeName , \$message .

assertClassNotHasStaticAttribute() , .

#### 4.16 assertClassNotHasAttribute()

```
<?php
class ClassHasStaticAttributeTest extends PHPUnit_Framework_TestCase
{
    public function testFailure()
    {
        $this->assertClassHasStaticAttribute('foo', 'stdClass');
    }
}
?>
```

PHPUnit 3.7.0 by Sebastian Bergmann.

F

Time: 0 seconds, Memory: 4.75Mb

There was 1 failure:

1) ClassHasStaticAttributeTest::testFailure

Failed asserting that class "stdClass" has static attribute "foo".

/home/sb/ClassHasStaticAttributeTest.php:6

FAILURES!

Tests: 1, Assertions: 1, Failures: 1.phpunit ClassHasStaticAttributeTest

PHPUnit 3.7.0 by Sebastian Bergmann.

F

Time: 0 seconds, Memory: 4.75Mb

There was 1 failure:

1) ClassHasStaticAttributeTest::testFailure

Failed asserting that class "stdClass" has static attribute "foo".

/home/sb/ClassHasStaticAttributeTest.php:6

FAILURES!

Tests: 1, Assertions: 1, Failures: 1.

### assertContains()

assertContains(mixed \$needle, Iterator|array \$haystack[, string \$message = ])

\$needle \$haystack 가 , \$message .

assertNotContains() , .

assertAttributeContains() assertAttributeNotContains() wrapper , public,  
protected, private \$haystack .

#### 4.17 assertContains()

```
<?php
class ContainsTest extends PHPUnit_Framework_TestCase
{
    public function testFailure()
    {
        $this->assertContains(4, array(1, 2, 3));
    }
}
?>
```

```
PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.00Mb
There was 1 failure:
1) ContainsTest::testFailure
Failed asserting that an array contains 4.
/home/sb/ContainsTest.php:6
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.phpunit ContainsTest
PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.00Mb
There was 1 failure:
1) ContainsTest::testFailure
Failed asserting that an array contains 4.
/home/sb/ContainsTest.php:6
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.
```

assertContains(string \$needle, string \$haystack[, string \$message = ])

\$needle \$haystack , \$message .

#### 4.18 assertContains()

```
<?php
class ContainsTest extends PHPUnit_Framework_TestCase
{
    public function testFailure()
    {
        $this->assertContains('baz', 'foobar');
    }
}
```

```

    }
}
?>

PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.00Mb
There was 1 failure:
1) ContainsTest::testFailure
Failed asserting that 'foobar' contains "baz".
/home/sb/ContainsTest.php:6
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.phpunit ContainsTest
PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.00Mb
There was 1 failure:
1) ContainsTest::testFailure
Failed asserting that 'foobar' contains "baz".
/home/sb/ContainsTest.php:6
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.

```

**assertContainsOnly()**

assertContainsOnly(string \$type, Iterator|array \$haystack[, boolean \$isNativeType = NULL, string \$message = ])

\$haystack \$type 가 가 , \$message .  
 assertNotContainsOnly() , .  
 assertAttributeContainsOnly() assertAttributeNotContainsOnly() wrapper ,  
 public, protected, private haystack .

4.19 assertContainsOnly()

```

<?php
class ContainsOnlyTest extends PHPUnit_Framework_TestCase
{
    public function testFailure()
    {
        $this->assertContainsOnly('string', array('1', '2', 3));
    }
}
?>

```

```

PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.00Mb
There was 1 failure:
1) ContainsOnlyTest::testFailure
Failed asserting that Array (

```

```

0 => '1'
1 => '2'
2 => 3
) contains only values of type "string".
/home/sb/ContainsOnlyTest.php:6
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.phpunit ContainsOnlyTest
PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.00Mb
There was 1 failure:
1) ContainsOnlyTest::testFailure
Failed asserting that Array (
0 => '1'
1 => '2'
2 => 3
) contains only values of type "string".
/home/sb/ContainsOnlyTest.php:6
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.

```

### assertContainsOnlyInstancesOf()

assertContainsOnlyInstancesOf(string \$classname, Traversable|array \$haystack[, string \$message = ])

\$haystack \$classname , \$message .

#### 4.20 assertContainsOnlyInstancesOf()

```

<?php
class ContainsOnlyInstancesOfTest extends PHPUnit_Framework_TestCase
{
    public function testFailure()
    {
        $this->assertContainsOnlyInstancesOf('Foo', array(new Foo(), new
Bar(), new Foo())
    }
}
?>

```

```

PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.00Mb
There was 1 failure:
1) ContainsOnlyInstancesOfTest::testFailure
Failed asserting that Array ([0]=> Bar Object(...)) is an instance of
class "Foo".
/home/sb/ContainsOnlyInstancesOfTest.php:6
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.phpunit
ContainsOnlyInstancesOfTest

```

```
PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.00Mb
There was 1 failure:
1) ContainsOnlyInstancesOfTest::testFailure
Failed asserting that Array ([0]=> Bar Object(...)) is an instance of
class "Foo".
/home/sb/ContainsOnlyInstancesOfTest.php:6
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.
```

### assertCount()

assertCount(\$expectedCount, \$haystack[, string \$message = ])

\$haystack            가 \$expectedCount            ,            \$message            .

assertNotCount()

#### 4.21 assertCount()

```
<?php
class CountTest extends PHPUnit_Framework_TestCase
{
    public function testFailure()
    {
        $this->assertCount(0, array('foo'));
    }
}
?>
```

```
PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 4.75Mb
There was 1 failure:
1) CountTest::testFailure
Failed asserting that actual size 1 matches expected size 0.
/home/sb/CountTest.php:6
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.phpunit CountTest
```

```
PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 4.75Mb
There was 1 failure:
1) CountTest::testFailure
Failed asserting that actual size 1 matches expected size 0.
/home/sb/CountTest.php:6
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.
```

## assertEmpty()

assertEmpty(mixed \$actual[, string \$message = ])

\$actual, \$message

assertNotEmpty(),

assertAttributeEmpty() assertAttributeNotEmpty() wrapper, public,

protected, private haystack

### 4.22 assertEmpty()

```
<?php
class EmptyTest extends PHPUnit_Framework_TestCase
{
    public function testFailure()
    {
        $this->assertEmpty(array('foo'));
    }
}
?>
```

```
PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 4.75Mb
There was 1 failure:
1) EmptyTest::testFailure
Failed asserting that an array is empty.
/home/sb/EmptyTest.php:6
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.phpunit EmptyTest
PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 4.75Mb
There was 1 failure:
1) EmptyTest::testFailure
Failed asserting that an array is empty.
/home/sb/EmptyTest.php:6
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.
```



## assertEqualXMLStructure()

assertEqualXMLStructure(DOMElement \$expectedElement, DOMElement \$actualElement[, boolean \$checkAttributes = FALSE, string \$message = ])

\$actualElement DOMElement XML 가 \$expectedElement DOMElement XML  
 , \$message .

### 4.23 assertEqualXMLStructure()

```
<?php
class EqualXMLStructureTest extends PHPUnit_Framework_TestCase
{
    public function testFailureWithDifferentNodeNames()
    {
        $expected = new DOMElement('foo');
        $actual = new DOMElement('bar');
        $this->assertEqualXMLStructure($expected, $actual);
    }
    public function testFailureWithDifferentNodeAttributes()
    {
        $expected = new DOMDocument;
        $expected->loadXML('<foo bar="true" />');
        $actual = new DOMDocument;
        $actual->loadXML('<foo/>');
        $this->assertEqualXMLStructure(
            $expected->firstChild, $actual->firstChild, TRUE
        );
    }
    public function testFailureWithDifferentChildrenCount()
    {
        $expected = new DOMDocument;
        $expected->loadXML('<foo><bar/><bar/><bar/></foo>');
        $actual = new DOMDocument;
        $actual->loadXML('<foo><bar/></foo>');
        $this->assertEqualXMLStructure(
            $expected->firstChild, $actual->firstChild
        );
    }
    public function testFailureWithDifferentChildren()
    {
        $expected = new DOMDocument;
        $expected->loadXML('<foo><bar/><bar/><bar/></foo>');
        $actual = new DOMDocument;
        $actual->loadXML('<foo><baz/><baz/><baz/></foo>');
        $this->assertEqualXMLStructure(
            $expected->firstChild, $actual->firstChild
        );
    }
}
```

?>

```

PHPUnit 3.7.0 by Sebastian Bergmann.
FFFF
Time: 0 seconds, Memory: 5.75Mb
There were 4 failures:
1) EqualXMLStructureTest::testFailureWithDifferentNodeNames
Failed asserting that two strings are equal.
--- Expected
+++ Actual
@@ @@
-'foo'
+'bar'
/home/sb/EqualXMLStructureTest.php:9
2) EqualXMLStructureTest::testFailureWithDifferentNodeAttributes
Number of attributes on node "foo" does not match
Failed asserting that 0 matches expected 1.
/home/sb/EqualXMLStructureTest.php:22
3) EqualXMLStructureTest::testFailureWithDifferentChildrenCount
Number of child nodes of "foo" differs
Failed asserting that 1 matches expected 3.
/home/sb/EqualXMLStructureTest.php:35
4) EqualXMLStructureTest::testFailureWithDifferentChildren
Failed asserting that two strings are equal.
--- Expected
+++ Actual
@@ @@
-'bar'
+'baz'
/home/sb/EqualXMLStructureTest.php:48
FAILURES!
Tests: 4, Assertions: 8, Failures: 4.phpunit EqualXMLStructureTest
PHPUnit 3.7.0 by Sebastian Bergmann.
FFFF
Time: 0 seconds, Memory: 5.75Mb
There were 4 failures:
1) EqualXMLStructureTest::testFailureWithDifferentNodeNames
Failed asserting that two strings are equal.
--- Expected
+++ Actual
@@ @@
-'foo'
+'bar'
/home/sb/EqualXMLStructureTest.php:9
2) EqualXMLStructureTest::testFailureWithDifferentNodeAttributes
Number of attributes on node "foo" does not match
Failed asserting that 0 matches expected 1.
/home/sb/EqualXMLStructureTest.php:22

```

```

3) EqualXMLStructureTest::testFailureWithDifferentChildrenCount
Number of child nodes of "foo" differs
Failed asserting that 1 matches expected 3.
/home/sb/EqualXMLStructureTest.php:35
4) EqualXMLStructureTest::testFailureWithDifferentChildren
Failed asserting that two strings are equal.
--- Expected
+++ Actual
@@ @@
-'bar'
+'baz'
/home/sb/EqualXMLStructureTest.php:48
FAILURES!
Tests: 4, Assertions: 8, Failures: 4.

```

### assertEquals()

assertEquals(mixed \$expected, mixed \$actual[, string \$message = ])

2 \$expected \$actual, \$message .

assertNotEquals()

assertAttributeEquals() assertAttributeNotEquals() wrapper , public,  
protected, private .

4.24 assertEquals()

```

<?php
class EqualsTest extends PHPUnit_Framework_TestCase
{
    public function testFailure()
    {
        $this->assertEquals(1, 0);
    }
    public function testFailure2()
    {
        $this->assertEquals('bar', 'baz');
    }
    public function testFailure3()
    {
        $this->assertEquals("foo\nbar\nbaz\n", "foo\nbah\nbaz\n");
    }
}
?>

```

PHPUnit 3.7.0 by Sebastian Bergmann.

FFF

Time: 0 seconds, Memory: 5.25Mb

There were 3 failures:

1) EqualsTest::testFailure

```
Failed asserting that 0 matches expected 1.
/home/sb/EqualsTest.php:6
2) EqualsTest::testFailure2
Failed asserting that two strings are equal.
--- Expected
+++ Actual
@@ @@
-'bar'
+'baz'
/home/sb/EqualsTest.php:11
3) EqualsTest::testFailure3
Failed asserting that two strings are equal.
--- Expected
+++ Actual
@@ @@
'foo
-bar
+bah
baz
'
/home/sb/EqualsTest.php:16
FAILURES!
Tests: 3, Assertions: 3, Failures: 3.phpunit EqualsTest
PHPUnit 3.7.0 by Sebastian Bergmann.
FFF
Time: 0 seconds, Memory: 5.25Mb
There were 3 failures:
1) EqualsTest::testFailure
Failed asserting that 0 matches expected 1.
/home/sb/EqualsTest.php:6
2) EqualsTest::testFailure2
Failed asserting that two strings are equal.
--- Expected
+++ Actual
@@ @@
-'bar'
+'baz'
/home/sb/EqualsTest.php:11
3) EqualsTest::testFailure3
Failed asserting that two strings are equal.
--- Expected
+++ Actual
@@ @@
'foo
-bar
+bah
baz
```

```

,
/home/sb/EqualsTest.php:16
FAILURES!
Tests: 3, Assertions: 3, Failures: 3.

```

```

    $expected    $actual
assertEquals(float $expected, float $actual[, string $message = , float $delta = 0])
2   float  $expected  $actual   가 $delta      ,      $message
$delta 가          "What Every Computer Scientist Should Know About Floating-Point
Arithmetic [1]"
    4.25 float          assertEquals()

```

```

<?php
class EqualsTest extends PHPUnit_Framework_TestCase
{
    public function testSuccess()
    {
        $this->assertEquals(1.0, 1.1, '', 0.2);
    }
    public function testFailure()
    {
        $this->assertEquals(1.0, 1.1);
    }
}
?>

```

```

PHPUnit 3.7.0 by Sebastian Bergmann.
.F
Time: 0 seconds, Memory: 5.75Mb
There was 1 failure:
1) EqualsTest::testFailure
Failed asserting that 1.1 matches expected 1.0.
/home/sb/EqualsTest.php:11
FAILURES!
Tests: 2, Assertions: 2, Failures: 1.phpunit EqualsTest
PHPUnit 3.7.0 by Sebastian Bergmann.
.F
Time: 0 seconds, Memory: 5.75Mb
There was 1 failure:
1) EqualsTest::testFailure
Failed asserting that 1.1 matches expected 1.0.
/home/sb/EqualsTest.php:11
FAILURES!
Tests: 2, Assertions: 2, Failures: 1.

```

assertEquals(DOMDocument \$expected, DOMDocument \$actual[, string \$message = ])

```

2   DOMDocument      $expected $actual      XML   가 (
   )                , $message   .
4.26 DOMDocument   assertEquals()

```

```

<?php
class EqualsTest extends PHPUnit_Framework_TestCase
{
    public function testFailure()
    {
        $expected = new DOMDocument;
        $expected->loadXML('<foo><bar/></foo>');
        $actual = new DOMDocument;
        $actual->loadXML('<bar><foo/></bar>');
        $this->assertEquals($expected, $actual);
    }
}
?>

```

```

PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.00Mb
There was 1 failure:
1) EqualsTest::testFailure
Failed asserting that two DOM documents are equal.
--- Expected
+++ Actual
@@ @@
<?xml version="1.0"?>
-<foo>
- <bar/>
-</foo>
+<bar>
+ <foo/>
+</bar>
/home/sb/EqualsTest.php:12
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.phpunit EqualsTest
PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.00Mb
There was 1 failure:
1) EqualsTest::testFailure
Failed asserting that two DOM documents are equal.
--- Expected
+++ Actual
@@ @@
<?xml version="1.0"?>
-<foo>

```

```

- <bar/>
-</foo>
+<bar>
+ <foo/>
+</bar>
/home/sb/EqualsTest.php:12
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.

```

assertEquals(object \$expected, object \$actual[, string \$message = ])

2            \$expected    \$actual            가            ,            \$message            .

4.27                    assertEquals()

```

<?php
class EqualsTest extends PHPUnit_Framework_TestCase
{
    public function testFailure()
    {
        $expected = new stdClass;
        $expected->foo = 'foo';
        $expected->bar = 'bar';
        $actual = new stdClass;
        $actual->foo = 'bar';
        $actual->baz = 'bar';
        $this->assertEquals($expected, $actual);
    }
}
?>

```

```

PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.25Mb
There was 1 failure:
1) EqualsTest::testFailure
Failed asserting that two objects are equal.
--- Expected
+++ Actual
@@ @@
stdClass Object (
-
'foo' => 'foo'
-
'bar' => 'bar'
+
'foo' => 'bar'
+
'baz' => 'bar'
)

```

```

/home/sb/EqualsTest.php:14
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.phpunit EqualsTest
PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.25Mb
There was 1 failure:
1) EqualsTest::testFailure
Failed asserting that two objects are equal.
--- Expected
+++ Actual
@@ @@
stdClass Object (
-
'foo' => 'foo'
-
'bar' => 'bar'
+
'foo' => 'bar'
+
'baz' => 'bar'
)
/home/sb/EqualsTest.php:14
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.

```

```

assertEquals(array $expected, array $actual[, string $message = ])
2          $expected  $actual          ,      $message
          4.28          assertEquals()

```

```

<?php
class EqualsTest extends PHPUnit_Framework_TestCase
{
    public function testFailure()
    {
        $this->assertEquals(array('a', 'b', 'c'), array('a', 'c',
'd'));
    }
}
?>

```

```

PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.25Mb
There was 1 failure:
1) EqualsTest::testFailure
Failed asserting that two arrays are equal.
--- Expected

```



```
+++ Actual
@@ @@
Array (
0 => 'a'
-
1 => 'b'
-
2 => 'c'
+
1 => 'c'
+
2 => 'd'
)
/home/sb/EqualsTest.php:6
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.phpunit EqualsTest
PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.25Mb
There was 1 failure:
1) EqualsTest::testFailure
Failed asserting that two arrays are equal.
--- Expected
+++ Actual
@@ @@
Array (
0 => 'a'
-
1 => 'b'
-
2 => 'c'
+
1 => 'c'
+
2 => 'd'
)
/home/sb/EqualsTest.php:6
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.
```

### assertFalse()

assertFalse(bool \$condition[, string \$message = ])

\$condition TRUE , \$message

4.29 assertFalse()

```
<?php
class FalseTest extends PHPUnit_Framework_TestCase
{
    public function testFailure()
    {
        $this->assertFalse(TRUE);
    }
}
?>
```

```
PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.00Mb
There was 1 failure:
1) FalseTest::testFailure
Failed asserting that true is false.
/home/sb/FalseTest.php:6
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.phpunit FalseTest
PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.00Mb
There was 1 failure:
1) FalseTest::testFailure
Failed asserting that true is false.
/home/sb/FalseTest.php:6
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.
```

### assertFileEquals()

assertFileEquals(string \$expected, string \$actual[, string \$message = ])

\$expected \$actual , \$message .

assertFileNotEquals() , .

4.30 assertFileEquals()

```
<?php
class FileEqualsTest extends PHPUnit_Framework_TestCase
{
    public function testFailure()
    {
        $this->assertFileEquals('/home/sb/expected',
```

```

'/home/sb/actual');
    }
}
?>

```

```

PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.25Mb
There was 1 failure:
1) FileEqualsTest::testFailure
Failed asserting that two strings are equal.
--- Expected
+++ Actual
@@ @@
-'expected
+'actual
'
/home/sb/FileEqualsTest.php:6
FAILURES!
Tests: 1, Assertions: 3, Failures: 1.phpunit FileEqualsTest
PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.25Mb
There was 1 failure:
1) FileEqualsTest::testFailure
Failed asserting that two strings are equal.
--- Expected
+++ Actual
@@ @@
-'expected
+'actual
'
/home/sb/FileEqualsTest.php:6
FAILURES!
Tests: 1, Assertions: 3, Failures: 1.

```

**assertFileExists()**

assertFileExists(string \$filename[, string \$message = ])

\$filename , \$message .

assertFileNotExists()

, .

4.31 assertFileExists()

```

<?php
class FileExistsTest extends PHPUnit_Framework_TestCase
{
    public function testFailure()

```

```

    {
        $this->assertFileExists('/path/to/file');
    }
}
?>

```

PHPUnit 3.7.0 by Sebastian Bergmann.

```

F
Time: 0 seconds, Memory: 4.75Mb
There was 1 failure:
1) FileExistsTest::testFailure
Failed asserting that file "/path/to/file" exists.
/home/sb/FileExistsTest.php:6
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.phpunit FileExistsTest
PHPUnit 3.7.0 by Sebastian Bergmann.

```

```

F
Time: 0 seconds, Memory: 4.75Mb
There was 1 failure:
1) FileExistsTest::testFailure
Failed asserting that file "/path/to/file" exists.
/home/sb/FileExistsTest.php:6
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.

```

### assertGreaterThan()

assertGreaterThan(mixed \$expected, mixed \$actual[, string \$message = ])

\$actual      \$expected                      ,      \$message                      .  
assertAttributeGreaterThan()                      wrapper      ,                      public, protected, private

#### 4.32 assertAttributeGreaterThan()

```

<?php
class GreaterThanTest extends PHPUnit_Framework_TestCase
{
    public function testFailure()
    {
        $this->assertGreaterThan(2, 1);
    }
}
?>

```

PHPUnit 3.7.0 by Sebastian Bergmann.

```

F
Time: 0 seconds, Memory: 5.00Mb
There was 1 failure:
1) GreaterThanTest::testFailure

```

```
Failed asserting that 1 is greater than 2.
/home/sb/GreaterThanTest.php:6
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.phpunit GreaterThanTest
PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.00Mb
There was 1 failure:
1) GreaterThanTest::testFailure
Failed asserting that 1 is greater than 2.
/home/sb/GreaterThanTest.php:6
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.
```

### assertGreaterThanOrEqual()

assertGreaterThanOrEqual(mixed \$expected, mixed \$actual[, string \$message = ])

\$actual      \$expected                      (                      ),      \$message                      .  
 assertAttributeGreaterThanOrEqual()                      wrapper      ,                      public, protected, private

#### 4.33 assertAttributeGreaterThanOrEqual()

```
<?php
class GreatThanOrEqualTest extends PHPUnit_Framework_TestCase
{
    public function testFailure()
    {
        $this->assertGreaterThanOrEqual(2, 1);
    }
}
?>
```

```
PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.25Mb
There was 1 failure:
1) GreatThanOrEqualTest::testFailure
Failed asserting that 1 is equal to 2 or is greater than 2.
/home/sb/GreaterThanOrEqualTest.php:6
FAILURES!
Tests: 1, Assertions: 2, Failures: 1.phpunit GreaterThanOrEqualTest
PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.25Mb
There was 1 failure:
1) GreatThanOrEqualTest::testFailure
Failed asserting that 1 is equal to 2 or is greater than 2.
```

```
/home/sb/GreaterThanOrEqualTest.php:6
FAILURES!
Tests: 1, Assertions: 2, Failures: 1.
```

### assertInstanceOf()

assertInstanceOf(\$expected, \$actual[, \$message = ])

\$actual \$expected 가 , \$message .

assertNotInstanceOf() , .

assertAttributeInstanceOf() assertAttributeNotInstanceOf() wrapper ,

public, protected, private .

#### 4.34 assertAttributeInstanceOf()

```
<?php
class InstanceOfTest extends PHPUnit_Framework_TestCase
{
    public function testFailure()
    {
        $this->assertInstanceOf('RuntimeException', new Exception);
    }
}
?>
```

PHPUnit 3.7.0 by Sebastian Bergmann.

F

Time: 0 seconds, Memory: 5.00Mb

There was 1 failure:

1) InstanceOfTest::testFailure

Failed asserting that Exception Object (...) is an instance of class "RuntimeException".

/home/sb/InstanceOfTest.php:6

FAILURES!

Tests: 1, Assertions: 1, Failures: 1.phpunit InstanceOfTest

PHPUnit 3.7.0 by Sebastian Bergmann.

F

Time: 0 seconds, Memory: 5.00Mb

There was 1 failure:

1) InstanceOfTest::testFailure

Failed asserting that Exception Object (...) is an instance of class "RuntimeException".

/home/sb/InstanceOfTest.php:6

FAILURES!

Tests: 1, Assertions: 1, Failures: 1.

### assertInternalType()

assertInternalType(\$expected, \$actual[, \$message = ])

\$actual (type) \$expected , \$message .

assertNotInternalType() , .

assertAttributeInternalType() assertAttributeNotInternalType() wrapper ,

public, protected, private .

#### 4.35 assertInternalType()

```
<?php
class InternalTypeTest extends PHPUnit_Framework_TestCase
{
    public function testFailure()
    {
        $this->assertInternalType('string', 42);
    }
}
?>
```

PHPUnit 3.7.0 by Sebastian Bergmann.

F

Time: 0 seconds, Memory: 5.00Mb

There was 1 failure:

1) InternalTypeTest::testFailure

Failed asserting that 42 is of type "string".

/home/sb/InternalTypeTest.php:6

FAILURES!

Tests: 1, Assertions: 1, Failures: 1.phpunit InternalTypeTest

PHPUnit 3.7.0 by Sebastian Bergmann.

F

Time: 0 seconds, Memory: 5.00Mb

There was 1 failure:

1) InternalTypeTest::testFailure

Failed asserting that 42 is of type "string".

/home/sb/InternalTypeTest.php:6

FAILURES!

Tests: 1, Assertions: 1, Failures: 1.

### assertJsonFileEqualsJsonFile()

assertJsonFileEqualsJsonFile(mixed \$expectedFile, mixed \$actualFile[, string \$message = ])

\$actualFile \$expectedFile , \$message .

#### 4.36 assertJsonFileEqualsJsonFile()

```
<?php
class JsonFileEqualsJsonFileTest extends PHPUnit_Framework_TestCase
{
    public function testFailure()
```

```
{
    $this->assertJsonFileEqualsJsonFile(
        'path/to/fixture/file', 'path/to/actual/file');
}
?>
```

PHPUnit 3.7.0 by Sebastian Bergmann.  
 F  
 Time: 0 seconds, Memory: 5.00Mb  
 There was 1 failure:  
 1) JsonFileEqualsJsonFile::testFailure  
 Failed asserting that '{"Mascott": "Tux"}' matches JSON string  
 '{"Mascott", "Tux", "OS", "  
 /home/sb/JsonFileEqualsJsonFileTest.php:5  
 FAILURES!  
 Tests: 1, Assertions: 3, Failures: 1.phpunit JsonFileEqualsJsonFileTest  
 PHPUnit 3.7.0 by Sebastian Bergmann.

F  
 Time: 0 seconds, Memory: 5.00Mb  
 There was 1 failure:  
 1) JsonFileEqualsJsonFile::testFailure  
 Failed asserting that '{"Mascott": "Tux"}' matches JSON string  
 '{"Mascott", "Tux", "OS", "  
 /home/sb/JsonFileEqualsJsonFileTest.php:5  
 FAILURES!  
 Tests: 1, Assertions: 3, Failures: 1.

**assertJsonStringEqualsJsonFile()**

assertJsonStringEqualsJsonFile(mixed \$expectedFile, mixed \$actualJson[, string \$message = ])

\$actualJson                  \$expectedFile                  ,                  \$message                  .

4.37 assertJsonStringEqualsJsonFile()

```
<?php
class JsonStringEqualsJsonFileTest extends PHPUnit_Framework_TestCase
{
    public function testFailure()
    {
        $this->assertJsonStringEqualsJsonFile(
            'path/to/fixture/file',
            json_encode(array("Mascott" => "ux"));
    }
}
?>
```

PHPUnit 3.7.0 by Sebastian Bergmann.  
 F



```

Time: 0 seconds, Memory: 5.00Mb
There was 1 failure:
1) JsonStringEqualsJsonFile::testFailure
Failed asserting that '{"Mascott":"ux"}' matches JSON string
 '{"Mascott":"Tux"}'.
/home/sb/JsonStringEqualsJsonFileTest.php:5
FAILURES!
Tests: 1, Assertions: 3, Failures: 1.phpunit
JsonStringEqualsJsonFileTest
PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.00Mb
There was 1 failure:
1) JsonStringEqualsJsonFile::testFailure
Failed asserting that '{"Mascott":"ux"}' matches JSON string
 '{"Mascott":"Tux"}'.
/home/sb/JsonStringEqualsJsonFileTest.php:5
FAILURES!
Tests: 1, Assertions: 3, Failures: 1.

```

### assertJsonStringEqualsJsonString()

```
assertJsonStringEqualsJsonString(mixed $expectedJson, mixed $actualJson[, string $message = ])
```

```

$actualJson        $expectedJson                ,      $message
                .
4.38 assertJsonStringEqualsJsonString()

```

```

<?php
class JsonStringEqualsJsonStringTest extends PHPUnit_Framework_TestCase
{
    public function testFailure()
    {
        $this->assertJsonStringEqualsJsonString(
            json_encode(array("Mascott" => "Tux"),
            json_encode(array("Mascott" => "ux"));
    }
}
?>

```

```

PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.00Mb
There was 1 failure:
1) JsonStringEqualsJsonStringTest::testFailure
Failed asserting that two objects are equal.
--- Expected
+++ Actual
@@ @@
stdClass Object (

```

```

-
'Mascott' => 'Tux'
+
'Mascott' => 'ux'
)
/home/sb/JsonStringEqualsJsonStringTest.php:5
FAILURES!
Tests: 1, Assertions: 3, Failures: 1.phpunit
JsonStringEqualsJsonStringTest
PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.00Mb
There was 1 failure:
1) JsonStringEqualsJsonStringTest::testFailure
Failed asserting that two objects are equal.
--- Expected
+++ Actual
@@ @@
stdClass Object (
-
'Mascott' => 'Tux'
+
'Mascott' => 'ux'
)
/home/sb/JsonStringEqualsJsonStringTest.php:5
FAILURES!
Tests: 1, Assertions: 3, Failures: 1.

```

**assertLessThan()**

assertLessThan(mixed \$expected, mixed \$actual[, string \$message = ])

\$actual                    \$expected                                    ,                    \$message                                    .

assertAttributeLessThan()                    wrapper                    ,                    public, protected, private

4.39 assertLessThan()

```

<?php
class LessThanTest extends PHPUnit_Framework_TestCase
{
    public function testFailure()
    {
        $this->assertLessThan(1, 2);
    }
}
?>

```

PHPUnit 3.7.0 by Sebastian Bergmann.
F

```

Time: 0 seconds, Memory: 5.00Mb
There was 1 failure:
1) LessThanTest::testFailure
Failed asserting that 2 is less than 1.
/home/sb/LessThanTest.php:6
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.phpunit LessThanTest
PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.00Mb
There was 1 failure:
1) LessThanTest::testFailure
Failed asserting that 2 is less than 1.
/home/sb/LessThanTest.php:6
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.
    
```

### assertLessThanOrEqual()

assertLessThanOrEqual(mixed \$expected, mixed \$actual[, string \$message = ])

\$actual            \$expected            가            (            ),            \$message            .  
 assertAttributeLessThanOrEqual()            wrapper            ,            public, protected, private

#### 4.40 assertLessThanOrEqual()

```

<?php
class LessThanOrEqualTest extends PHPUnit_Framework_TestCase
{
    public function testFailure()
    {
        $this->assertLessThanOrEqual(1, 2);
    }
}
?>
    
```

```

PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.25Mb
There was 1 failure:
1) LessThanOrEqualTest::testFailure
Failed asserting that 2 is equal to 1 or is less than 1.
/home/sb/LessThanOrEqualTest.php:6
FAILURES!
Tests: 1, Assertions: 2, Failures: 1.phpunit LessThanOrEqualTest
PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.25Mb
    
```

```

There was 1 failure:
1) LessThanOrEqualTest::testFailure
Failed asserting that 2 is equal to 1 or is less than 1.
/home/sb/LessThanOrEqualTest.php:6
FAILURES!
Tests: 1, Assertions: 2, Failures: 1.
    
```

**assertNull()**

```

assertNull(mixed $variable[, string $message = ])
$variable NULL , $message .
assertNotNull() .
        4.41 assertNull()
    
```

```

<?php
class NullTest extends PHPUnit_Framework_TestCase
{
    public function testFailure()
    {
        $this->assertNull('foo');
    }
}
?>
    
```

```

PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.00Mb
There was 1 failure:
1) NullTest::testFailure
Failed asserting that 'foo' is null.
/home/sb/NotNullTest.php:6
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.phpunit NotNullTest
PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.00Mb
There was 1 failure:
1) NullTest::testFailure
Failed asserting that 'foo' is null.
/home/sb/NotNullTest.php:6
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.
    
```

### assertObjectHasAttribute()

assertObjectHasAttribute(string \$attributeName, object \$object[, string \$message = ])

\$object->attributeName , \$message .

assertObjectNotHasAttribute() , .

4.42 assertObjectHasAttribute()

```
<?php
class ObjectHasAttributeTest extends PHPUnit_Framework_TestCase
{
    public function testFailure()
    {
        $this->assertObjectHasAttribute('foo', new stdClass);
    }
}
?>
```

PHPUnit 3.7.0 by Sebastian Bergmann.

F

Time: 0 seconds, Memory: 4.75Mb

There was 1 failure:

1) ObjectHasAttributeTest::testFailure

Failed asserting that object of class "stdClass" has attribute "foo".

/home/sb/ObjectHasAttributeTest.php:6

FAILURES!

Tests: 1, Assertions: 1, Failures: 1.phpunit ObjectHasAttributeTest

PHPUnit 3.7.0 by Sebastian Bergmann.

F

Time: 0 seconds, Memory: 4.75Mb

There was 1 failure:

1) ObjectHasAttributeTest::testFailure

Failed asserting that object of class "stdClass" has attribute "foo".

/home/sb/ObjectHasAttributeTest.php:6

FAILURES!

Tests: 1, Assertions: 1, Failures: 1.

### assertTag()

assertTag(array \$matcher, string \$actual[, string \$message = , boolean \$isHtml = TRUE])

\$actual \$matcher , \$message .

\$matcher associative array , .

- id: id 가
- tag: 가
- attributes: associative array \$attributes
- content:
- parent: parent 가 associative array \$parent
- child: child 가 associative array \$child

- ancestor: 가 associative array \$ancestor
- descendant: 가 associative array \$descendant
- children: child associative array
- count: child 가
- less\_than: child 가
- greater\_than: child 가
- only: associative array child , child

assertNotTag()

4.54 assertTag()

```
<?php
// Matcher that asserts that there is an element with an id="my_id".
$matcher = array('id' => 'my_id');

// Matcher that asserts that there is a "span" tag.
$matcher = array('tag' => 'span');

// Matcher that asserts that there is a "span" tag with the content
// "Hello World".
$matcher = array('tag' => 'span', 'content' => 'Hello World');

// Matcher that asserts that there is a "span" tag with content
matching the
// regular expression pattern.
$matcher = array('tag' => 'span', 'content' => 'regexp:/Try
P(HP|ython)/');

// Matcher that asserts that there is a "span" with an "list" class
attribute.
$matcher = array(
    'tag' => 'span',
    'attributes' => array('class' => 'list')
);

// Matcher that asserts that there is a "span" inside of a "div".
$matcher = array(
    'tag' => 'span',
    'parent' => array('tag' => 'div')
);

// Matcher that asserts that there is a "span" somewhere inside a
"table".
$matcher = array(
    'tag' => 'span',
    'ancestor' => array('tag' => 'table')
);
```

```
// Matcher that asserts that there is a "span" with at least one "em"
child.
$matcher = array(
    'tag' => 'span',
    'child' => array('tag' => 'em')
);

// Matcher that asserts that there is a "span" containing a (possibly
nested)
// "strong" tag.
$matcher = array(
    'tag' => 'span',
    'descendant' => array('tag' => 'strong')
);

// Matcher that asserts that there is a "span" containing 5-10 "em"
tags as
// immediate children.
$matcher = array(
    'tag' => 'span',
    'children' => array(
        'less_than' => 11,
        'greater_than' => 4, 'only' => array('tag' =>
'em')
    )
);

// Matcher that asserts that there is a "div", with an "ul" ancestor
and a "li"
// parent (with class="enum"), and containing a "span" descendant that
contains
// an element with id="my_test" and the text "Hello World".
$matcher = array(
    'tag' => 'div',
    'ancestor' => array('tag' => 'ul'),
    'parent' => array(
        'tag' => 'li',
        'attributes' => array('class' => 'enum')
    ),
    'descendant' => array(
        'tag' => 'span',
        'child' => array(
            'id' => 'my_test',
            'content' => 'Hello World'
        )
    )
);
```

```
// Use assertTag() to apply a $matcher to a piece of $html.
$this->assertTag($matcher, $html);

// Use assertTag() to apply a $matcher to a piece of $xml.
$this->assertTag($matcher, $xml, '', FALSE);
?>
```

**assertThat()**

PHPUnit\_Framework\_Constraint  
 assertThat() 가 . 4.55 logicalNot() equalTo()  
 (constraint) assertNotEquals()  
 assertThat(mixed \$value, PHPUnit\_Framework\_Constraint \$constraint[, \$message = ])  
 \$value 가 \$constraint , \$message

4.55: Usage of assertThat()

```
<?php
class BiscuitTest extends PHPUnit_Framework_TestCase
{
    public function testEquals()
    {
        $theBiscuit = new Biscuit('Ginger');
        $myBiscuit = new Biscuit('Ginger');
        $this->assertThat(
            $theBiscuit,
            $this->logicalNot(
                $this->equalTo($myBiscuit)
            )
        );
    }
}
?>
```

4.3 가 PHPUnit\_Framework\_Constraint

PHPUnit_Framework_Constraint_Attribute attribute(PHPUnit_Framework_Constraint \$constraint, \$attributeName)	,
PHPUnit_Framework_Constraint_IsAnything anything()	
PHPUnit_Framework_Constraint_ArrayHasKey arrayHasKey(mixed \$key)	
PHPUnit_Framework_Constraint_TraversableContains contains(mixed \$value)	Iterator 가 가, 가
PHPUnit_Framework_Constraint_TraversableContainsOnly containsOnly(string \$type)	가 array, Iterator 가, type 가
PHPUnit_Framework_Constraint_TraversableContainsOnly containsOnlyInstancesOf(string \$classname)	가 array, Iterator 가, 가



PHPUnit_Framework_Constraint_IsEqual equalTo(\$value, \$delta = 0, \$maxDepth = 10)	
PHPUnit_Framework_Constraint_AttributeattributeEqualTo(\$attributeName,\$value, \$delta = 0, \$maxDepth = 10)	
PHPUnit_Framework_Constraint_FileExists fileExists()	
PHPUnit_Framework_Constraint_GreaterThan greaterThan(mixed \$value)	가
PHPUnit_Framework_Constraint_OrgreaterThanOrEqual(mixed \$value)	가
PHPUnit_Framework_Constraint_ClassHasAttribute classHasAttribute(string \$attributeName)	가      가      가
PHPUnit_Framework_Constraint_ClassHasStaticAttributeclassHasStaticAttribute(string \$attributeName)	가      가      static      가
PHPUnit_Framework_Constraint_ObjectHasAttributehasAttribute(string \$attributeName)	가      가      가
PHPUnit_Framework_Constraint_IsIdentical identicalTo(mixed \$value)	
PHPUnit_Framework_Constraint_IsFalse isFalse()	가      FALSE
PHPUnit_Framework_Constraint_IsInstanceOf isInstanceOf(string \$className)	가      가
PHPUnit_Framework_Constraint_IsNull isNull()	가      NULL
PHPUnit_Framework_Constraint_IsTrue isTrue()	가      TRUE
PHPUnit_Framework_Constraint_IsType isType(string \$type)	가      type
PHPUnit_Framework_Constraint_LessThan lessThan(mixed \$value)	가
PHPUnit_Framework_Constraint_OrlessThanOrEqual(mixed \$value)	가
logicalAnd()	AND
logicalNot(PHPUnit_Framework_Constraint \$constraint)	NOT
logicalOr()	OR
logicalXor()	XOR
PHPUnit_Framework_Constraint_PCERMatch matchesRegularExpression(string \$pattern)	가
PHPUnit_Framework_Constraint_StringContains stringContains(string \$string, bool \$case)	가
PHPUnit_Framework_Constraint_StringEndsWith stringEndsWith(string \$suffix)	가
PHPUnit_Framework_Constraint_StringStartsWith stringStartsWith(string \$prefix)	가
4.3	

### assertTrue()

assertTrue(bool \$condition[, string \$message = ])

\$condition FALSE , \$message .

4.56 assertTrue()

```
<?php
class TrueTest extends PHPUnit_Framework_TestCase
{
    public function testFailure()
    {
        $this->assertTrue(FALSE);
    }
}
```

```
PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.00Mb
There was 1 failure:
1) TrueTest::testFailure
Failed asserting that false is true.
/home/sb/TrueTest.php:6
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.phpunit TrueTest
PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.00Mb
There was 1 failure:
1) TrueTest::testFailure
Failed asserting that false is true.
/home/sb/TrueTest.php:6
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.
```

### assertXmlFileEqualsXmlFile()

assertXmlFileEqualsXmlFile(string \$expectedFile, string \$actualFile[, string \$message = ])

\$actualFile XML 가, \$expectedFile XML , \$message

assertXmlFileNotEqualsXmlFile() , .

4.57 assertXmlFileEqualsXmlFile()

```
<?php
class XmlFileEqualsXmlFileTest extends PHPUnit_Framework_TestCase
{
    public function testFailure()
    {
```

```

$this->assertXmlFileEqualsXmlFile(
    '/home/sb/expected.xml',
    '/home/sb/actual.xml');
}
}
?>

```

PHPUnit 3.7.0 by Sebastian Bergmann.

F

Time: 0 seconds, Memory: 5.25Mb

There was 1 failure:

1) XmlFileEqualsXmlFileTest::testFailure

Failed asserting that two DOM documents are equal.

--- Expected

+++ Actual

@@ @@

```
<?xml version="1.0"?>
```

```
<foo>
```

```
- <bar/>
```

```
+ <baz/>
```

```
</foo>
```

/home/sb/XmlFileEqualsXmlFileTest.php:7

FAILURES!

Tests: 1, Assertions: 3, Failures: 1.phpunit XmlFileEqualsXmlFileTest

PHPUnit 3.7.0 by Sebastian Bergmann.

F

Time: 0 seconds, Memory: 5.25Mb

There was 1 failure:

1) XmlFileEqualsXmlFileTest::testFailure

Failed asserting that two DOM documents are equal.

--- Expected

+++ Actual

@@ @@

```
<?xml version="1.0"?>
```

```
<foo>
```

```
- <bar/>
```

```
+ <baz/>
```

```
</foo>
```

/home/sb/XmlFileEqualsXmlFileTest.php:7

FAILURES!

Tests: 1, Assertions: 3, Failures: 1.

### assertXmlStringEqualsXmlFile()

assertXmlStringEqualsXmlFile(string \$expectedFile, string \$actualXml[, string \$message = ])

\$actualXml XML 7| \$expectedFile XML , \$message

assertXmlStringNotEqualsXmlFile()

#### 4.58 assertXmlStringEqualsXmlFile()

```
<?php
class XmlStringEqualsXmlFileTest extends PHPUnit_Framework_TestCase
{
    public function testFailure()
    {
        $this->assertXmlStringEqualsXmlFile(
            '/home/sb/expected.xml', '<foo><baz/></foo>');
    }
}
?>
```

```
PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.25Mb
There was 1 failure:
1) XmlStringEqualsXmlFileTest::testFailure
Failed asserting that two DOM documents are equal.
--- Expected
+++ Actual
@@ @@
<?xml version="1.0"?>
<foo>
- <bar/>
+ <baz/>
</foo>
/home/sb/XmlStringEqualsXmlFileTest.php:7
FAILURES!
Tests: 1, Assertions: 2, Failures: 1.phpunit XmlStringEqualsXmlFileTest
PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.25Mb
There was 1 failure:
1) XmlStringEqualsXmlFileTest::testFailure
Failed asserting that two DOM documents are equal.
--- Expected
+++ Actual
@@ @@
<?xml version="1.0"?>
<foo>
- <bar/>
```

```
+ <baz/>
</foo>
/home/sb/XmlStringEqualsXmlFileTest.php:7
FAILURES!
Tests: 1, Assertions: 2, Failures: 1.
```

### assertXmlStringEqualsXmlString()

assertXmlStringEqualsXmlString(string \$expectedXml, string \$actualXml[, string \$message = ])
\$actualXml XML 가 \$expectedXml XML , \$message

assertXmlStringNotEqualsXmlString()

4.59 assertXmlStringEqualsXmlString()

```
<?php
class XmlStringEqualsXmlStringTest extends PHPUnit_Framework_TestCase
{
    public function testFailure()
    {
        $this->assertXmlStringEqualsXmlString(
            '<foo><bar/></foo>', '<foo><baz/></foo>');
    }
}
?>
```

```
PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.00Mb
There was 1 failure:
1) XmlStringEqualsXmlStringTest::testFailure
Failed asserting that two DOM documents are equal.
--- Expected
+++ Actual
@@ @@
<?xml version="1.0"?>
<foo>
- <bar/>
+ <baz/>
</foo>
/home/sb/XmlStringEqualsXmlStringTest.php:7
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.phpunit
XmlStringEqualsXmlStringTest
PHPUnit 3.7.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.00Mb
There was 1 failure:
1) XmlStringEqualsXmlStringTest::testFailure
```

```
Failed asserting that two DOM documents are equal.
--- Expected
+++ Actual
@@ @@
<?xml version="1.0"?>
<foo>
- <bar/>
+ <baz/>
</foo>
/home/sb/XmlStringEqualsXmlStringTest.php:7
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.
```

## Notes

# PHPUnitManual:4.7

---

4.7

PHPUnit

가

4.60.

```
<?php
class ArrayDiffTest extends PHPUnit_Framework_TestCase
{
    public function testEquality() {
        $this->assertEquals(
            array(1,2,3 ,4,5,6),
            array(1,2,33,4,5,6)
        );
    }
}
```

```
PHPUnit 3.6.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.25Mb
There was 1 failure:
1) ArrayDiffTest::testEquality
Failed asserting that two arrays are equal.
--- Expected
+++ Actual
@@ @@
Array (
0 => 1
1 => 2
```

```

-
2 => 3
+
2 => 33
3 => 4
4 => 5
5 => 6
)
/home/sb/ArrayDiffTest.php:7
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.phpunit ArrayDiffTest
PHPUnit 3.6.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.25Mb
There was 1 failure:
1) ArrayDiffTest::testEquality
Failed asserting that two arrays are equal.
--- Expected
+++ Actual
@@ @@
Array (
0 => 1
1 => 2
-
2 => 3
+
2 => 33
3 => 4
4 => 5
5 => 6
)
/home/sb/ArrayDiffTest.php:7
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.

```

가 가

PHPUnit 가

4.61. 가

```

<?php
class LongArrayDiffTest extends PHPUnit_Framework_TestCase
{
    public function testEquality() {
        $this->assertEquals(
            array(0,0,0,0,0,0,0,0,0,0,0,0,1,2,3 ,4,5,6),
            array(0,0,0,0,0,0,0,0,0,0,0,0,1,2,33,4,5,6)
        );
    }
}

```

```

        );
    }
}
?>

```

```

PHPUnit 3.6.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.25Mb
There was 1 failure:
1) LongArrayDiffTest::testEquality
Failed asserting that two arrays are equal.
--- Expected
+++ Actual
@@ @@
13 => 2
-
14 => 3
+
14 => 33
15 => 4
16 => 5
17 => 6
)
/home/sb/LongArrayDiffTest.php:7
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.phpunit LongArrayDiffTest
PHPUnit 3.6.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.25Mb
There was 1 failure:
1) LongArrayDiffTest::testEquality
Failed asserting that two arrays are equal.
--- Expected
+++ Actual
@@ @@
13 => 2
-
14 => 3
+
14 => 33
15 => 4
16 => 5
17 => 6
)
/home/sb/LongArrayDiffTest.php:7
FAILURES!
Tests: 1, Assertions: 1, Failures: 1.

```



Notes

# PHPUnitManual:4.8

---

4.8 (Edge case)  
 가 PHPUnit  
 assertEquals (weak)  
 4.62. (weak) (generation) (edge case)

```
<?php
class ArrayWeakComparisonTest extends PHPUnit_Framework_TestCase
{
    public function testEquality() {
        $this->assertEquals(
            array(1, 2, 3, 4, 5, 6),
            array('1', 2, 33, 4, 5, 6)
        );
    }
}
?>
```

```
PHPUnit 3.6.0 by Sebastian Bergmann.
F
Time: 0 seconds, Memory: 5.25Mb
There was 1 failure:
1) ArrayWeakComparisonTest::testEquality
Failed asserting that two arrays are equal.
--- Expected
+++ Actual
@@ @@
Array (
-
0 => 1
+
0 => '1'
1 => 2
-
2 => 3
+
2 => 33
3 => 4
4 => 5
5 => 6
)
/home/sb/ArrayWeakComparisonTest.php:7
```

```

FAILURES!
Tests: 1, Assertions: 1, Failures: 1.

```

```

    (match)      , 1 '1'      가      assertEquals가
                .[1]

```

### Notes

[1] 1 '1' .

## PHPUnitManual:5

### 5 The Command-Line Test Runner

phpunit PHPUnit command-line test runner . PHPUnit  
 command-line test runner .

```

phpunit ArrayTest
PHPUnit 3.7.0 by Sebastian Bergmann.

..

Time: 0 seconds

OK (2 tests, 2 assertions)

```

, PHPUnit command-line tool .

```

.
가 .
F
method .
E
method 가 .
S
가 skip ( 9 , ).
I
가 " " " " ( 9 , ).
PHPUnit (failures) (error) . PHPUnitit ,
assertEquals() , PHP 가 .
가 . 가 ,
, .

```

## Notes

# PHPUnitManual:5.1

---

### 5.1 Command-Line switches

, PHPUnit command-line tool .

```
phpunit --help
```

PHPUnit 3.7.0 by Sebastian Bergmann.

Usage: phpunit [switches] UnitTest [UnitTest.php]

phpunit [switches] <directory>

```

--log-junit <file>           Log test execution in JUnit XML format to file.
--log-tap <file>            Log test execution in TAP format to file.
--log-json <file>           Log test execution in JSON format.

--coverage-clover <file>    Generate code coverage report in Clover XML
format.
--coverage-html <dir>       Generate code coverage report in HTML format.
--coverage-php <file>       Serialize PHP_CodeCoverage object to file.
--coverage-text=<file>      Generate code coverage report in text format.
                             Default to writing to the standard output.

--testdox-html <file>       Write agile documentation in HTML format to file.
--testdox-text <file>      Write agile documentation in Text format to file.

--filter <pattern>         Filter which tests to run.
--group ...                Only runs tests from the specified
group(s) .
--exclude-group ...        Exclude tests from the specified group(s).
--list-groups              List available test groups.

--loader <loader>          TestSuiteLoader implementation to use.
--printer <printer>        TestSuiteListener implementation to use.
--repeat <times>          Runs the test(s) repeatedly.

--tap                      Report test execution progress in TAP
format.
--testdox                  Report test execution progress in TestDox
format.

--colors                   Use colors in output.
--stderr                   Write to STDERR instead of STDOUT.
--stop-on-error            Stop execution upon first error.
--stop-on-failure          Stop execution upon first error or failure.

```

```

--stop-on-skipped      Stop execution upon first skipped test.
--stop-on-incomplete  Stop execution upon first incomplete test.
--strict              Run tests in strict mode.
-v|--verbose          Output more verbose information.
--debug               Display debugging information during test
execution.

--process-isolation   Run each test in a separate PHP process.
--no-globals-backup   Do not backup and restore $GLOBALS for each
test.
--static-backup       Backup and restore static attributes for
each test.

--bootstrap <file>   A "bootstrap" PHP file that is run before the
tests.
-c|--configuration <file> Read configuration from XML file.
--no-configuration    Ignore default configuration file
(PHPUnit.xml).
--include-path <path(s)> Prepend PHP's include_path with given path(s).
-d key[=value]        Sets a php.ini value.

-h|--help             Prints this usage information.
--version             Prints the version and exits.
    
```

Option	
phpunit UnitTest	UnitTest UnitTest PHPUnit_Framework_TestCase return ( :PHPUnit_Framework_TestSuite)
phpunit UnitTest UnitTest.php	UnitTest
--log-junit	JUnit XML . "18 "
--log-tap	Test Anything Protocol (TAP) [1] . "18 "
--log-json	JSON [2] . "18 "
--coverage-html	code coverage HTML . "14 code coverage " , tokenizer Xdebug , 가 .
--coverage-clover	XML , code coverage . "18 " , tokenizer Xdebug , 가 .
--coverage-php	(serialized) PHP_CodeCoverage , code coverage . , tokenizer Xdebug , 가 .
--coverage-text	, command-line code coverage . "18 " , tokenizer Xdebug , 가 .

<b>--testdox-html</b> <b>and --testdox-text</b>	HTML text . "15 "
<b>--filter</b>	가 [3]
<b>--group</b>	@author @group @group
<b>--exclude-group</b>	@group
<b>--list-groups</b>	가
<b>--loader</b>	PHPUnit_Runner_TestSuiteLoader test suite loader PHP include_path . PEAR Project_Package_Class Project/Package/Class.php
<b>--printer</b>	PHPUnit_Util_Printer PHPUnit_Framework_TestListener
<b>--repeat</b>	
<b>--tap</b>	Test Anything Protocol (TAP) [1] . "18 "
<b>--testdox</b>	agile . "15 "
<b>--colors</b>	
<b>--stderr</b>	STDOUT 가 STDERR
<b>--stop-on-error</b>	
<b>--stop-on-failure</b>	
<b>--stop-on-skipped</b>	skip
<b>--stop-on-incomplete</b>	
<b>--strict</b>	strict
<b>--verbose</b>	skip
<b>--process-isolation</b>	PHP
<b>--no-globals-backup</b>	\$GLOBALS . "Global State"
<b>--static-backup</b>	. "Global State"
<b>--bootstrap</b>	"bootstrap" PHP
<b>--configuration,</b> <b>-c</b>	XML . " C XML " phpunit.xml phpunit.xml.dist ( ) 가 , --configuration 가
<b>--no-configuration</b>	phpunit.xml phpunit.xml.dist
<b>--include-path</b>	PHP include_path 가
<b>-d</b>	PHP
<b>--debug</b>	
<b>phpunit</b>	

Notes

# PHPUnitManual:6

---

6 . Fixtures

```

class StackTest extends PHPUnit_Framework_TestCase
{
    protected $stack;

    protected function setUp()
    {
        $this->stack = array();
    }

    public function testEmpty()
    {
        $this->assertTrue(empty($this->stack));
    }

    public function testPush()
    {
        array_push($this->stack, 'foo');
        $this->assertEquals('foo', $this->stack[count($this->stack)-1]);
        $this->assertFalse(empty($this->stack));
    }
}
    
```

6.1: setUp() stack fixture

```

<?php
class StackTest extends PHPUnit_Framework_TestCase
{
    protected $stack;

    protected function setUp()
    {
        $this->stack = array();
    }

    public function testEmpty()
    {
        $this->assertTrue(empty($this->stack));
    }

    public function testPush()
    {
        array_push($this->stack, 'foo');
        $this->assertEquals('foo', $this->stack[count($this->stack)-1]);
        $this->assertFalse(empty($this->stack));
    }
}
    
```

```

    }

    public function testPop()
    {
        array_push($this->stack, 'foo');
        $this->assertEquals('foo', array_pop($this->stack));
        $this->assertTrue(empty($this->stack));
    }
}
?>

```

setUp() tearDown() , ( ) .  
 , setUpBeforeClass() tearDownAfterClass() ,  
 가 .

6.2: 가

```

<?php
class TemplateMethodsTest extends PHPUnit_Framework_TestCase
{
    public static function setUpBeforeClass()
    {
        fwrite(STDOUT, __METHOD__ . "\n");
    }

    protected function setUp()
    {
        fwrite(STDOUT, __METHOD__ . "\n");
    }

    protected function assertPreConditions()
    {
        fwrite(STDOUT, __METHOD__ . "\n");
    }

    public function testOne()
    {
        fwrite(STDOUT, __METHOD__ . "\n");
        $this->assertTrue(TRUE);
    }

    public function testTwo()
    {
        fwrite(STDOUT, __METHOD__ . "\n");
        $this->assertTrue(FALSE);
    }
}

```

```
protected function assertPostConditions ()
{
    fwrite(STDOUT, __METHOD__ . "\n");
}

protected function tearDown()
{
    fwrite(STDOUT, __METHOD__ . "\n");
}

public static function tearDownAfterClass ()
{
    fwrite(STDOUT, __METHOD__ . "\n");
}

protected function onNotSuccessfulTest (Exception $e)
{
    fwrite(STDOUT, __METHOD__ . "\n");
    throw $e;
}
}
?>
```

```
phpunit TemplateMethodsTest
PHPUnit 3.7.0 by Sebastian Bergmann.

TemplateMethodsTest::setUpBeforeClass
TemplateMethodsTest::setUp
TemplateMethodsTest::assertPreConditions
TemplateMethodsTest::testOne
TemplateMethodsTest::assertPostConditions
TemplateMethodsTest::tearDown
.TemplateMethodsTest::setUp
TemplateMethodsTest::assertPreConditions
TemplateMethodsTest::testTwo
TemplateMethodsTest::tearDown
TemplateMethodsTest::onNotSuccessfulTest
FTemplateMethodsTest::tearDownAfterClass

Time: 0 seconds, Memory: 5.25Mb

There was 1 failure:

1) TemplateMethodsTest::testTwo
Failed asserting that <boolean:false> is true.
/home/sb/TemplateMethodsTest.php:30
```



```
FAILURES!
Tests: 2, Assertions: 2, Failures: 1.
```

**Notes**

# PHPUnitManual:6.1

---

6.1 tearDown() setUp()  
 setUp() tearDown() , tearDown() 가 setUp()  
 ( ) . setUp() PHP  
 , tearDown() , setUp()  
 , tearDown() unset()  
 gabage collection 가 .

**Notes**

# PHPUnitManual:6.2

---

6.2 Variations  
 가 , 가 가  
 .  
 setUp() 가 , setUp()  
 setUp() 가 , 가 .

**Notes**

---

# PHPUnitManual:6.3

---

## 6.3 Fixture

가 fixture  
 fixture  
 가 fixture  
 가  
 " **6.3: test suite** **fixutre** " , setUpBeforeClass()  
 tearDownAfterClass()  
 , 가  
 6.3: test suite **fixutre**

```
<?php
class DatabaseTest extends PHPUnit_Framework_TestCase
{
    protected static $dbh;

    public static function setUpBeforeClass()
    {
        self::$dbh = new PDO('sqlite::memory:');
    }

    public static function tearDownAfterClass()
    {
        self::$dbh = NULL;
    }
}
?>
```

fixture 가  
 , stub (10 Test Doubles )

### Notes

# PHPUnitManual:6.4

## 6.4 Global State

singleton

[1]. 가 ,  
가

PHP

- \$foo = 'bar'; \$GLOBALS['foo'] = 'bar';
  - \$GLOBALS
  - (scope) 가 built-in
  - \$foo \$GLOBALS['foo']
- , global \$foo;
- global state 가
- (default) , PHPUnit
- (\$GLOBALS<, \$\_ENV, \$\_POST, \$\_GET, \$\_COOKIE, \$\_SERVER, \$\_FILES, \$\_REQUEST)

/ PHP 5.3  
/ serialize() unserialize()

PHP serialize ( :  
PDO ). 가 \$GLOBALS ,  
"@backupGlobals" @backupGlobals , /

```
class MyTest extends PHPUnit_Framework_TestCase
{
    protected $backupGlobalsBlacklist = array('globalVariable');

    // ...
}
```

\$backupGlobalsBlacklist setUp() 가  
@backupStaticAttributes \$backupGlobalsBlacklist , /

```
class MyTest extends PHPUnit_Framework_TestCase
{
    protected $backupStaticAttributesBlacklist = array(
        'className' => array('attributeName')
    );
}
```

```
// ...
}
```

"\$backupStaticAttributesBlacklist" setUp() 가 .

### Notes

## PHPUnitManual:7

---

7

PHPUnit (" 2 PHPUnit " ) , " " " " " ,

PHPUnit test suite .

### Notes

## PHPUnitManual:7.1

---

7.1 test suite

test suite 가 ,  
 . PHPUnit

Object\_Freezer test suite . ,  
 가 "Test" . ,  
 (SUT) 가 "Object" .

Object	Tests
-- Freezer	-- Freezer
-- HashGenerator	-- HashGenerator
`-- NonRecursiveSHA1.php	`--
NonRecursiveSHA1Test.php	
-- HashGenerator.php	
-- IdGenerator	-- IdGenerator
`-- UUID.php	`-- UUIDTest.php
-- IdGenerator.php	
-- LazyProxy.php	
-- Storage	-- Storage
`-- CouchDB.php	`-- CouchDB
	--
WithLazyLoadTest.php	

```

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
WithoutLazyLoadTest.php |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|-- Freezer.php         |-- FreezerTest.php

```

PHPUnit command-line test runner 가

```

phpunit Tests
PHPUnit 3.7.0 by Sebastian Bergmann.

..... 60 / 75
.....

Time: 0 seconds

OK (75 tests, 164 assertions)

```

PHPUnit comman-line test runner , \*Test.php

Tests/FreezerTest.php Object\_FreezerTest

```

phpunit Tests/FreezerTest
PHPUnit 3.7.0 by Sebastian Bergmann.

.....

Time: 0 seconds

OK (28 tests, 60 assertions)

```

, --filter .

```

phpunit --filter testFreezingAnObjectWorks Tests
PHPUnit 3.7.0 by Sebastian Bergmann.

.

Time: 0 seconds

OK (1 test, 2 assertions)

```

가 ,  
 XML . " " . ,

### Notes

## PHPUnitManual:7.2

7.2 XML test suite  
 PHPUnit XML ( C XML ) test suite . " 7.1  
 XML test suite " . 7.1 Tests  
 \*Test.php \*Test 가 .  
 7.1: XML test suite

```
<phpunit>
  <testsuites>
    <testsuite name="Object_Freezer">
      <directory>Tests</directory>
    </testsuite>
  </testsuites>
</phpunit>
```

7.2: XML test suite

```
<phpunit>
  <testsuites>
    <testsuite name="Object_Freezer">
      <file>Tests/Freezer/HashGenerator/NonRecursiveSHA1Test.php</file>
      <file>Tests/Freezer/IdGenerator/UUIDTest.php</file>
      <file>Tests/Freezer/UtilTest.php</file>
      <file>Tests/FreezerTest.php</file>
      <file>Tests/Freezer/StorageTest.php</file>
      <file>Tests/Freezer/Storage/CouchDB/WithLazyLoadTest.php</file>
      <file>Tests/Freezer/Storage/CouchDB/WithoutLazyLoadTest.php</file>
    </testsuite>
  </testsuites>
</phpunit>
```

### Notes

# PHPUnitManual:8

---

8 .  
 , Wordpress TYP03, Symfony Doctrine Propel PHPUnit  
 가 가  
 . PHPUnit  
 , 가 JOIN  
 SELECT, INSERT, UPDATE, DELETE 가  
 ( )

DbUnit , ,

```
localhost# pear install phpunit/DbUnit
```

## Notes

# PHPUnitManual:8.1

---

8.1  
 DbUnit 가 , MySQL, PostgreSQL, Oracle, SQLite . Zend  
 Framework[1] Doctrine 2[2] IBM DB2 Microsoft SQL Server

## Notes

# PHPUnitManual:8.2

---

8.2

, ?  
 .  
 ,  
 • /  
 •  
 •  
 •  
 PDO MySQLi, OCIS API ,  
 .  
 가 .  
 • ,  
 • 가 .  
 , test suite  
 2 ,  
 . 가 ,  
 " 가 " ,  
 " " 가 .  
 DbUnit ,  
 PHPUnit ,  
 .  
 가 ,  
 , test suite 1 .  
 Doctrine 2 project[1] . test suite 1000 가  
 .  
 MySQL test suite 15 .

## Notes



# PHPUnitManual:8.3

---

8.3

4

## 4

Gerard Meszaros xUnit Test Patterns

4

- fixture (Setup)
- (Exercise)
- (Verify)
- (Teardown)

### Fixture ?

fixture ,

clean-up insert , setup teardown fixture  
 가 . work flow , Database Extension 4

### 1. Clean-Up Database

. PHPUnit TRUNCATE 가

### 2. Set up fixture

PHPUnit fixture insert .

### 3–5. Run Test, Verify outcome and Teardown

insert , PHPUnit .  
 Database Extension 가 ,  
 assertEquals()  
 "Database Assertions"

### Notes

# PHPUnitManual:8.4

---

## 8.4 PHPUnit

### PHPUnit

, PHPUnit

PHPUnit\_Framework\_TestCase

```
require_once "PHPUnit/Framework/TestCase.php";

class MyTest extends PHPUnit_Framework_TestCase
{
    public function testCalculate()
    {
        $this->assertEquals(2, 1 + 1);
    }
}
```

### Database Extension

, 2

getConnection() getDataSet()

```
require_once "PHPUnit/Extensions/Database/TestCase.php";

class MyGuestbookTest extends PHPUnit_Extensions_Database_TestCase
{
    /**
     * @return PHPUnit_Extensions_Database_DB_IDatabaseConnection
     */
    public function getConnection()
    {
        $pdo = new PDO('sqlite::memory:');
        return $this->createDefaultDBConnection($pdo, ':memory:');
    }

    /**
     * @return PHPUnit_Extensions_Database_DataSet_IDataSet
     */
    public function getDataSet()
    {
        return
$this->createFlatXMLDataSet(dirname(__FILE__) . '/_files/guestbook-seed.xml');
    }
}
```

### getConnection()

clean-up fixture , PHPUnit Database Extension  
 PDO  
 PHPUnit Database Extension PDO  
 clean-up fixture  
 , in-memory Sqlite connection createDefaultDBConnection  
 PDO wrap ,2 ( )  
 type PHPUnit\_Extensions\_Database\_DB\_IDatabaseCon  
 nection  
 " " API

### getDataSet()

getDataSet() , DataSet DataTable ,  
 PHPUnit\_Extensions\_Database\_DataSet\_IDataSet PHPUnit\_Extensions\_Database\_DataSet\_IDataTable  
 , getDataSet() setUp()  
 fixture dataset insert ,  
 createFlatXMLDataSet(\$filename) XML dataset

### Database Schema (DDL) ?

PHPUnit , ( , , , )  
 가 가 , test suite  
 SQLite 가 , phpMyAdmin for MySQL  
 Doctrine 2 Propel , API  
 . PHPUnit Bootstrap

### Tip:

, getConnection()

fixture

```
require_once "PHPUnit/Extensions/Database/TestCase.php";

abstract class MyApp_Tests_DatabaseTestCase extends
PHPUnit_Extensions_Database_TestCase
{
    // PDO , clean-up fixture
    static private $pdo = null;

    // PHPUnit_Extensions_Database_DB_IDatabaseConnection
```

```

private $conn = null;

final public function getConnection()
{
    if ($this->conn === null) {
        if (self::$pdo == null) {
            self::$pdo = new PDO('sqlite::memory:');
        }
        $this->conn = $this->createDefaultDBConnection(self::$pdo,
':memory:');
    }

    return $this->conn;
}
}

```

PDO

. PHPUnit

. XML

. , "phpunit.xml" , tests/

```

<?xml version="1.0" encoding="UTF-8" ?>
<phpunit>
  <php>
    <var name="DB_DSN" value="mysql:dbname=myguestbook;host=localhost" />
    <var name="DB_USER" value="user" />
    <var name="DB_PASSWD" value="passwd" />
    <var name="DB_DBNAME" value="myguestbook" />
  </php>
</phpunit>

```

```

abstract class Generic_Tests_DatabaseTestCase extends
PHPUnit_Extensions_Database_TestCase
{
    // PDO , clean-up fixture
    static private $pdo = null;

    // PHPUnit_Extensions_Database_DB_IDatabaseConnection

    private $conn = null;

    final public function getConnection()
    {
        if ($this->conn === null) {
            if (self::$pdo == null) {
                self::$pdo = new PDO( $GLOBALS['DB_DSN'],

```

```

$GLOBALS['DB_USER'], $GLOBALS['DB_PASSWD'] );
    }
    $this->conn = $this->createDefaultDBConnection(self::$pdo,
$GLOBALS['DB_DBNAME']);
    }

    return $this->conn;
}
}

```

test suite

```

user@desktop> phpunit --configuration developer-a.xml MyTests/
user@desktop> phpunit --configuration developer-b.xml MyTests/

```

race condition ( ) 가

### Notes

## PHPUnitManual:8.5

---

### 8.5 Dataset

#### Dataset

Dataset PHPUnit Database Extension . Dataset  
 PHPUnit row, column . API XML YAML, CSV  
 , PHP . Dataset  
 • , 3 work flow .  
 • , ( dataset).  
 • (YAML, XML, ...).  
 • .  
 PHPUnit Database Extension Dataset , .  
 , Dataset .  
 . fixture dataset .  
 • Dataset row .  
 • row insert .

- 3 dataset /
- dataset /
- Query dataset /
- / dataset /
- dataset / fixture

**Flat XML DataSet**

가 dataset Flat XML xml root node <dataset>  
 tag 가 row tag 가 row , tag  
 column

```
<?xml version="1.0" ?>
<dataset>
  <guestbook id="1" content="Hello buddy!" user="joe" created="2010-04-24 17:15:23" />
  <guestbook id="2" content="I like it!" user="nancy" created="2010-04-26 12:14:20" />
</dataset>
```

<guestbook> 가 2 row 가 가  
 . "id", "content", "user" and "created"

, tag 가 guestbook flat XML

```
<?xml version="1.0" ?>
<dataset>
  <guestbook />
</dataset>
```

Flat XML NULL flat XML NULL

(Oracle). flat XML  
 . NULL row  
 가 , user column NULL 가

guestbook

```
<?xml version="1.0" ?>
<dataset>
  <guestbook id="1" content="Hello buddy!" user="joe" created="2010-04-24 17:15:23" />
  <guestbook id="2" content="I like it!" created="2010-04-26 12:14:20" />
</dataset>
```

, 2 가 column  
 . Dataset , dataset column  
 row NULL , Database Extension column

Flat XML dataset

column , guestbook column "id", "

```

content", "user"          "created"          . 2 row "user"
                        NULL insert
guestbook                dataset              , guestbook column "id", "content"
"created"                "user" 가
Flat XML dataset        , row NULL
                        . 2 row NULL
                        row 가
                        , column flat XML dataset column
(default)                , column "NOT NULL DEFAULT NULL" 가
                        , flat XML dataset , NULL
    
```

Flat XML dataset  
 createFlatXmlDataSet(\$filename)

```

class MyTestCase extends PHPUnit_Extensions_Database_TestCase
{
    public function getDataSet()
    {
        return $this->createFlatXmlDataSet('myFlatXmlFixture.xml');
    }
}
    
```

**XML DataSet**

more structured XML dataset 가 flat XML dataset  
 NULL . root node <dataset> 가 tag ,  
 , <column>, <row>, <value> <null /> . guestbook flat XML dataset  
 dataset .

```

<?xml version="1.0" ?>
<dataset>
  <table name="guestbook">
    <column>id</column>
    <column>content</column>
    <column>user</column>
    <column>created</column>
    <row>
      <value>1</value>
      <value>Hello buddy!</value>
      <value>joe</value>
      <value>2010-04-24 17:15:23</value>
    </row>
    <row>
      <value>2</value>
      <value>I like it!</value>
      <null />
      <value>2010-04-26 12:14:20</value>
    </row>
  </table>
</dataset>
    
```

```
</table>
</dataset>
```

# PHPUnitManual:8.6

## 8.6 Connection API

Connection ,3 가 .  
 getConnection() .

```
interface PHPUnit_Extensions_Database_DB_IDatabaseConnection
{
    public function createDataSet(Array $tableNames = NULL);
    public function createQueryTable($resultName, $sql);
    public function getRowCount($tableName, $whereClause = NULL);

    // ...
}
```

1. createDataSet() , Database (DB) dataset . "dataset "

```
class ConnectionTest extends PHPUnit_Extensions_Database_TestCase
{
    public function testCreateDataSet()
    {
        $tableNames = array('guestbook');
        $dataset = $this->getConnection()->createDataSet();
    }
}
```

2. createQueryTable() QueryTable .  
 SQL query ( API)

```
class ConnectionTest extends PHPUnit_Extensions_Database_TestCase
{
    public function testCreateQueryTable()
    {
        $tableNames = array('guestbook');
        $queryTable =
        $this->getConnection()->createQueryTable('guestbook', 'SELECT * FROM
        guestbook');
    }
}
```

3. getRowCount() row . where  
 가 . 가 .



```
class ConnectionTest extends PHPUnit_Extensions_Database_TestCase
{
    public function testGetRowCount()
    {
        $this->assertEquals(2,
$this->getConnection()->getRowCount('guestbook'));
    }
}
```

**Notes**

# PHPUnitManual:8.7

---

8.7 API

## API

, Database Extension 가 .  
 , row . , .  
**row**  
 row 가 가 Connection API  
 . guestbook row 가 , 2  
 1 row 가 가 3 row 가 .

```
class GuestbookTest extends PHPUnit_Extensions_Database_TestCase
{
    public function testAddEntry()
    {
        $this->assertEquals(2,
$this->getConnection()->getRowCount('guestbook'), "Pre-Condition");

        $guestbook = new Guestbook();
        $guestbook->addEntry("suzy", "Hello world!");

        $this->assertEquals(3,
$this->getConnection()->getRowCount('guestbook'), "Inserting failed");
    }
}
```

### Asserting the State of a Table

가 , column  
 QueryTable , SQL query ,  
 , dataset .

```
class GuestbookTest extends PHPUnit_Extensions_Database_TestCase
{
    public function testAddEntry()
    {
        $guestbook = new Guestbook();
        $guestbook->addEntry("suzy", "Hello world!");

        $queryTable = $this->getConnection()->createQueryTable(
            'guestbook', 'SELECT * FROM guestbook'
        );
        $expectedTable = $this->createFlatXmlDataSet("expectedBook.xml")
            ->getTable("guestbook");
        $this->assertTablesEqual($expectedTable, $queryTable);
    }
}
```

flat XML expectedBook.xml .

```
<?xml version="1.0" ?>
<dataset>
  <guestbook id="1" content="Hello buddy!" user="joe" created="2010-04-24 17:15:23" />
  <guestbook id="2" content="I like it!" user="nancy" created="2010-04-26 12:14:20" />
  <guestbook id="3" content="Hello world!" user="suzy" created="2010-05-01 21:47:08" />
</dataset>
```

2010-05-01 21:47:08  
 가 가 , "created" column

flat XML expectedBook.xml , 가

```
<?xml version="1.0" ?>
<dataset>
  <guestbook id="1" content="Hello buddy!" user="joe" />
  <guestbook id="2" content="I like it!" user="nancy" />
  <guestbook id="3" content="Hello world!" user="suzy" />
</dataset>
```

```
QueryTable .
$queryTable = $this->getConnection()->createQueryTable(
    'guestbook', 'SELECT id, content, user FROM guestbook'
);
```

## Query

query  
 , dataset  
 QueryTable  
 .  
 query

```
class ComplexQueryTest extends PHPUnit_Extensions_Database_TestCase
{
    public function testComplexQuery()
    {
        $queryTable = $this->getConnection()->createQueryTable(
            'myComplexQuery', 'SELECT complexQuery...'
        );
        $expectedTable =
$this->createFlatXmlDataSet("complexQueryAssertion.xml")
            ->getTable("myComplexQuery");
        $this->assertTablesEqual($expectedTable, $queryTable);
    }
}
```

,  
 . dataset 2가  
 Database (DB) dataset , dataset  
 , query dataset dataset

```
class DataSetAssertionsTest extends
PHPUnit_Extensions_Database_TestCase
{
    public function testCreateDataSetAssertion()
    {
        $dataSet =
$this->getConnection()->createDataSet(array('guestbook'));
        $expectedDataSet = $this->createFlatXmlDataSet('guestbook.xml');
        $this->assertDataSetsEqual($expectedDataSet, $dataSet);
    }
}
```

dataset .

```
class DataSetAssertionsTest extends
PHPUnit_Extensions_Database_TestCase
{
    public function testManualDataSetAssertion()
    {
        $dataSet = new
PHPUnit_Extensions_Database_DataSet_QueryDataSet();
        $dataSet->addTable('guestbook', 'SELECT id, content, user FROM
guestbook'); // additional tables
        $expectedDataSet = $this->createFlatXmlDataSet('guestbook.xml');

        $this->assertDataSetsEqual($expectedDataSet, $dataSet);
    }
}
```

```

    }
}

```

### Notes

# PHPUnitManual:8.8

## 8.8 Frequently Asked Questions

### PHPUnit

PHPUnit, test suite 가 test suite  
 Doctrine 2 eZ Components 가 PHPUnit extension  
 (hooked). test suite

### PDO

?  
 , PDO 가 fixture

### Database Extension

### Too much Connections 가

getConnection() PDO connection MySQL  
 connection 100 , PDO  
 " " PDO

### Flat XML

### CSV dataset

### NULL

?

. NULL XML YAML dataset

### Notes

# PHPUnitManual:9

---

- 9 (skip)
- 9 (skip)

## Notes

# PHPUnitManual:9.1

---

9.1

```
public function testSomething()
{
}
```

, PHPUnit framework " " 가 .  
 가 . 가  
 가  
 \$this->fail() , " "  
 가 " " 가 .( : )  
 , 가  
 가 PHPUnit\_Framework\_IncompleteTest  
 가  
 PHPUnit\_Framework\_IncompleteTestError .  
 9.1 " 가" SampleTest .  
 markTestIncomplete() (  
 PHPUnit\_Framework\_IncompleteTestError ) 가

9.1: 가

```
<?php
class SampleTest extends PHPUnit_Framework_TestCase
{
    public function testSomething()
    {
        // Optional: , 가
        $this->assertTrue(TRUE, 'This should already work.');
```

```
        // , 가
        $this->markTestIncomplete(
            'This test has not been implemented yet.'
        );
    }
}
```

```

    }
}
?>

```

**PHPUnit command-line test runner**

I( i)

```

phpunit --verbose SampleTest
PHPUnit 3.7.0 by Sebastian Bergmann.

```

```

I

Time: 0 seconds, Memory: 3.75Mb

There was 1 incomplete test:

1) SampleTest::testSomething
This test has not been implemented yet.

/home/sb/SampleTest.php:12
OK, but incomplete or skipped tests!
Tests: 1, Assertions: 1, Incomplete: 1.

```

9.1 "

API"

API

<b>void markTestIncomplete()</b>	가
<b>void markTestIncomplete(string \$message)</b>	가 \$message 가.
9.1	API

**Notes**

# PHPUnitManual:9.2

9.2

가 가 . , . MySQL  
 , MySQL  
 9.2 " " DatabaseTest testConnection() 가  
 . setUp() , MySQLi 가 ,  
 markTestSkipped()

9.2

```
<?php
class DatabaseTest extends PHPUnit_Framework_TestCase
{
    protected function setUp()
    {
        if (!extension_loaded('mysqli')) {
            $this->markTestSkipped(
                'MySQLi . '
            );
        }
    }

    public function testConnection()
    {
        // ...
    }
}
?>
```

, PHPUnit command-line test runner S .

```
phpunit --verbose DatabaseTest
PHPUnit 3.7.0 by Sebastian Bergmann.

S

Time: 0 seconds, Memory: 3.75Mb

There was 1 skipped test:

1) DatabaseTest::testConnection
The MySQLi extension is not available.

/home/sb/DatabaseTest.php:9
OK, but incomplete or skipped tests!
Tests: 1, Assertions: 0, Skipped: 1.
```

9.2 " API" API .

void markTestSkipped()		
void markTestSkipped(string \$message)	\$message	가.
9.2	API	

## Notes

# PHPUnitManual:9.3

## 9.3 @requires

, @requires

### 9.3 @requires

Type	Possible Values	Examples	Another example
PHP	PHP (Any PHP version identifier)	@requires PHP 5.3.3	@requires PHP 5.4-dev
PHPUnit	PHPUnit (Any PHPUnit version identifier)	@requires PHPUnit 3.6.3	@requires PHPUnit 3.7
function	function_exists (Valid parameter)	@requires function imap_open	@requires function ReflectionMethod::setAccessible
extension	(Any extension name)	@requires extension mysqli	@requires extension curl

9.3 @requires

### 9.3 @requires

```

<?php
/**
 * @requires extension mysqli
 */
class DatabaseTest extends PHPUnit_Framework_TestCase
{
    /**
     * @requires PHP 5.3
     */
    public function testConnection()
    {
        //          mysqli          PHP >= 5.3
    }

    // ...          mysqli
}
?>

```

PHP , "test suite" XML  
include .



Notes

# PHPUnitManual:10

---

10 (Test Doubles)

Gerard Meszaros [Meszaros2007]

Sometimes it is just plain hard to test the system under test (SUT) because it depends on other components that cannot be used in the test environment. This could be because they aren't available, they will not return the results needed for the test or because executing them would have undesirable side effects. In other cases, our test strategy requires us to have more control or visibility of the internal behavior of the SUT.

(SUT:system under test)

When we are writing a test in which we cannot (or chose not to) use a real depended-on component (DOC), we can replace it with a Test Double. The Test Double doesn't have to behave exactly like the real DOC; it merely has to provide the same API as the real one so that the SUT thinks it is the real one!

(DOC:depended-on component)

```

--Gerard Meszaros
PHPUnit getMock($className)
{
    return new Mock($className);
}

class Mock
{
    public function __call($method, $arguments)
    {
        return $this->returnNull();
    }

    public function returnNull()
    {
        return null;
    }
}

final, private, static stub mock PHPUnit
stub mock

10.14 "
가 "
10.15 "
가 mock "
```

Notes

---

# PHPUnitManual:10.1

## 10.1 Stubs

PHPUnit\Framework\TestCase::getMock() method. This method returns a stub of the class or interface specified in the \$className argument. The stub is a PHPUnit\Framework\MockObject\MockObject object that implements the class or interface. The stub is configured with the following options:

- `$className`: The name of the class or interface to mock.
- `$mockName`: The name of the mock object.
- `$mockOriginalClassName`: The name of the original class or interface.
- `$mockOriginalClass`: The original class or interface object.
- `$mockOriginalInterface`: The original interface object.
- `$mockOriginalParentClass`: The original parent class object.
- `$mockOriginalParentInterface`: The original parent interface object.
- `$mockOriginalParentClassAndInterface`: The original parent class and interface object.
- `$mockOriginalParentClassAndInterfaceAndParentClass`: The original parent class and interface and parent class object.
- `$mockOriginalParentClassAndInterfaceAndParentClassAndParentInterface`: The original parent class and interface and parent class and parent interface object.
- `$mockOriginalParentClassAndInterfaceAndParentClassAndParentInterfaceAndParentClassAndParentInterface`: The original parent class and interface and parent class and parent interface and parent class and parent interface object.

## 10.1 stub

```
<?php
class SomeClass
{
    public function doSomething()
    {
        // Do something.
    }
}
?>
```

## 10.2 stub

```
<?php
require_once 'SomeClass.php';

class StubTest extends PHPUnit\Framework\TestCase
{
    public function testStub()
    {
        // Create a stub for the SomeClass class.
        $stub = $this->getMock('SomeClass');

        // Configure the stub.
        $stub->expects($this->any())
            ->method('doSomething')
            ->will($this->returnValue('foo'));

        // Calling $stub->doSomething() will now return
        // 'foo'.
        $this->assertEquals('foo', $stub->doSomething());
    }
}
?>
```

PHPUnit getMock() 가 ,  
 PHP getMock()  
 .  
 • 가 NULL  
 will(\$this->returnValue())  
 • 2 ,  
 NULL ,  
 • 3 constructor  
 ( constructor ).  
 • 4 ,  
 • 5 constructor  
 • 6 clone constructor  
 • 7 , \_\_autoload()  
 mock builder API . 10.3 "mock builder  
 API " 가 . mock builder 가

- setMethods(array \$methods) mock builder . setMethods(NULL)
  - setConstructorArgs(array \$args) , constructor  
 ( constructor ).
  - setMockClassName(\$name)
  - disableOriginalConstructor() constructor
  - disableOriginalClone() clone constructor
  - disableAutoload() \_\_autoload()
- 10.3 mock builder API

```
<?php
require_once 'SomeClass.php';

class StubTest extends PHPUnit_Framework_TestCase
{
    public function testStub()
    {
        // Create a stub for the SomeClass class.
        $stub = $this->getMockBuilder('SomeClass')
            ->disableOriginalConstructor()
            ->getMock();

        // Configure the stub.
        $stub->expects($this->any())
            ->method('doSomething')
            ->will($this->returnValue('foo'));

        // Calling $stub->doSomething() will now return
        // 'foo'.
        $this->assertEquals('foo', $stub->doSomething());
    }
}
```

```

    }
}
?>

```

10.4 " stub ( ) stub 가  
 stub" returnArgument() returnValue()

10.4 stub

```

<?php
require_once 'SomeClass.php';

class StubTest extends PHPUnit_Framework_TestCase
{
    public function testReturnArgumentStub()
    {
        // Create a stub for the SomeClass class.
        $stub = $this->getMock('SomeClass');

        // Configure the stub.
        $stub->expects($this->any())
            ->method('doSomething')
            ->will($this->returnArgument(0));

        // $stub->doSomething('foo') returns 'foo'
        $this->assertEquals('foo', $stub->doSomething('foo'));

        // $stub->doSomething('bar') returns 'bar'
        $this->assertEquals('bar', $stub->doSomething('bar'));
    }
}
?>

```

10.5: stub (reference)  
 stub" 가

```

<?php
require_once 'SomeClass.php';

class StubTest extends PHPUnit_Framework_TestCase
{
    public function testReturnSelf()
    {
        // Create a stub for the SomeClass class.
        $stub = $this->getMock('SomeClass');

        // Configure the stub.
        $stub->expects($this->any())

```

```

        ->method('doSomething')
        ->will($this->returnSelf());

        // $stub->doSomething() returns $stub
        $this->assertSame($stub, $stub->doSomething());
    }
}
?>

```

stub  
returnValueMap()                    map                    ,                    .                    10.6 "

map                    stub" 가                    .

10.6                    map                    stub

```

<?php
require_once 'SomeClass.php';

class StubTest extends PHPUnit_Framework_TestCase
{
    public function testReturnValueMapStub()
    {
        // Create a stub for the SomeClass class.
        $stub = $this->getMock('SomeClass');

        // Create a map of arguments to return values.
        $map = array(
            array('a', 'b', 'c', 'd'),
            array('e', 'f', 'g', 'h')
        );

        // Configure the stub.
        $stub->expects($this->any())
            ->method('doSomething')
            ->will($this->returnValueMap($map));

        // $stub->doSomething() returns different values depending on
        // the provided arguments.
        $this->assertEquals('d', $stub->doSomething('a', 'b', 'c'));
        $this->assertEquals('h', $stub->doSomething('e', 'f', 'g'));
    }
}
?>

```

stub                    ,                    (returnValue() )                    ( )                    (returnArgument() ) 가                    stub

callback                    , returnCallback()                    .                    stub

stub"                    .                    10.7 "                    callback

10.7                    callback                    stub

```

<?php
require_once 'SomeClass.php';

class StubTest extends PHPUnit_Framework_TestCase
{
    public function testReturnCallbackStub()
    {
        // Create a stub for the SomeClass class.
        $stub = $this->getMock('SomeClass');

        // Configure the stub.
        $stub->expects($this->any())
            ->method('doSomething')
            ->will($this->returnCallback('str_rot13'));

        // $stub->doSomething($argument) returns str_rot13($argument)
        $this->assertEquals('fbzrguvat',
        $stub->doSomething('something'));
    }
}
?>

```

10.8 method stub onConsecutiveCalls() stub" 가

```

<?php
require_once 'SomeClass.php';

class StubTest extends PHPUnit_Framework_TestCase
{
    public function testOnConsecutiveCallsStub()
    {
        // Create a stub for the SomeClass class.
        $stub = $this->getMock('SomeClass');

        // Configure the stub.
        $stub->expects($this->any())
            ->method('doSomething')
            ->will($this->onConsecutiveCalls(2, 3, 5, 7));

        // $stub->doSomething() returns a different value each time
        $this->assertEquals(2, $stub->doSomething());
        $this->assertEquals(3, $stub->doSomething());
        $this->assertEquals(5, $stub->doSomething());
    }
}
?>

```

stub , . 10.9 " throw  
 stub" throwException() .  
 10.9 throw stub

```

<?php
require_once 'SomeClass.php';

class StubTest extends PHPUnit_Framework_TestCase
{
    public function testThrowExceptionStub()
    {
        // Create a stub for the SomeClass class.
        $stub = $this->getMock('SomeClass');

        // Configure the stub.
        $stub->expects($this->any())
            ->method('doSomething')
            ->will($this->throwException(new Exception));

        // $stub->doSomething() throws Exception
        $stub->doSomething();
    }
}
?>
    
```

stub  
 (façade) , stub ,  
 , IDatabase Database  
 , IDatabase stub 가  
 stub 가 가  
 stub , ,

**Notes**

# PHPUnitManual:10.2

## 10.2 Mock

가 mock  
 mock SUT mock  
 stub  
 SUT , mock stub 가  
 가 (update() 가  
 . 10.10 " (SUT) Subject Observer  
 Observer " (SUT) Subject Observer  
 .  
 10.10 (SUT) Subject Observer

```

<?php
class Subject
{
    protected $observers = array();

    public function attach(Observer $observer)
    {
        $this->observers[] = $observer;
    }

    public function doSomething()
    {
        // Do something.
        // ...

        // Notify observers that we did something.
        $this->notify('something');
    }

    public function doSomethingBad()
    {
        foreach ($this->observers as $observer) {
            $observer->reportError(42, 'Something bad happened', $this);
        }
    }

    protected function notify($argument)
    {
        foreach ($this->observers as $observer) {
            $observer->update($argument);
        }
    }
}
    
```



```

    }

    // Other methods.
}

class Observer
{
    public function update($argument)
    {
        // Do something.
    }

    public function reportError($errorCode, $errorMessage, Subject
$subject)
    {
        // Do something
    }

    // Other methods.
}
?>

```

10.11 " 가 " mock  
 Subject Observer .  
 , PHPUnit\_Framework\_TestCase getMock() Observer mock  
 . getMock() 2 , Observer  
 update() mock .  
 10.11 가

```

<?php
class SubjectTest extends PHPUnit_Framework_TestCase
{
    public function testObserversAreUpdated()
    {
        // Create a mock for the Observer class,
        // only mock the update() method.
        $observer = $this->getMock('Observer', array('update'));

        // Set up the expectation for the update() method
        // to be called only once and with the string 'something'
        // as its parameter.
        $observer->expects($this->once())
            ->method('update')
            ->with($this->equalTo('something'));

        // Create a Subject object and attach the mocked
        // Observer object to it.
        $subject = new Subject;
    }
}

```

```

        $subject->attach($observer);

        // Call the doSomething() method on the $subject object
        // which we expect to call the mocked Observer object's
        // update() method with the string 'something'.
        $subject->doSomething();
    }
}
?>

```

with()

mock

10.12 가 가

```

<?php
class SubjectTest extends PHPUnit_Framework_TestCase
{
    public function testErrorReported()
    {
        // Create a mock for the Observer class, mocking the
        // reportError() method
        $observer = $this->getMock('Observer', array('reportError'));

        $observer->expects($this->once())
            ->method('reportError')
            ->with($this->greaterThan(0),
                $this->stringContains('Something'),
                $this->anything());

        $subject = new Subject;
        $subject->attach($observer);

        // The doSomethingBad() method should report an error to the
observer
        // via the reportError() method
        $subject->doSomethingBad();
    }
}
?>

```

4.3 " " 가 , 10.1 "Matchers"  
 가 matcher .

Matcher	
PHPUnit_Framework_MockObject_Matcher_AnyInvokedCount any()	가 가 0
PHPUnit_Framework_MockObject_Matcher_InvokedCount never()	가 가
PHPUnit_Framework_MockObject_Matcher_InvokedAtLeastOnce atLeastOnce()	가 가 1
PHPUnit_Framework_MockObject_Matcher_InvokedCount once()	가 가
PHPUnit_Framework_MockObject_Matcher_InvokedCount exactly(int \$count)	가 가 \$count
PHPUnit_Framework_MockObject_Matcher_InvokedAtIndex at(int \$index)	가 가 \$index

10.1. Matchers

getMockForAbstractClass() mock (concrete)

10.13:

```
<?php
abstract class AbstractClass
{
    public function concreteMethod()
    {
        return $this->abstractMethod();
    }

    public abstract function abstractMethod();
}

class AbstractClassTest extends PHPUnit_Framework_TestCase
{
    public function testConcreteMethod()
    {
        $stub = $this->getMockForAbstractClass('AbstractClass');
        $stub->expects($this->any())
            ->method('abstractMethod')
            ->will($this->returnValue(TRUE));

        $this->assertTrue($stub->concreteMethod());
    }
}
?>
```

10.14 가 , 가

```
<?php
class FooTest extends PHPUnit_Framework_TestCase
```

```

{
    public function testIdenticalObjectPassed()
    {
        $expectedObject = new stdClass;

        $mock = $this->getMock('stdClass', array('foo'));
        $mock->expects($this->once())
            ->method('foo')
            ->with($this->identicalTo($expectedObject));

        $mock->foo($expectedObject);
    }
}
?>

```

10.15:          clone                  mock

```

<?php
class FooTest extends PHPUnit_Framework_TestCase
{
    public function testIdenticalObjectPassed()
    {
        $cloneArguments = true;

        $mock = $this->getMock(
            'stdClass',
            array(),
            array(),
            '',
            FALSE,
            TRUE,
            TRUE,
            $cloneArguments
        );

        // or using the mock builder
        $mock =
$this->getMockBuilder('stdClass')->enableArgumentCloning()->getMock();

        // now your mock clones parameters so the identicalTo
constraint will fail.
    }
}
?>

```

Notes

# PHPUnitManual:10.3

---

10.3 stub mock

가 . stub mockgetMockFromWsdل() 가  
 . getMock() ( )  
 getMockFromWsdل() 가 stub mock PHP

10.16 " stub" getMockFromWsdل() GoogleSearch.wsdل  
 stub .

10.16 stub

```
<?php
class GoogleTest extends PHPUnit_Framework_TestCase
{
    public function testSearch()
    {
        $googleSearch = $this->getMockFromWsdل(
            'GoogleSearch.wsdل', 'GoogleSearch'
        );

        $directoryCategory = new StdClass;
        $directoryCategory->fullViewableName = '';
        $directoryCategory->specialEncoding = '';

        $element = new StdClass;
        $element->summary = '';
        $element->URL = 'http://www.phpunit.de/';
        $element->snippet = '...';
        $element->title = '<b>PHPUnit</b>';
        $element->cachedSize = '11k';
        $element->relatedInformationPresent = TRUE;
        $element->hostName = 'www.phpunit.de';
        $element->directoryCategory = $directoryCategory;
        $element->directoryTitle = '';

        $result = new StdClass;
        $result->documentFiltering = FALSE;
        $result->searchComments = '';
        $result->estimatedTotalResultsCount = 378000;
        $result->estimateIsExact = FALSE;
        $result->resultElements = array($element);
        $result->searchQuery = 'PHPUnit';
        $result->startIndex = 1;
    }
}
```

```

$result->endIndex = 1;
$result->searchTips = '';
$result->directoryCategories = array();
$result->searchTime = 0.248822;

$googleSearch->expects($this->any())
    ->method('doGoogleSearch')
    ->will($this->returnValue($result));

/**
 * $googleSearch->doGoogleSearch() will now return a stubbed
result and
 * the web service's doGoogleSearch() method will not be
invoked.
 */
$this->assertEquals(
    $result,
    $googleSearch->doGoogleSearch(
        '00000000000000000000000000000000',
        'PHPUnit',
        0,
        1,
        FALSE,
        '',
        FALSE,
        '',
        '',
        ''
    )
);
}
}
?>

```

## Notes

# PHPUnitManual:10.4

10.4 mock  
 vfsStream[1] 가 [2] stream wrapper[3] , mock

vfsStream , PEAR PEAR (pear.bovigo.org)

```
localhost# pear channel-discover pear.bovigo.org
```

PEAR vfsStream

```
pear install bovigo/vfsStream-beta
```

10.17 " "  
 10.17

```
<?php
class Example
{
    protected $id;
    protected $directory;

    public function __construct($id)
    {
        $this->id = $id;
    }

    public function setDirectory($directory)
    {
        $this->directory = $directory . DIRECTORY_SEPARATOR . $this->id;

        if (!file_exists($this->directory)) {
            mkdir($this->directory, 0700, TRUE);
        }
    }
}
}??>
```

vfsStream 가 , setDirectory()  
 ( 10.18 ).

10.18

```
<?php
require_once 'Example.php';

class ExampleTest extends PHPUnit_Framework_TestCase
{
```

```

protected function setUp()
{
    if (file_exists(dirname(__FILE__) . '/id')) {
        rmdir(dirname(__FILE__) . '/id');
    }
}

public function testDirectoryIsCreated()
{
    $example = new Example('id');
    $this->assertFalse(file_exists(dirname(__FILE__) . '/id'));

    $example->setDirectory(dirname(__FILE__));
    $this->assertTrue(file_exists(dirname(__FILE__) . '/id'));
}

protected function tearDown()
{
    if (file_exists(dirname(__FILE__) . '/id')) {
        rmdir(dirname(__FILE__) . '/id');
    }
}
}
?>

```

가

- 가 가 가
- setUp() tearDown() 가
- tearDown() 가 , 가
- 10.19 " mock " vfsStream
- mock
- 10.19 mock

```

<?php
require_once 'vfsStream/vfsStream.php';
require_once 'Example.php';

class ExampleTest extends PHPUnit_Framework_TestCase
{
    public function setUp()
    {
        vfsStreamWrapper::register();
        vfsStreamWrapper::setRoot(new vfsStreamDirectory('exampleDir'));
    }

    public function testDirectoryIsCreated()
    {

```



```

    $example = new Example('id');
    $this->assertFalse(vfsStreamWrapper::getRoot()->hasChild('id'));

    $example->setDirectory(vfsStream::url('exampleDir'));
    $this->assertTrue(vfsStreamWrapper::getRoot()->hasChild('id'));
}
}
?>

```

- 가 .
- 가
- 가
- 가 , tearDown() .

**Notes**

# PHPUnitManual:11

---

11 (Testing Practices)

You can always write more tests. However, you will quickly find that only a fraction of the tests you can imagine are actually useful. What you want is to write tests that fail even though you think they should work, or tests that succeed even though you think they should fail. Another way to think of it is in cost/benefit terms. You want to write tests that will pay you back with information.

--Erich Gamma

**Notes**

---

# PHPUnitManual:11.1

---

## 11.1 (During Development)

PHPUnit의 test suite 가 (refactorings)[1] 가 test suite 가 , 가

1. 가 .
2. 가 .
3. 가 .
4. 가 ,가 . 가

### Notes

# PHPUnitManual:11.2

---

## 11.2 (During Debugging)

PHPUnit의 test suite 가 (refactorings)[1] 가 test suite 가 , 가

1. 가 .
2. 가 .
3. ( 가 ) .
4. 가 .

가 가 . (Unit testing offers many advantages)

- 가 (patch) . (Testing gives code authors and reviewers confidence that patches produce the correct results.)
- . (Authoring testcases is a good impetus for developers to discover edge cases.)
- (regression) ,
- API 가 . 가 . (Unit tests provide working examples for how to use an API and can significantly aid documentation efforts.)

가  
( ) ( )

### Notes

# PHPUnitManual:12

12 (Test-Driven Development)

Test-First Programming, Extreme Programming[1], Test-Driven Development[2]

(Design-by-Contract[3])

PHPUnit 가 가 가 " "

가 , ?

Extreme Programming Test-Driven Development Test-First Programming

가 가

Test-First Programming (object-oriented programming)

가) 가 (

가

The point of Test-Driven Development[2] is to drive out the functionality the software actually needs, rather than what the programmer thinks it probably ought to have. The way it does this seems at first counterintuitive, if not downright silly, but it not only makes sense, it also quickly becomes a natural and elegant way to develop software.

Test-Driven Development[2] 가 가 가 ,가

--Dan North

Test-Driven Development ,  
 Test-Driven Development (Beck2002) by Kent Beck Dave Astels' A Practical Guide to Test-Driven  
 Development (Astels2003)

Notes

# PHPUnitManual:12.1

---

12.1

- 가 BankAccount 2가
- 0
- 가 0
- BankAccount
- 12.1 "BankAccount"

12.1 BankAccount

```
<?php
require_once 'BankAccount.php';

class BankAccountTest extends PHPUnit_Framework_TestCase
{
    protected $ba;

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

    public function testBalanceIsInitiallyZero()
    {
        $this->assertEquals(0, $this->ba->getBalance());
    }

    public function testBalanceCannotBecomeNegative()
    {
        try {
            $this->ba->withdrawMoney(1);
        }

        catch (BankAccountException $e) {
            $this->assertEquals(0, $this->ba->getBalance());

            return;
        }

        $this->fail();
    }
}
```

```

public function testBalanceCannotBecomeNegative2()
{
    try {
        $this->ba->depositMoney(-1);
    }

    catch (BankAccountException $e) {
        $this->assertEquals(0, $this->ba->getBalance());

        return;
    }

    $this->fail();
}
?>

```

```

, testBalanceIsInitiallyZero()
. BankAccount getBalance() 12.2 " testBalanceIsInitiallyZero()
"
12.2 testBalanceIsInitiallyZero()

```

```

<?php
class BankAccount
{
    protected $balance = 0;

    public function getBalance()
    {
        return $this->balance;
    }
}
?>

```

가

```

phpunit BankAccountTest
PHPUnit 3.7.0 by Sebastian Bergmann.

Fatal error: Call to undefined method BankAccount::withdrawMoney()

```

```

, withdrawMoney(), depositMoney(), setBalance()
12.3 " BankAccount "
, BankAccountException
12.3 BankAccount

```

```
<?php
class BankAccount
{
    protected $balance = 0;

    public function getBalance()
    {
        return $this->balance;
    }

    protected function setBalance($balance)
    {
        if ($balance >= 0) {
            $this->balance = $balance;
        } else {
            throw new BankAccountException;
        }
    }

    public function depositMoney($balance)
    {
        $this->setBalance($this->getBalance() + $balance);

        return $this->getBalance();
    }

    public function withdrawMoney($balance)
    {
        $this->setBalance($this->getBalance() - $balance);

        return $this->getBalance();
    }
}
?>
```

```
phpunit BankAccountTest
PHPUnit 3.7.0 by Sebastian Bergmann.

...

Time: 0 seconds

OK (3 tests, 3 assertions)
```

PHPUnit\_Framework\_Assert 가  
 " " . 12.4 ""  
 BankAccount "가  
 PHPUnit\_Framework\_AssertionFailedError 가  
 가 PHPUnit  
 12.4 " " BankAccount

```
<?php
class BankAccount
{
    private $balance = 0;

    public function getBalance()
    {
        return $this->balance;
    }

    protected function setBalance($balance)
    {
        PHPUnit_Framework_Assert::assertTrue($balance >= 0);

        $this->balance = $balance;
    }

    public function depositMoney($amount)
    {
        PHPUnit_Framework_Assert::assertTrue($amount >= 0);

        $this->setBalance($this->getBalance() + $amount);

        return $this->getBalance();
    }

    public function withdrawMoney($amount)
    {
        PHPUnit_Framework_Assert::assertTrue($amount >= 0);
        PHPUnit_Framework_Assert::assertTrue($this->balance >= $amount);

        $this->setBalance($this->getBalance() - $amount);

        return $this->getBalance();
    }
}
?>
```

" " BankAccount  
 , Test-First Programming  
 가 , setBalance(), depositMoney(),

withdrawMoney()

가 , 가 "code-coverage "

Notes

# PHPUnitManual:13

## 13 Behaviour-Driven Development

Astels2006 Dave Astels

- Extreme Programming[1] 가 가
- Extreme Programming (Test-Driven Development)[2] (12 )
- (specification) (assertion)

가 가?

It's about figuring out what you are trying to do before you run off half-cocked to try to do it. You write a specification that nails down a small aspect of behaviour in a concise, unambiguous, and executable form. It's that simple. Does that mean you write tests? No. It means you write specifications of what your code will have to do. It means you specify the behaviour of your code ahead of time. But not far ahead of time. In fact, just before you write the code is best because that's when you have as much information at hand as you will up to that point. Like well done TDD, you work in tiny increments... specifying one small aspect of behaviour at a time, then implementing it.

가 , 가  
 가 ?  
 가 " 가 가" (specification)

When you realize that it's all about specifying behaviour and not writing tests, your point of view shifts. Suddenly the idea of having a Test class for each of your production classes is ridiculously limiting. And the thought of testing each of your methods with its own test method (in a 1-1 relationship) will be laughable.

1 1

--Dave Astels

Behaviour-Driven Development[1] 가  
 (interactions) .BDD , Domain-Driven Design[2]  
 가" Domain 가가  
 Domain



PHPUnit\_Extensions\_Story\_TestCase      BDD      (Domain-Specific Language)

```
localhost# pear install phpunit/PHPUnit_Story
```

- scenario      , given(), when(),      then()      step      . and()      step
- PHPUnit\_Extensions\_Story\_TestCase      abstract      ,
- runGiven(&\$world, \$action, \$arguments)
  - runWhen(&\$world, \$action, \$arguments)
  - runThen(&\$world, \$action, \$arguments)

**Notes**

# PHPUnitManual:13.1

---

13.1

- 10
- 10      2      가      가
- , 2      10
- ,      10
- ,      2

13.1 "BowlingGame"      PHPUnit\_Extensions\_Story\_TestCase

13.1: BowlingGame

```
<?php
require_once 'PHPUnit/Extensions/Story/TestCase.php';
require_once 'BowlingGame.php';

class BowlingGameSpec extends PHPUnit_Extensions_Story_TestCase
{
    /**
     * @scenario
     */
    public function scoreForGutterGameIs0 ()
    {
        $this->given('New game')
            ->then('Score should be', 0);
    }
}
```

```

}

/**
 * @scenario
 */
public function scoreForAllOnesIs20()
{
    $this->given('New game')
        ->when('Player rolls', 1)
        ->and('Player rolls', 1)
        ->and('Player rolls', 1)
        ->and('Player rolls', 1)
        ->and('Player rolls', 1)
        ->and('Player rolls', 1)
        ->and('Player rolls', 1)
        ->and('Player rolls', 1)
        ->and('Player rolls', 1)
        ->and('Player rolls', 1)
        ->and('Player rolls', 1)
        ->and('Player rolls', 1)
        ->and('Player rolls', 1)
        ->and('Player rolls', 1)
        ->and('Player rolls', 1)
        ->and('Player rolls', 1)
        ->and('Player rolls', 1)
        ->and('Player rolls', 1)
        ->and('Player rolls', 1)
        ->and('Player rolls', 1)
        ->and('Player rolls', 1)
        ->and('Player rolls', 1)
        ->and('Player rolls', 1)
        ->then('Score should be', 20);
}

/**
 * @scenario
 */
public function scoreForOneSpareAnd3Is16()
{
    $this->given('New game')
        ->when('Player rolls', 5)
        ->and('Player rolls', 5)
        ->and('Player rolls', 3)
        ->then('Score should be', 16);
}

/**
 * @scenario
 */
public function scoreForOneStrikeAnd3And4Is24()

```

```
{
    $this->given('New game')
        ->when('Player rolls', 10)
        ->and('Player rolls', 3)
        ->and('Player rolls', 4)
        ->then('Score should be', 24);
}

/**
 * @scenario
 */
public function scoreForPerfectGameIs300()
{
    $this->given('New game')
        ->when('Player rolls', 10)
        ->and('Player rolls', 10)
        ->and('Player rolls', 10)
        ->and('Player rolls', 10)
        ->and('Player rolls', 10)
        ->and('Player rolls', 10)
        ->and('Player rolls', 10)
        ->and('Player rolls', 10)
        ->and('Player rolls', 10)
        ->and('Player rolls', 10)
        ->and('Player rolls', 10)
        ->and('Player rolls', 10)
        ->and('Player rolls', 10)
        ->and('Player rolls', 10)
        ->then('Score should be', 300);
}

public function runGiven(&$world, $action, $arguments)
{
    switch($action) {
        case 'New game': {
            $world['game'] = new BowlingGame;
            $world['rolls'] = 0;
        }
        break;

        default: {
            return $this->notImplemented($action);
        }
    }
}

public function runWhen(&$world, $action, $arguments)
{
    switch($action) {
```

```

        case 'Player rolls': {
            $world['game']->roll($arguments[0]);
            $world['rolls']++;
        }
        break;

        default: {
            return $this->notImplemented($action);
        }
    }
}

public function runThen(&$world, $action, $arguments)
{
    switch($action) {
        case 'Score should be': {
            for ($i = $world['rolls']; $i < 20; $i++) {
                $world['game']->roll(0);
            }

            $this->assertEquals($arguments[0],
                $world['game']->score());
        }
        break;

        default: {
            return $this->notImplemented($action);
        }
    }
}
?>

```

```

phpunit --printer PHPUnit_Extensions_Story_ResultPrinter_Text
BowlingGameSpec
PHPUnit 3.7.0 by Sebastian Bergmann.

BowlingGameSpec
[x] Score for gutter game is 0

    Given New game
    Then Score should be 0

[x] Score for all ones is 20

    Given New game
    When Player rolls 1
    and Player rolls 1

```



```

and Player rolls 10
and Player rolls 10
and Player rolls 10
and Player rolls 10
Then Score should be 300

```

Scenarios: 5, Failed: 0, Skipped: 0, Incomplete: 0.

### Notes

# PHPUnitManual:14

**WARNING: Article could not be rendered - ouputting plain text.**

Potential causes of the problem are: (a) a bug in the pdf-writer software (b) problematic Mediawiki markup (c) table is too wide

140. Code Coverage The beauty of testing is found not in the effort but in the efficiency. --Murali Nandigama PHPUnit code coverage covered) Code coverage 100% BankAccountTest PHPUnit 3.7.0 by Sebastian Bergmann. ... Time: 0 seconds OK (3 tests, 3 assertions) Generating report, this may take a moment. Figure 14.1 "setBalance() Code coverage" Code coverage setBalance(), depositMoney() withdrawMoney() BankAccountTest 14.1 "Code coverage BankAccountTest test case" 14.1 Code coverage <?php require\_once 'BankAccount.php'; class BankAccountTest extends PHPUnit\_Framework\_TestCase { // ... public function testDepositWithdrawMoney() { \$this->assertEquals(0, \$this->ba->getBalance()); \$this->ba->depositMoney(1); \$this->assertEquals(1, \$this->ba->getBalance()); \$this->ba->withdrawMoney(1); \$this->assertEquals(0, \$this->ba->getBalance()); } } ?> Figure 14.2 ?setBalance() Code coverage" setBalance() Code coverage Figure 14.2 setBalance() Code coverage Notes

# PHPUnitManual:14.1

---

## 14.1 Cover

`@covers` (annotation) ( Table B.1 "Cover  
 , 가 . ,  
 Code coverage .  
 14.2 " 가 .  
 14.2

```

<?php
require_once 'BankAccount.php';

class BankAccountTest extends PHPUnit_Framework_TestCase
{
    protected $ba;

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

    /**
     * @covers BankAccount::getBalance
     */
    public function testBalanceIsInitiallyZero()
    {
        $this->assertEquals(0, $this->ba->getBalance());
    }

    /**
     * @covers BankAccount::withdrawMoney
     */
    public function testBalanceCannotBecomeNegative()
    {
        try {
            $this->ba->withdrawMoney(1);
        }

        catch (BankAccountException $e) {
            $this->assertEquals(0, $this->ba->getBalance());

            return;
        }

        $this->fail();
    }
}
    
```

```

/**
 * @covers BankAccount::depositMoney
 */
public function testBalanceCannotBecomeNegative2 ()
{
    try {
        $this->ba->depositMoney(-1);
    }

    catch (BankAccountException $e) {
        $this->assertEquals(0, $this->ba->getBalance());

        return;
    }

    $this->fail();
}

/**
 * @covers BankAccount::getBalance
 * @covers BankAccount::depositMoney
 * @covers BankAccount::withdrawMoney
 */

public function testDepositWithdrawMoney ()
{
    $this->assertEquals(0, $this->ba->getBalance());
    $this->ba->depositMoney(1);
    $this->assertEquals(1, $this->ba->getBalance());
    $this->ba->withdrawMoney(1);
    $this->assertEquals(0, $this->ba->getBalance());
}
}
?>

```

```

@coversNothing , ( "@coversNothing" ).
, Code coverage .
14.3

```

```

<?php
class GuestbookIntegrationTest extends
PHPUnit_Extensions_Database_TestCase
{
    /**
     * @coversNothing
     */
    public function testAddEntry ()

```



```

{
    $guestbook = new Guestbook();
    $guestbook->addEntry("suzy", "Hello world!");

    $queryTable = $this->getConnection()->createQueryTable(
        'guestbook', 'SELECT * FROM guestbook'
    );
    $expectedTable = $this->createFlatXmlDataSet("expectedBook.xml")
        ->getTable("guestbook");
    $this->assertTablesEqual($expectedTable, $queryTable);
}
}
?>

```

Notes

# PHPUnitManual:14.2

14.2 Code Block

Code coverage

```

PHPUnit
@codeCoverageIgnoreEnd
@codeCoverageIgnoreStart
14.4 @codeCoverageIgnore, @codeCoverageIgnoreStart

```

가 14.4 "@codeCoverageIgnore, @codeCoverageIgnoreEnd"

```

<?php
/**
 * @codeCoverageIgnore
 */
class Foo
{
    public function bar()
    {
    }
}

class Bar
{
    /**
     * @codeCoverageIgnore
     */
    public function foo()
    {
    }
}

```

```

if (FALSE) {
    // @codeCoverageIgnoreStart
    print '*';
    // @codeCoverageIgnoreEnd
}
?>

```

### Notes

## PHPUnitManual:14.3

---

14.3

1 가 ( )

가 ,PHPUnit ( ) , 가

가 가 , 가 " "

PHPUnit , processUncoveredFilesFromWhitelist="true" ("Code coverage 가/ ")

) 가 , PHP\_CodeCoverage include , 가 가

processUncoveredFilesFromWhitelist="true" 가 , scope

가 가

PHPUnit XML ("Code Coverage 가/ ") ,

/ Code coverage

### Notes

# PHPUnitManual:14.4

---

## 14.4 Edge cases

, PHPUnit " (line based)" code coverage  
Edge case 가 가 .

### 14.5

```

<?php
// Coverage " " 가 " "
// , Coverage 가 .
if (false) this_function_call_shows_up_as_covered();

// Code coverage , 2 .
// " "
if (false)
// " 가 "
// , if coverage 가
will_also_show_up_as_coveraged();

// ,
if (false) {
    this_call_will_never_show_up_as_covered();
}
?>
    
```

## Notes

# PHPUnitManual:15

15

## Notes

# PHPUnitManual:15.1

## 15.1 (Agile Documentation)

, Extreme Programming  
 (collective code ownership) 가 , Extreme Programming 가  
 , PHPUnit TestDox 가 " "  
 PHPUnit TestDox , PHP  
 , testBalanceIsInitiallyZero() 가 "Balance is initially  
 zero" 가 ( : testBalanceCannotBecomeNegative()  
 testBalanceCannotBecomeNegative2()), "Balance cannot become negative"  
 가  
 BankAccount (from 12.1 "BankAccount " )

```
phpunit --testdox BankAccountTest
PHPUnit 3.7.0 by Sebastian Bergmann.

BankAccount
[x] Balance is initially zero
[x] Balance cannot become negative
```

, HTML PLAIN TEXT  
 --testdox-html --testdox-text  
 , 가 가  
 , 가 가  
 , " " test case 가  
 가

**Notes**

# PHPUnitManual:15.2

---

15.2 (Cross-Team Tests)

가 .

,

.

,

test case . " "

,

가 .

Stub ("Mock Objects" ) 가 .

가 , 가 .

가 stub

,2 .

**Notes**

# PHPUnitManual:16

---

16 Skeleton Generator

PHPUnit Skeleton Generator , . PHPUnit Skeleton Generator

```
localhost# pear install phpunit/PHPUnit_SkeletonGenerator
```

**Notes**

---

# PHPUnitManual:16.1

16.1 Test Case Skeleton

```
public function testMethod()
{
}
```

PHPUnit Skeleton Generator skeleton

16.1 Calculator

```
<?php
class Calculator
{
    public function add($a, $b)
    {
        return $a + $b;
    }
}
```

, Calculator ( "16.1 Calculator" ) skeleton

```
phpunit-skelgen --test Calculator
PHPUnit Skeleton Generator 1.0.0 by Sebastian Bergmann.

Wrote skeleton for "CalculatorTest" to "/home/sb/CalculatorTest.php".
```

, test case test case ( 9 )

**Namespace** 가 **Skeleton Generator**  
 Namespace Skeleton Generator ,  
 qualified name , 가  
 , namespace Project Calculator 가 , Skeleton  
 Generator .

```
phpunit-skelgen --test -- "project\Calculator" Calculator.php
PHPUnit Skeleton Generator 1.0.0 by Sebastian Bergmann.

Wrote skeleton for "project\CalculatorTest" to
"/home/sb/CalculatorTest.php".
```

```
phpunit --bootstrap Calculator.php --verbose CalculatorTest
PHPUnit 3.7.0 by Sebastian Bergmann.

I

Time: 0 seconds, Memory: 3.50Mb

There was 1 incomplete test:

1) CalculatorTest::testAdd
This test has not been implemented yet.

/home/sb/CalculatorTest.php:38
OK, but incomplete or skipped tests!
Tests: 1, Assertions: 0, Incomplete: 1.
```

```
@assert
test case 가 . 16.2 "@assert
Calculator " 가 .
16.2 @assert
Calculator
```

```
<?php
class Calculator
{
    /**
     * @assert (0, 0) == 0
     * @assert (0, 1) == 1
     * @assert (1, 0) == 1
     * @assert (1, 1) == 2
     */
    public function add($a, $b)
    {
        return $a + $b;
    }
}
?>
```

```
, @assert . , 가
```

```
/**
 * Generated from @assert (0, 0) == 0.
 */
public function testAdd() {
    $o = new Calculator;
    $this->assertEquals(0, $o->add(0, 0));
}
```

test case .

```
phpunit --bootstrap Calculator.php --verbose CalculatorTest
PHPUnit 3.7.0 by Sebastian Bergmann.

....

Time: 0 seconds, Memory: 3.50Mb

OK (4 tests, 4 assertions)
```

@assert (...) == X	assertEquals(X, method(...))
@assert (...) != X	assertNotEquals(X, method(...))
@assert (...) === X	assertSame(X, method(...))
@assert (...) !== X	assertNotSame(X, method(...))
@assert (...) > X	assertGreaterThan(X, method(...))
@assert (...) >= X	assertGreaterThanOrEqual(X, method(...))
@assert (...) < X	assertLessThan(X, method(...))
@assert (...) <= X	assertLessThanOrEqual(X, method(...))
@assert (...) throws X	@expectedException X
16.1 @assert	

## Notes



# PHPUnitManual:16.2

16.2 Test case skeleton

```

    ( 12 )
    . PHPUnit test case skeleton
    , Unit UnitTest 가 . test case
    , Unit 가
    . 16.4 " BowlingGame skeleton"
    skeleton 16.3 "BowlingGame "
    16.3 BowlingGameTest
    
```

```

<?php
class BowlingGameTest extends PHPUnit_Framework_TestCase
{
    protected $game;

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

    protected function rollMany($n, $pins)
    {
        for ($i = 0; $i < $n; $i++) {
            $this->game->roll($pins);
        }
    }

    public function testScoreForGutterGameIs0()
    {
        $this->rollMany(20, 0);
        $this->assertEquals(0, $this->game->score());
    }
}
?>
    
```

```

phpunit-skelgen --class BowlingGameTest
PHPUnit Skeleton Generator 1.0.0 by Sebastian Bergmann.

Wrote skeleton for "BowlingGame" to "./BowlingGame.php".
    
```

```

16.4 BowlingGame skeleton

<?php
/**
 * Generated by PHPUnit_SkeletonGenerator on 2012-01-09 at 16:55:58.
 */
    
```

```
class BowlingGame
{
    /**
     * @todo Implement roll().
     */
    public function roll()
    {
        // Remove the following line when you implement this method.
        throw new RuntimeException('Not yet implemented.');
```

```
    }

    /**
     * @todo Implement score().
     */
    public function score()
    {
        // Remove the following line when you implement this method.
        throw new RuntimeException('Not yet implemented.');
```

```
    }
}
?>
```

```
phpunit --bootstrap BowlingGame.php BowlingGameTest
PHPUnit 3.7.0 by Sebastian Bergmann.
```

E

Time: 0 seconds, Memory: 3.50Mb

There was 1 error:

1) BowlingGameTest::testScoreForGutterGameIs0  
RuntimeException: Not yet implemented.

```
/home/sb/BowlingGame.php:13
/home/sb/BowlingGameTest.php:14
/home/sb/BowlingGameTest.php:20
```

FAILURES!

Tests: 1, Assertions: 0, Errors: 1.

## Notes

# PHPUnitManual:17

---

17 PHPUnit Selenium

17 PHPUnit Selenium

## Notes

# PHPUnitManual:17.1

---

### 17.1 Selenium Server

Selenium Server[1]

OS

Selenium RC Selenium Core (http://seleniumhq.org/). Selenium Javascript Selenium 가 가 Selenium (acceptance testing, 가 OS / )

PHPUnit\_Selenium

, Selenium 2.x server

Selenium 2.x server

, 1.0

Selenium RC API

, PHPUnit\_Selenium 1.2

WebDriver API

, Selenium 2

, Selenium RC

## Notes

# PHPUnitManual:17.2

---

## 17.2 Selenium

, Selenium Server

### 1. Selenium Server

2. selenium-server-standalone-2.9.0.jar ( ) ( :/usr/local/bin).
3. java -jar /usr/local/bin/selenium-server-standalone-2.9.0.jar Selenium RC

, PHPUnit\_Selenium

PHPUnit

Selenium Server

```
localhost# pear install phpunit/PHPUnit_Selenium
```

Selenium server

## Notes

# PHPUnitManual:17.3

---

## 17.3 PHPUnit\_Extensions\_Selenium2TestCase

PHPUnit\_Extensions\_Selenium2TestCase test case WebDriver API ( ).

17.1 "PHPUnit\_Extensions\_Selenium2TestCase " , http://www.example.com/ <title> element

### 17.1 PHPUnit\_Extensions\_Selenium2TestCase

```
<?php
class WebTest extends PHPUnit_Extensions_Selenium2TestCase
{
    protected function setUp()
    {
        $this->setBrowser('firefox');
        $this->setBrowserUrl('http://www.example.com/');
    }

    public function testTitle()
    {
        $this->url('http://www.example.com/');
        $this->assertEquals('Example WWW Page', $this->title());
    }
}
?>
```

```
phpunit WebTest
PHPUnit 3.6.10 by Sebastian Bergmann.

F

Time: 28 seconds, Memory: 3.00Mb

There was 1 failure:

1) WebTest::testTitle
Failed asserting that two strings are equal.
--- Expected
+++ Actual
@@ @@
-'Example WWW Page'
+'IANA — Example domains'

/home/giorgio/WebTest.php:13

FAILURES!
Tests: 1, Assertions: 1, Failures: 1.
```

```
Selenium2TestCare __call() . 가
PHPUnit_Extensions_Selenium2TestCase end-to-end test[1] .
```

**Notes**

# PHPUnitManual:17.4

## 17.4 PHPUnit\_Extensions\_SeleniumTestCase

PHPUnit\_Extensions\_SeleniumTestCase test case , Selenium Server /

17.2 "PHPUnit\_Extensions\_SeleniumTestCase" , http://www.example.com/ <title>  
 element .

### 17.2 PHPUnit\_Extensions\_SeleniumTestCase

```
<?php
require_once 'PHPUnit/Extensions/SeleniumTestCase.php';

class WebTest extends PHPUnit_Extensions_SeleniumTestCase
{
    protected function setUp()
    {
        $this->setBrowser('*firefox');
        $this->setBrowserUrl('http://www.example.com/');
    }

    public function testTitle()
    {
        $this->open('http://www.example.com/');
        $this->assertTitle('Example WWW Page');
    }
}
?>
```

```
phpunit WebTest
PHPUnit 3.7.0 by Sebastian Bergmann.

F

Time: 9 seconds, Memory: 6.00Mb

There was 1 failure:

1) WebTest::testTitle
Current URL: http://www.iana.org/domains/example/

Failed asserting that 'IANA — Example domains' matches PCRE pattern
"/Example WWW Page/".

FAILURES!
Tests: 1, Assertions: 1, Failures: 1.
```

PHPUnit\_Framework\_TestCase , PHPUnit\_Extensions\_SeleniumTestCase test case  
 setUp() 가 , Selenium Server session  
 . 17.1 "Selenium Server API : Setup" 가 .

Method	
void setBrowser(string \$browser)	Selenium Server 가 .
void setBrowserUrl(string \$browserUrl)	base URL .
void setHost(string \$host)	Selenium Server HOST .
void setPort(int \$port)	Selenium Server port .
void setTimeout(int \$timeout)	Selenium Server timeout .
void setSleep(int \$seconds)	Selenium Server client 가 Selenium Server ( ) .
17.1 Selenium Server API : Setup	

PHPUnit Selenium 가 , screenshot .  
 , \$captureScreenshotOnFailure, \$screenshotPath, \$screenshotUrl test case  
 ( 17.3 " screenshot " .  
 17.3 screenshot

```
<?php
require_once 'PHPUnit/Extensions/SeleniumTestCase.php';

class WebTest extends PHPUnit_Extensions_SeleniumTestCase
{
    protected $captureScreenshotOnFailure = TRUE;
    protected $screenshotPath = '/var/www/localhost/htdocs/screenshots';
    protected $screenshotUrl = 'http://localhost/screenshots';

    protected function setUp()
    {
        $this->setBrowser('*firefox');
        $this->setBrowserUrl('http://www.example.com/');
    }

    public function testTitle()
    {
        $this->open('http://www.example.com/');
        $this->assertTitle('Example WWW Page');
    }
}
?>
```

```
phpunit WebTest
PHPUnit 3.7.0 by Sebastian Bergmann.

F
```

Time: 7 seconds, Memory: 6.00Mb

There was 1 failure:

1) WebTest::testTitle

Current URL: http://www.iana.org/domains/example/

Screenshot:

http://localhost/screenshots/334b080f2364b5f11568ee1c7f6742c9.png

Failed asserting that 'IANA — Example domains' matches PCRE pattern "/Example WWW Page/".

FAILURES!

Tests: 1, Assertions: 1, Failures: 1.

```

        , setBrowser()
    , test case      public static $browsers
        Selenium Server
        "
    17.4
    17.4
    
```

```

<?php
require_once 'PHPUnit/Extensions/SeleniumTestCase.php';

class WebTest extends PHPUnit_Extensions_SeleniumTestCase
{
    public static $browsers = array(
        array(
            'name' => 'Firefox on Linux',
            'browser' => '*firefox',
            'host' => 'my.linux.box',
            'port' => 4444,
            'timeout' => 30000,
        ),
        array(
            'name' => 'Safari on MacOS X',
            'browser' => '*safari',
            'host' => 'my.macosx.box',
            'port' => 4444,
            'timeout' => 30000,
        ),
        array(
            'name' => 'Safari on Windows XP',
            'browser' => '*custom C:\Program Files\Safari\Safari.exe -url',
            'host' => 'my.windowsxp.box',
            'port' => 4444,
            'timeout' => 30000,
        )
    );
}
    
```



```

    ),
    array(
        'name' => 'Internet Explorer on Windows XP',
        'browser' => '*iexplore',
        'host' => 'my.windowsxp.box',
        'port' => 4444,
        'timeout' => 30000,
    )
);

protected function setUp()
{
    $this->setBrowserUrl('http://www.example.com/');
}

public function testTitle()
{
    $this->open('http://www.example.com/');
    $this->assertTitle('Example Web Page');
}
}
?>

```

- PHPUnit\_Extensions\_SeleniumTestCase Selenium coverage
1. PHPUnit/Extensions/SeleniumCommon/phpunit\_coverage.php document root directory
  2. php.ini , PHPUnit/Extensions/SeleniumCommon/prepend.php  
 PHPUnit/Extensions/SeleniumCommon/append.php auto\_prepend\_file auto\_append\_file
  3. PHPUnit\_Extensions\_SeleniumTestCase test case ,

```
protected $coverageScriptUrl = 'http://host/phpunit_coverage.php';
```

phpunit\_coverage.php URL

17.2 " " PHPUnit\_Extensions\_SeleniumTestCase 가

Assertion			
void assertElementValueEquals(string \$locator, string \$text)	\$locator	element	\$text
void assertElementValueNotEquals(string \$locator, string \$text)	\$locator	element	\$text
void assertElementValueContains(string \$locator, string \$text)	\$locator	element	\$text
void assertElementValueNotContains(string \$locator, string \$text)	\$locator	element	\$text
void assertElementContainsText(string \$locator, string \$text)	\$locator	element 가	\$text

void assertElementNotContainsText(string \$locator, string \$text)	\$locator element 가 \$text
void assertSelectHasOption(string \$selectLocator, string \$option)	가
void assertSelectNotHasOption(string \$selectLocator, string \$option)	가
void assertSelected(\$selectLocator, \$option)	label
void assertNotSelected(\$selectLocator, \$option)	label
void assertIsSelected(string \$selectLocator, string \$value)	
void assertIsNotSelected(string \$selectLocator, string \$value)	
17.2 (Assertions)	

17.3 "templete" " PHPUnit\_Extensions\_SeleniumTestCase templete

void defaultAsserti ons()	Test case 가 Override Selenium Server
17.3 Templete	

가 , Selenium [1]

Selenium 1 \_\_call

PHPUnit\_Extensions\_SeleniumTestCase\_Driver::\_\_call() API [2] , PHP

가 type

runSelenese(\$filename) , Selenese/HTML Selenium

, \$seleneseDirectory , Selenese/HTML

.htm . Selenese/HTML . 17.5 "Selenium/HTML"

" 가

17.5 Selenium/HTML

```
<?php
require_once 'PHPUnit/Extensions/SeleniumTestCase.php';

class SeleneseTests extends PHPUnit_Extensions_SeleniumTestCase
{
    public static $seleneseDirectory = '/path/to/files';
}
?>
```

Selenium 1.1.1 가 가 session session

. Session , bootstrap

PHPUnit\_Extensions\_SeleniumTestCase::shareSession(true) (

) session . Cookie , (tearDown())

가

Notes

# PHPUnitManual:18

---

18 Log

PHPUnit log .

Notes

# PHPUnitManual:18.1

---

18.1 (XML)

PHPUnit XML log , Apache Ant JUnit 가  
 . ArrayTest 가 XML log .

```
<?xml version="1.0" encoding="UTF-8"?>
<testsuites>
  <testsuite name="ArrayTest"
    file="/home/sb/ArrayTest.php"
    tests="2"
    assertions="2"
    failures="0"
    errors="0"
    time="0.016030">
    <testcase name="testNewArrayIsEmpty"
      class="ArrayTest"
      file="/home/sb/ArrayTest.php"
      line="6"
      assertions="1"
      time="0.008044"/>
    <testcase name="testArrayContainsAnElement"
      class="ArrayTest"
      file="/home/sb/ArrayTest.php"
      line="15"
      assertions="1"
      time="0.007986"/>
  </testsuite>
</testsuites>
```

XML log , FailureErrorTest testFailure testError 가  
 가 .

```
<?xml version="1.0" encoding="UTF-8"?>
<testsuites>
  <testsuite name="FailureErrorTest"
    file="/home/sb/FailureErrorTest.php"
```

```
    tests="2"
    assertions="1"
    failures="1"
    errors="1"
    time="0.019744">
<testcase name="testFailure"
  class="FailureErrorTest"
  file="/home/sb/FailureErrorTest.php"
  line="6"
  assertions="1"
  time="0.011456">
<failure type="PHPUnit_Framework_ExpectationFailedException">
  testFailure(FailureErrorTest)
  Failed asserting that &lt;integer:2&gt; matches expected value
&lt;integer:1&gt;.

  /home/sb/FailureErrorTest.php:8
</failure>
</testcase>
<testcase name="testError"
  class="FailureErrorTest"
  file="/home/sb/FailureErrorTest.php"
  line="11"
  assertions="0"
  time="0.008288">
<error type="Exception">testError(FailureErrorTest)
Exception:

  /home/sb/FailureErrorTest.php:13
</error>
</testcase>
</testsuite>
</testsuites>
```

## Notes

# PHPUnitManual:18.2

---

18.2 (TAP)

Test Anything Protocol (TAP)[1] Perl  
 ArrayTest 가 TAP log .

```
TAP version 13
ok 1 - testNewArrayIsEmpty (ArrayTest)
ok 2 - testArrayContainsAnElement (ArrayTest)
1..2
```

TAP log , FailureErrorTest testFailure testError 가 .,  
 of a test case class named 가 .

```
TAP version 13
not ok 1 - Failure: testFailure(FailureErrorTest)
---
message: 'Failed asserting that <integer:2> matches expected value <integer:1>.'
severity: fail
data:
  got: 2
  expected: 1
...
not ok 2 - Error: testError(FailureErrorTest)
1..2
```

## Notes

# PHPUnitManual:18.3

18.3 (JSON)

JavaScript Object Notation (JSON)[2]

ArrayTest 가

JSON

```
{ "event": "suiteStart", "suite": "ArrayTest", "tests": 2 }
{ "event": "test", "suite": "ArrayTest",
  "test": "testNewArrayIsEmpty (ArrayTest)", "status": "pass",
  "time": 0.000460147858, "trace": [], "message": "" }
{ "event": "test", "suite": "ArrayTest",
  "test": "testArrayContainsAnElement (ArrayTest)", "status": "pass",
  "time": 0.000422954559, "trace": [], "message": "" }
```

JSON

FailureErrorTest

testFailure

testError 가

가

```
{ "event": "suiteStart", "suite": "FailureErrorTest", "tests": 2 }
{ "event": "test", "suite": "FailureErrorTest",
  "test": "testFailure (FailureErrorTest)", "status": "fail",
  "time": 0.0082459449768066, "trace": [],
  "message": "Failed asserting that <integer:2> is equal to <integer:1>." }
{ "event": "test", "suite": "FailureErrorTest",
  "test": "testError (FailureErrorTest)", "status": "error",
  "time": 0.0083680152893066, "trace": [], "message": "" }
```

## Notes

# PHPUnitManual:18.4

## 18.4 Code Coverage (XML)

PHPUnit code coverage log XML , Clover .  
 BankAccountTest 가 XML log .

```
<?xml version="1.0" encoding="UTF-8"?>
<coverage generated="1184835473" phpunit="3.6.0">
  <project name="BankAccountTest" timestamp="1184835473">
    <file name="/home/sb/BankAccount.php">
      <class name="BankAccountException">
        <metrics methods="0" coveredmethods="0" statements="0"
          coveredstatements="0" elements="0" coveredelements="0"/>
      </class>
      <class name="BankAccount">
        <metrics methods="4" coveredmethods="4" statements="13"
          coveredstatements="5" elements="17" coveredelements="9"/>
      </class>
      <line num="77" type="method" count="3"/>
      <line num="79" type="stmt" count="3"/>
      <line num="89" type="method" count="2"/>
      <line num="91" type="stmt" count="2"/>
      <line num="92" type="stmt" count="0"/>
      <line num="93" type="stmt" count="0"/>
      <line num="94" type="stmt" count="2"/>
      <line num="96" type="stmt" count="0"/>
      <line num="105" type="method" count="1"/>
      <line num="107" type="stmt" count="1"/>
      <line num="109" type="stmt" count="0"/>
      <line num="119" type="method" count="1"/>
      <line num="121" type="stmt" count="1"/>
      <line num="123" type="stmt" count="0"/>
      <metrics loc="126" ncloc="37" classes="2" methods="4" coveredmethods="4"
        statements="13" coveredstatements="5" elements="17"
        coveredelements="9"/>
    </file>
    <metrics files="1" loc="126" ncloc="37" classes="2" methods="4"
      coveredmethods="4" statements="13" coveredstatements="5"
      elements="17" coveredelements="9"/>
  </project>
</coverage>
```

### Notes

# PHPUnitManual:18.5

## 18.5 Code Coverage (TEXT)

```

code coverage --command-line text --filter --colors
coverage --filter --colors
Command-line php://stdout --colors
Command-line
coverage --xml showUncoveredFiles
"Log "

```

Figure 18.1 Command-line code coverage

```

Code Coverage Report for "BankAccount"
2011-10-21 13:12:17

Summary:
Classes: 87.50% (21/24)
Methods: 78.95% (30/38)
Lines: 90.86% (169/186)

@bankaccount.controller::BankAccountController
Methods: 100.00% ( 2/ 2) Lines: 100.00% ( 13/ 13)
@bankaccount.controller::BankAccountListController
Methods: 100.00% ( 1/ 1) Lines: 100.00% ( 2/ 2)
@bankaccount.framework::ControllerException
Methods: 100.00% ( 0/ 0) Lines: 100.00% ( 0/ 0)
@bankaccount.framework::ControllerFactory
Methods: 100.00% ( 2/ 2) Lines: 100.00% ( 13/ 13)
@bankaccount.framework::FrontController
Methods: 100.00% ( 2/ 2) Lines: 100.00% ( 13/ 13)
@bankaccount.framework::HashMap
Methods: 100.00% ( 2/ 2) Lines: 100.00% ( 5/ 5)
@bankaccount.framework::IdentityMap
Methods: 0.00% ( 0/ 6) Lines: 0.00% ( 0/ 13)

```

### Notes



# PHPUnitManual:19

---

19 PHPUnit

, PHPUnit  
PHPUnit

## Notes

# PHPUnitManual:19.1

---

19.1 PHPUnit\_Framework\_TestCase Subclass

PHPUnit\_Framework\_TestCase subclass PHPUnit 가

## Notes

# PHPUnitManual:19.2

---

19.2 (custom assertions)

, PHPUnit . 19.1 "  
PHPUnit\_Framework\_Assert assertTrue() assertTrue() " , assertTrue()  
assertTrue() assertTrue() wrapper . assertTrue() 가 matcher ,  
assertThat() 가 가 .  
19.1 PHPUnit\_Framework\_Assert assertTrue() assertTrue()

```
<?php
abstract class PHPUnit_Framework_Assert
{
    // ...

    /**
     * Asserts that a condition is true.
     *
     * @param boolean $condition
     * @param string $message
     * @throws PHPUnit_Framework_AssertionFailedError
     */
    public static function assertTrue($condition, $message = '')
    {
        self::assertThat($condition, self::assertTrue(), $message);
    }

    // ...
}
```

```

/**
 * Returns a PHPUnit_Framework_Constraint_IsTrue matcher object.
 *
 * @return PHPUnit_Framework_Constraint_IsTrue
 * @since Method available since Release 3.3.0
 */
public static function assertTrue()
{
    return new PHPUnit_Framework_Constraint_IsTrue;
}

// ...
}??>

```

19.2 "PHPUnit\_Framework\_Constraint\_IsTrue" , PHPUnit\_Framework\_Constraint\_IsTrue가 matcher  
 ( (constraints)) PHPUnit\_Framework\_Constraint

.

19.2 PHPUnit\_Framework\_Constraint\_IsTrue

```

<?php
class PHPUnit_Framework_Constraint_IsTrue extends
PHPUnit_Framework_Constraint
{
    /**
     * Evaluates the constraint for parameter $other. Returns TRUE if
the
     * constraint is met, FALSE otherwise.
     *
     * @param mixed $other Value or object to evaluate.
     * @return bool
     */
    public function matches($other)
    {
        return $other === TRUE;
    }

    /**
     * Returns a string representation of the constraint.
     *
     * @return string
     */
    public function toString()
    {
        return 'is true';
    }
}??>

```

assertTrue() isTrue() PHPUnit\_Framework\_Constraint\_IsTrue ,  
가 ( ) assertThat() . ,  
mock matcher isTrue() .

**Notes**

# PHPUnitManual:19.3

---

## 19.3 PHPUnit\_Framework\_TestListener

19.3 "simple test listener" , PHPUnit\_Framework\_TestListener .

19.3: simple test listener

```
<?php
class SimpleTestListener implements PHPUnit_Framework_TestListener
{
    public function addError(PHPUnit_Framework_Test $test, Exception
    $e, $time)
    {
        printf("Error while running test '%s'.\n", $test->getName());
    }

    public function addFailure(PHPUnit_Framework_Test $test,
    PHPUnit_Framework_AssertionFailedError $e, $time)
    {
        printf("Test '%s' failed.\n", $test->getName());
    }

    public function addIncompleteTest(PHPUnit_Framework_Test $test,
    Exception $e, $time)
    {
        printf("Test '%s' is incomplete.\n", $test->getName());
    }

    public function addSkippedTest(PHPUnit_Framework_Test $test,
    Exception $e, $time)
    {
        printf("Test '%s' has been skipped.\n", $test->getName());
    }

    public function startTest(PHPUnit_Framework_Test $test)
    {
        printf("Test '%s' started.\n", $test->getName());
    }

    public function endTest(PHPUnit_Framework_Test $test, $time)
    {

```

```

        printf("Test '%s' ended.\n", $test->getName());
    }

    public function startTestSuite(PHPUnit_Framework_TestSuite $suite)
    {
        printf("TestSuite '%s' started.\n", $suite->getName());
    }

    public function endTestSuite(PHPUnit_Framework_TestSuite $suite)
    {
        printf("TestSuite '%s' ended.\n", $suite->getName());
    }
}
?>

```

"Test Listeners"      test listener      attach      PHPUnit

**Notes**

# PHPUnitManual:19.4

19.4 PHPUnit\_Extensions\_TestDecorator      subclass

PHPUnit\_Extensions\_TestDecorator      subclass      test case      test suite      wrapping      Decorator design pattern

PHPUnit      , PHPUnit\_Extensions\_RepeatedTest      PHPUnit\_Extensions\_TestSetup      2가      (concrete) decorator 가      . PHPUnit\_Extensions\_RepeatedTest 가      . PHPUnit\_Extensions\_TestSetup      6      Fixture

19.4 "RepeatedTest Decorator"      , test decorator PHPUnit\_Extensions\_RepeatedTest      .      test decorators      .

19.4 RepeatedTest Decorator

```

<?php
require_once 'PHPUnit/Extensions/TestDecorator.php';

class PHPUnit_Extensions_RepeatedTest extends
PHPUnit_Extensions_TestDecorator
{
    private $timesRepeat = 1;

    public function __construct(PHPUnit_Framework_Test $test,
$timesRepeat = 1)
    {
        parent::__construct($test);
    }
}

```

```
        if (is_integer($timesRepeat) &&
            $timesRepeat >= 0) {
            $this->timesRepeat = $timesRepeat;
        }
    }

    public function count()
    {
        return $this->timesRepeat * $this->test->count();
    }

    public function run(PHPUnit_Framework_TestResult $result = NULL)
    {
        if ($result === NULL) {
            $result = $this->createResult();
        }

        for ($i = 0; $i < $this->timesRepeat && !$result->shouldStop(); $i++) {
            $this->test->run($result);
        }

        return $result;
    }
}
?>
```

## Notes

# PHPUnitManual:19.5

## 19.5 PHPUnit\_Framework\_Test

PHPUnit\_Framework\_Test

PHPUnit\_Framework\_Test

PHPUnit\_Framework\_TestCase

PHPUnit\_Framework\_Test

data-driven

CSV

data-driven

19.5 "data-driven

"

foo;bar

( : CSV

...),

,2

### 19.5 data-driven

```

<?php
class DataDrivenTest implements PHPUnit_Framework_Test
{
    private $lines;

    public function __construct($dataFile)
    {
        $this->lines = file($dataFile);
    }

    public function count()
    {
        return 1;
    }

    public function run(PHPUnit_Framework_TestResult $result = NULL)
    {
        if ($result === NULL) {
            $result = new PHPUnit_Framework_TestResult;
        }

        foreach ($this->lines as $line) {
            $result->startTest($this);
            PHP_Timer::start();
            $stopTime = NULL;

            list($expected, $actual) = explode(';', $line);

            try {
                PHPUnit_Framework_Assert::assertEquals(
                    trim($expected), trim($actual)
                );
            }

            catch (PHPUnit_Framework_AssertionFailedError $e) {
                $stopTime = PHP_Timer::stop();
            }
        }
    }
}

```

```
        $result->addFailure($this, $e, $stopTime);
    }

    catch (Exception $e) {
        $stopTime = PHP_Timer::stop();
        $result->addError($this, $e, $stopTime);
    }

    if ($stopTime === NULL) {
        $stopTime = PHP_Timer::stop();
    }

    $result->endTest($this, $stopTime);
}

return $result;
}
}

$test = new DataDrivenTest('data_file.csv');
$result = PHPUnit_TextUI_TestRunner::run($test);
?>
```

PHPUnit 3.7.0 by Sebastian Bergmann.

.F

Time: 0 seconds

There was 1 failure:

```
1) DataDrivenTest
Failed asserting that two strings are equal.
expected string <bar>
difference      < x>
got string      <baz>
/home/sb/DataDrivenTest.php:32
/home/sb/DataDrivenTest.php:53
```

FAILURES!

Tests: 2, Failures: 1.

## Notes

# PHPUnitManual:A

## Appendix A. Assertions

### A.1 " (Assertions)"

Assertion
<code>assertArrayHasKey(\$key, \$array, \$message = )</code>
<code>assertArrayNotHasKey(\$key, \$array, \$message = )</code>
<code>assertAttributeContains(\$needle, \$haystackAttributeName, \$haystackClassOrObject, \$message = , \$ignoreCase = FALSE, \$checkForObjectIdentity = TRUE)</code>
<code>assertAttributeContainsOnly(\$type, \$haystackAttributeName, \$haystackClassOrObject, \$isNativeType = NULL, \$message = )</code>
<code>assertAttributeCount(\$expectedCount, \$haystackAttributeName, \$haystackClassOrObject, \$message = )</code>
<code>assertAttributeEmpty(\$haystackAttributeName, \$haystackClassOrObject, \$message = )</code>
<code>assertAttributeEquals(\$expected, \$actualAttributeName, \$actualClassOrObject, \$message = , \$delta = 0, \$maxDepth = 10, \$canonicalize = FALSE, \$ignoreCase = FALSE)</code>
<code>assertAttributeGreaterThan(\$expected, \$actualAttributeName, \$actualClassOrObject, \$message = )</code>
<code>assertAttributeGreaterThanOrEqual(\$expected, \$actualAttributeName, \$actualClassOrObject, \$message = )</code>
<code>assertAttributeInstanceOf(\$expected, \$attributeName, \$classOrObject, \$message = )</code>
<code>assertAttributeInternalType(\$expected, \$attributeName, \$classOrObject, \$message = )</code>
<code>assertAttributeLessThan(\$expected, \$actualAttributeName, \$actualClassOrObject, \$message = )</code>
<code>assertAttributeLessThanOrEqual(\$expected, \$actualAttributeName, \$actualClassOrObject, \$message = )</code>
<code>assertAttributeNotContains(\$needle, \$haystackAttributeName, \$haystackClassOrObject, \$message = , \$ignoreCase = FALSE, \$checkForObjectIdentity = TRUE)</code>
<code>assertAttributeNotContainsOnly(\$type, \$haystackAttributeName, \$haystackClassOrObject, \$isNativeType = NULL, \$message = )</code>
<code>assertAttributeNotCount(\$expectedCount, \$haystackAttributeName, \$haystackClassOrObject, \$message = )</code>
<code>assertAttributeNotEmpty(\$haystackAttributeName, \$haystackClassOrObject, \$message = )</code>
<code>assertAttributeNotEquals(\$expected, \$actualAttributeName, \$actualClassOrObject, \$message = , \$delta = 0, \$maxDepth = 10, \$canonicalize = FALSE, \$ignoreCase = FALSE)</code>
<code>assertAttributeNotInstanceOf(\$expected, \$attributeName, \$classOrObject, \$message = )</code>
<code>assertAttributeNotInternalType(\$expected, \$attributeName, \$classOrObject, \$message = )</code>
<code>assertAttributeNotSame(\$expected, \$actualAttributeName, \$actualClassOrObject, \$message = )</code>
<code>assertAttributeSame(\$expected, \$actualAttributeName, \$actualClassOrObject, \$message = )</code>
<code>assertClassHasAttribute(\$attributeName, \$className, \$message = )</code>
<code>assertClassHasStaticAttribute(\$attributeName, \$className, \$message = )</code>
<code>assertClassNotHasAttribute(\$attributeName, \$className, \$message = )</code>
<code>assertClassNotHasStaticAttribute(\$attributeName, \$className, \$message = )</code>
<code>assertContains(\$needle, \$haystack, \$message = , \$ignoreCase = FALSE, \$checkForObjectIdentity = TRUE)</code>
<code>assertContainsOnly(\$type, \$haystack, \$isNativeType = NULL, \$message = )</code>
<code>assertContainsOnlyInstancesOf(\$classname, \$haystack, \$message = )</code>
<code>assertCount(\$expectedCount, \$haystack, \$message = )</code>
<code>assertEmpty(\$actual, \$message = )</code>



<code>assertEqualXMLStructure(DOMElement \$expectedElement, DOMElement \$actualElement, \$checkAttributes = FALSE, \$message = )</code>
<code>assertEquals(\$expected, \$actual, \$message = , \$delta = 0, \$maxDepth = 10, \$canonicalize = FALSE, \$ignoreCase = FALSE)</code>
<code>assertFalse(\$condition, \$message = )</code>
<code>assertFileEquals(\$expected, \$actual, \$message = , \$canonicalize = FALSE, \$ignoreCase = FALSE)</code>
<code>assertFileExists(\$filename, \$message = )</code>
<code>assertFileNotEquals(\$expected, \$actual, \$message = , \$canonicalize = FALSE, \$ignoreCase = FALSE)</code>
<code>assertFileNotExists(\$filename, \$message = )</code>
<code>assertGreaterThan(\$expected, \$actual, \$message = )</code>
<code>assertGreaterThanOrEqual(\$expected, \$actual, \$message = )</code>
<code>assertInstanceOf(\$expected, \$actual, \$message = )</code>
<code>assertInternalType(\$expected, \$actual, \$message = )</code>
<code>assertJsonFileEqualsJsonFile(\$expectedFile, \$actualFile, \$message = )</code>
<code>assertJsonFileNotEqualsJsonFile(\$expectedFile, \$actualFile, \$message = )</code>
<code>assertJsonStringEqualsJsonFile(\$expectedFile, \$actualJson, \$message = )</code>
<code>assertJsonStringEqualsJsonString(\$expectedJson, \$actualJson, \$message = )</code>
<code>assertJsonStringNotEqualsJsonFile(\$expectedFile, \$actualJson, \$message = )</code>
<code>assertJsonStringNotEqualsJsonString(\$expectedJson, \$actualJson, \$message = )</code>
<code>assertLessThan(\$expected, \$actual, \$message = )</code>
<code>assertLessThanOrEqual(\$expected, \$actual, \$message = )</code>
<code>assertNotContains(\$needle, \$haystack, \$message = , \$ignoreCase = FALSE, \$checkForObjectIdentity = TRUE)</code>
<code>assertNotContainsOnly(\$type, \$haystack, \$isNativeType = NULL, \$message = )</code>
<code>assertNotCount(\$expectedCount, \$haystack, \$message = )</code>
<code>assertNotEmpty(\$actual, \$message = )</code>
<code>assertNotEquals(\$expected, \$actual, \$message = , \$delta = 0, \$maxDepth = 10, \$canonicalize = FALSE, \$ignoreCase = FALSE)</code>
<code>assertNotInstanceOf(\$expected, \$actual, \$message = )</code>
<code>assertNotInternalType(\$expected, \$actual, \$message = )</code>
<code>assertNotNull(\$actual, \$message = )</code>
<code>assertNotRegExp(\$pattern, \$string, \$message = )</code>
<code>assertNotSame(\$expected, \$actual, \$message = )</code>
<code>assertNotSameSize(\$expected, \$actual, \$message = )</code>
<code>assertNotTag(\$matcher, \$actual, \$message = , \$isHtml = TRUE)</code>
<code>assertNull(\$actual, \$message = )</code>
<code>assertObjectHasAttribute(\$attributeName, \$object, \$message = )</code>
<code>assertObjectNotHasAttribute(\$attributeName, \$object, \$message = )</code>
<code>assertRegExp(\$pattern, \$string, \$message = )</code>
<code>assertSame(\$expected, \$actual, \$message = )</code>
<code>assertSameSize(\$expected, \$actual, \$message = )</code>
<code>assertSelectCount(\$selector, \$count, \$actual, \$message = , \$isHtml = TRUE)</code>
<code>assertSelectEquals(\$selector, \$content, \$count, \$actual, \$message = , \$isHtml = TRUE)</code>

<code>assertSelectRegExp(\$selector, \$pattern, \$count, \$actual, \$message = , \$isHtml = TRUE)</code>
<code>assertStringEndsNotWith(\$suffix, \$string, \$message = )</code>
<code>assertStringEndsWith(\$suffix, \$string, \$message = )</code>
<code>assertStringEqualsFile(\$expectedFile, \$actualString, \$message = , \$canonicalize = FALSE, \$ignoreCase = FALSE)</code>
<code>assertStringMatchesFormat(\$format, \$string, \$message = )</code>
<code>assertStringMatchesFormatFile(\$formatFile, \$string, \$message = )</code>
<code>assertStringNotEqualsFile(\$expectedFile, \$actualString, \$message = , \$canonicalize = FALSE, \$ignoreCase = FALSE)</code>
<code>assertStringNotMatchesFormat(\$format, \$string, \$message = )</code>
<code>assertStringNotMatchesFormatFile(\$formatFile, \$string, \$message = )</code>
<code>assertStringStartsNotWith(\$prefix, \$string, \$message = )</code>
<code>assertStringStartsWith(\$prefix, \$string, \$message = )</code>
<code>assertTag(\$matcher, \$actual, \$message = , \$isHtml = TRUE)</code>
<code>assertThat(\$value, PHPUnit_Framework_Constraint \$constraint, \$message = )</code>
<code>assertTrue(\$condition, \$message = )</code>
<code>assertXmlFileEqualsXmlFile(\$expectedFile, \$actualFile, \$message = )</code>
<code>assertXmlFileNotEqualsXmlFile(\$expectedFile, \$actualFile, \$message = )</code>
<code>assertXmlStringEqualsXmlFile(\$expectedFile, \$actualXml, \$message = )</code>
<code>assertXmlStringEqualsXmlString(\$expectedXml, \$actualXml, \$message = )</code>
<code>assertXmlStringNotEqualsXmlFile(\$expectedFile, \$actualXml, \$message = )</code>
<code>assertXmlStringNotEqualsXmlString(\$expectedXml, \$actualXml, \$message = )</code>
A.1 (Assertions)

## Notes

# PHPUnitManual:B

---

## Appendix B. Annotations

### Annotations

PHP `getDocComment()` Reflection API  
 PHP `PHPUnit` Reflection API  
 PHP `PHPUnit` Reflection API

#### @author

@author @group alias

#### @backupGlobals

Test case

```
/**
 * @backupGlobals disabled
 */
class MyTest extends PHPUnit_Framework_TestCase
{
    // ...
}
```

#### @backupGlobals

```
/**
 * @backupGlobals disabled
 */
class MyTest extends PHPUnit_Framework_TestCase
{
    /**
     * @backupGlobals enabled
     */
    public function testThatInteractsWithGlobalVariables()
    {
        // ...
    }
}
```

### @backupStaticAttributes

Test case

```
/**
 * @backupStaticAttributes disabled
 */
class MyTest extends PHPUnit_Framework_TestCase
{
    // ...
}
```

The @backupStaticAttributes

```
/**
 * @backupStaticAttributes disabled
 */
class MyTest extends PHPUnit_Framework_TestCase
{
    /**
     * @backupStaticAttributes enabled
     */
    public function testThatInteractsWithStaticAttributes()
    {
        // ...
    }
}
```

### @codeCoverageIgnore\*

@codeCoverageIgnore, @codeCoverageIgnoreStart @codeCoverageIgnoreEnd ,  
 coverage  
 "Code Blocks"

### @covers

@covers , 가

```
/**
 * @covers BankAccount::getBalance
 */
public function testBalanceIsInitiallyZero()
{
    $this->assertEquals(0, $this->ba->getBalance());
}
```

@covers , coverage 가

B.1 " 가 Cover " @covers

Annotation	
@covers ClassName::methodName	가 cover
@covers ClassName	가 cover
@covers ClassName<extended>	가 cover
@covers ClassName::<public>	가 public cover
@covers ClassName::<protected>	가 protected cover
@covers ClassName::<private>	가 private cover
@covers ClassName::<!public>	가 public cover
@covers ClassName::<!protected>	가 protected cover
@covers ClassName::<!private>	가 private cover
@covers ::functionName	가 cover

B.1. 가 Cover

**@coversNothing**

@coversNothing , test case code coverage  
 (integration testing) . 14.3 " cover  
 " , @covers

**@dataProvider**

provider() ( 4.4 "  
 , @dataProvider  
 "Data Providers"

**@depends**

PHPUnit 가 , fixture , fixture  
 , 가 4.2 "@depends  
 " @depends  
 "(Test Dependencies)"

**@expectedException**

4.7 "@expectedException" 가 throw  
 @expectedException  
 " (Testing Exceptions)"

**@expectedExceptionCode**

@expectedExceptionCode @expectedException , throw  
 가

```
class MyTest extends PHPUnit_Framework_TestCase
{
    /**
     * @expectedException MyException
     * @expectedExceptionCode 20
     */
    public function testExceptionHasErrorCode20()
    {
        throw new MyException('Some Message', 20);
    }
}
```

, 가  
 @expectedExceptionCode "@expectedExceptionCode ClassName::CONST"

```
class MyTest extends PHPUnit_Framework_TestCase
{
    /**
     * @expectedException MyException
     * @expectedExceptionCode MyClass::ERRORCODE
     */
    public function testExceptionHasErrorCode20()
    {
        throw new MyException('Some Message', 20);
    }
}
class MyClass
{
    const ERRORCODE = 20;
}
```

### @expectedExceptionMessage

@expectedExceptionMessage      @expectedExceptionCode

```
class MyTest extends PHPUnit_Framework_TestCase
{
    /**
     * @expectedException      MyException
     * @expectedExceptionMessage Some Message
     */
    public function testExceptionHasRightMessage()
    {
        throw new MyException('Some Message', 20);
    }
}
```

가 . 가

```
class MyTest extends PHPUnit_Framework_TestCase
{
    /**
     * @expectedException      MyException
     * @expectedExceptionMessage broken
     */
    public function testExceptionHasRightMessage()
    {
        $param = "broken";
        throw new MyException('Invalid parameter "'. $param. "'", 20);
    }
}
```

, 가

@expectedExceptionMessage      "@expectedExceptionMessage ClassName::CONST"

the section called "@expectedExceptionCode"

### @group

group      . @group

```
class MyTest extends PHPUnit_Framework_TestCase
{
    /**
     * @group specification
     */
    public function testSomething()
    {
    }
}
```

```

/**
 * @group regresssion
 * @group bug2204
 */
public function testSomethingElse()
{
}
}

```

group , command-line test runner --group  
 --exclude-group . XML , .

### @outputBuffering

@outputBuffering PHP output buffering .

```

/**
 * @outputBuffering enabled
 */
class MyTest extends PHPUnit_Framework_TestCase
{
    // ...
}

```

@outputBuffering . output buffering

```

/**
 * @outputBuffering disabled
 */
class MyTest extends PHPUnit_Framework_TestCase
{
    /**
     * @outputBuffering enabled
     */
    public function testThatPrintsSomething()
    {
        // ...
    }
}

```



### @preserveGlobalState

PHPUnit (state) serialize unserialize  
 serialize 가 가 가 ,  
 @preserveGlobalState

```
class MyTest extends PHPUnit_Framework_TestCase
{
    /**
     * @runInSeparateProcess
     * @preserveGlobalState disabled
     */
    public function testInSeparateProcess()
    {
        // ...
    }
}
```

### @requires

@requires ( :PHP )  
 9.3 "@requires "

### @runTestsInSeparateProcesses

test case , PHP  
 /\*\*  
 \* @runTestsInSeparateProcesses  
 \*/  
 class MyTest extends PHPUnit\_Framework\_TestCase  
 {  
 // ...  
 }  
 " : , PHPUnit  
 serialize , unserialize . serialize 가  
 가 가 ,  
 . "@preserveGlobalState"



# PHPUnitManual:C

---

Appendix C. XML

Appendix C. XML

## Notes

# PHPUnitManual:C.1

---

Appendix C.1 PHPUnit

PHPUnit

<phpunit> element

PHPUnit

```

<phpunit backupGlobals="true"
 backupStaticAttributes="false"
 <!--bootstrap="/path/to/bootstrap.php"-->
 cacheTokens="false"
 colors="false"
 convertErrorsToExceptions="true"
 convertNoticesToExceptions="true"
 convertWarningsToExceptions="true"
 forceCoversAnnotation="false"
 mapTestClassNameToCoveredClassName="false"
 printerClass="PHPUnit_TextUI_ResultPrinter"
 <!--printerFile="/path/to/ResultPrinter.php"-->
 processIsolation="false"
 stopOnError="false"
 stopOnFailure="false"
 stopOnIncomplete="false"
 stopOnSkipped="false"
 testSuiteLoaderClass="PHPUnit_Runner_StandardTestSuiteLoader"
 <!--testSuiteLoaderFile="/path/to/StandardTestSuiteLoader.php"-->
 strict="false"
 verbose="false">

<!-- ... -->
</phpunit>
    
```

XML , TextUI test runner

“Command-Line switches”

command-line

convertErrorsToExceptions	false, PHP handler
convertNoticesToExceptions	false, convertErrorsToExceptions handler 가 E_NOTICE, E_USER_NOTICE, E_STRICT
convertWarningsToExceptions	false, convertErrorsToExceptions handler 가 E_WARNING E_USER_WARNING
forceCoversAnnotation	Code Coverage, @covers . @covers "@covers"

## Notes

# PHPUnitManual:C.2

### Appendix C.2 Test Suites

<testsuites> element element <testsuite> , test suite test case  
test suite

```
<testsuites>
  <testsuite name="My Test Suite">
    <directory>/path/to/*Test.php files</directory>
    <file>/path/to/MyTest.php</file>
    <exclude>/path/to/exclude</exclude>
  </testsuite>
</testsuites>
```

phpVersion phpVersionOperator PHP  
PHP 5.3.0 /path/to/\*Test.php /path/to/MyTest.php 가

```
<testsuites>
  <testsuite name="My Test Suite">
    <directory suffix="Test.php" phpVersion="5.3.0" phpVersionOperator=">=">/path/to/files</directory>
    <file phpVersion="5.3.0" phpVersionOperator=">=">/path/to/MyTest.php</file>
  </testsuite>
</testsuites>
```

phpVersionOperator , >=

## Notes

# PHPUnitManual:C.3

---

## Appendix C.3 Groups

`<groups>` element                      element    `<include>`, `<exclude>`,                      `<group>`                      test suite  
 (                      )                      group                      .

```
<groups>
  <include>
    <group>name</group>
  </include>
  <exclude>
    <group>name</group>
  </exclude>
</groups>
```

XML                      , TextUI test runner                      .

- --group name
- --exclude-group name

## Notes

# PHPUnitManual:C.4

---

## Appendix C.4 Code Coverage                      가 /

`<filter>` element                      element                      , code coverage

```
<filter>
  <blacklist>
    <directory suffix=".php">/path/to/files</directory>
    <file>/path/to/file</file>
    <exclude>
      <directory suffix=".php">/path/to/files</directory>
      <file>/path/to/file</file>
    </exclude>
  </blacklist>
  <whitelist processUncoveredFilesFromWhitelist="true">
    <directory suffix=".php">/path/to/files</directory>
    <file>/path/to/file</file>
    <exclude>
      <directory suffix=".php">/path/to/files</directory>
      <file>/path/to/file</file>
    </exclude>
  </whitelist>
</filter>
```

## Notes

# PHPUnitManual:C.5

---

### Appendix C.5 Log

<logging> element                      element    <log>                      ,                      log                      .

```
<logging>
  <log type="coverage-html" target="/tmp/report" charset="UTF-8"
      highlight="false" lowUpperBound="35" highLowerBound="70"/>
  <log type="coverage-clover" target="/tmp/coverage.xml"/>
  <log type="coverage-php" target="/tmp/coverage.serialized"/>
  <log type="coverage-text" target="php://stdout" showUncoveredFiles="false"/>
  <log type="json" target="/tmp/logfile.json"/>
  <log type="tap" target="/tmp/logfile.tap"/>
  <log type="junit" target="/tmp/logfile.xml" logIncompleteSkipped="false"/>
  <log type="testdox-html" target="/tmp/testdox.html"/>
  <log type="testdox-text" target="/tmp/testdox.txt"/>
</logging>
```

XML                      TextUI test runner                      .

- --coverage-html /tmp/report
- --coverage-clover /tmp/coverage.xml
- --coverage-php /tmp/coverage.serialized
- --coverage-text
- --log-json /tmp/logfile.json
- > /tmp/logfile.txt
- --log-tap /tmp/logfile.tap
- --log-junit /tmp/logfile.xml
- --testdox-html /tmp/testdox.html
- --testdox-text /tmp/testdox.txt

charset, highlight, lowUpperBound, highLowerBound, logIncompleteSkipped                      showUncoveredFiles

TextUI                      가                      .

- charset:                      html                      Character set
- highlight: true                      , code coverage                      (syntax highlighted)                      .
- lowUpperBound: cover                      , "                      "                      .
- highLowerBound: cover                      , "                      "                      .
- showUncoveredFiles: --coverage-text                      coverage

## Notes

# PHPUnitManual:C.6

---

## Appendix C.6 Test Listeners

<listeners> element element <listener> , test listener attach

```
<listeners>
  <listener class="MyListener" file="/optional/path/to/MyListener.php">
    <arguments>
      <array>
        <element key="0">
          <string>Sebastian</string>
        </element>
      </array>
      <integer>22</integer>
      <string>April</string>
      <double>19.78</double>
      <null/>
      <object class="stdClass"/>
    </arguments>
  </listener>
</listeners>
```

XML \$listener ( ) .

```
$listener = new MyListener(
    array('Sebastian'),
    22,
    'April',
    19.78,
    NULL,
    new stdClass
);
```

## Notes

# PHPUnitManual:C.7

Appendix C.7 PHP INI

<php> element                    element                    , PHP setting                    ,  
 include\_path                    가                    .

```
<php>
  <includePath>.</includePath>
  <ini name="foo" value="bar"/>
  <const name="foo" value="bar"/>
  <var name="foo" value="bar"/>
  <env name="foo" value="bar"/>
  <post name="foo" value="bar"/>
  <get name="foo" value="bar"/>
  <cookie name="foo" value="bar"/>
  <server name="foo" value="bar"/>
  <files name="foo" value="bar"/>
  <request name="foo" value="bar"/>
</php>
```

XML                    PHP

```
ini_set('foo', 'bar');
define('foo', 'bar');
$GLOBALS['foo'] = 'bar';
$_ENV['foo'] = 'bar';
$_POST['foo'] = 'bar';
$_GET['foo'] = 'bar';
$_COOKIE['foo'] = 'bar';
$_SERVER['foo'] = 'bar';
$_FILES['foo'] = 'bar';
$_REQUEST['foo'] = 'bar';
```

## Notes



# PHPUnitManual:C.8

---

## Appendix C.8 Selenium RC

<selenium> element                    element <browser>                    , Selenium RC                    .

```
<selenium>
  <browser name="Firefox on Linux"
    browser="*firefox /usr/lib/firefox/firefox-bin"
    host="my.linux.box"
    port="4444"
    timeout="30000"/>
</selenium>
```

XML

PHP

```
class WebTest extends PHPUnit_Extensions_SeleniumTestCase
{
    public static $browsers = array(
        array(
            'name'      => 'Firefox on Linux',
            'browser'   => '*firefox /usr/lib/firefox/firefox-bin',
            'host'      => 'my.linux.box',
            'port'      => 4444,
            'timeout'   => 30000
        )
    );

    // ...
}
```

## Notes

# PHPUnitManual:D

---

Appendix D. Index

## Index

### Symbols

\$backupGlobalsBlacklist, 86

\$backupStaticAttributesBlacklist, 86

@assert, 148

@author,, 173

@backupGlobals, 86, 173, 173

@backupStaticAttributes, 86, 173, 174

@codeCoverageIgnore, 143, 174

@codeCoverageIgnoreEnd, 143, 174

@codeCoverageIgnoreStart, 143, 174

@covers, 141, 174

@coversNothing, 142, 175

@dataProvider, 12, 15, 15, 15, 175

@depends, 10, 15, 15, 15, 175

@expectedException, 15, 16, 176

@expectedExceptionCode, 16, 176

@expectedExceptionMessage, 16, 176

@group,,,, 177

@outputBuffering, 177, 178

@preserveGlobalState, 178

@requires, 178, 178

@runInSeparateProcess, 179

@runTestsInSeparateProcesses, 178

@test,, 179

@testdox, 179

@ticket, 179

---

**A**

Agile Documentation,,, 145  
Annotation, 10,, 10, 12, 15, 15, 15, 15, 16,  
anything(),  
arrayHasKey(),  
assertArrayHasKey(), 22,  
assertArrayNotHasKey(), 22,  
assertAttributeContains(), 24,  
assertAttributeContainsOnly(), 26,  
assertAttributeCount(),  
assertAttributeEmpty(), 29,  
assertAttributeEquals(), 32,  
assertAttributeGreaterThan(), 41,  
assertAttributeGreaterThanOrEqual(), 42,  
assertAttributeInstanceOf(), 43,  
assertAttributeInternalType(), 44,  
assertAttributeLessThan(), 48,  
assertAttributeLessThanOrEqual(), 48,  
assertAttributeNotContains(), 24,  
assertAttributeNotContainsOnly(), 26,  
assertAttributeNotCount(),  
assertAttributeNotEmpty(), 29,  
assertAttributeNotEquals(), 32,  
assertAttributeNotInstanceOf(), 43,  
assertAttributeNotInternalType(), 44,  
assertAttributeNotSame(), 54,  
assertAttributeSame(), 54,  
assertClassHasAttribute(), 23,  
assertClassHasStaticAttribute(), 24,  
assertClassNotHasAttribute(), 23,  
assertClassNotHasStaticAttribute(), 24,  
assertContains(), 24,  
assertContainsOnly(), 26,  
assertContainsOnlyInstancesOf(), 27,  
assertCount(), 28,  
assertEmpty(), 29,  
assertEquals(), 32,  
assertEqualXMLStructure(), 30,  
assertFalse(), 38,

---

---

- assertFileEquals(), 39,
- assertFileExists(), 40,
- assertFileNotEquals(), 39,
- assertFileNotExists(), 40,
- assertGreaterThan(), 41,
- assertGreaterThanOrEqual(), 42,
- assertInstanceOf(), 43,
- assertInternalType(), 44,
- Assertions, 2
- assertJsonFileEqualsJsonFile(), 45,
- assertJsonFileNotEqualsJsonFile(), 45,
- assertJsonStringEqualsJsonFile(), 46,
- assertJsonStringEqualsJsonString(), 46,
- assertJsonStringNotEqualsJsonFile(), 46,
- assertJsonStringNotEqualsJsonString(), 46,
- assertLessThan(), 48,
- assertLessThanOrEqual(), 48,
- assertNotContains(), 24,
- assertNotContainsOnly(), 26,
- assertNotCount(), 28,
- assertNotEmpty(), 29,
- assertNotEquals(), 32,
- assertNotInstanceOf(), 43,
- assertNotInternalType(), 44,
- assertNotNull(), 49,
- assertNotRegExp(), 51,
- assertNotSame(), 54,
- assertNotSameSize(),
- assertNotTag(), 64,
- assertNull(), 49,
- assertObjectHasAttribute(), 50,
- assertObjectNotHasAttribute(), 50,
- assertPostConditions(), 83
- assertPreConditions(), 83
- assertRegExp(), 51,
- assertSame(), 54,
- assertSameSize(),
- assertSelectCount(), 56,
- assertSelectEquals(), 58,

---

assertSelectRegExp(), 59,  
assertStringEndsNotWith(), 61,  
assertStringEndsWith(), 61,  
assertStringEqualsFile(), 62,  
assertStringMatchesFormat(), 52,  
assertStringMatchesFormatFile(), 53,  
assertStringNotEqualsFile(), 62,  
assertStringNotMatchesFormat(), 52,  
assertStringNotMatchesFormatFile(), 53,  
assertStringStartsNotWith(), 63,  
assertStringStartsWith(), 63,  
assertTag(), 64,  
assertThat(), 66,  
assertTrue(), 68,  
assertXmlFileEqualsXmlFile(), 69,  
assertXmlFileNotEqualsXmlFile(), 69,  
assertXmlStringEqualsXmlFile(), 70,  
assertXmlStringEqualsXmlString(), 71,  
assertXmlStringNotEqualsXmlFile(), 70,  
assertXmlStringNotEqualsXmlString(), 71,  
attribute(),  
attributeEqualTo(),  
Automated Documentation, 145  
Automated Test, 2

## **B**

Behaviour-Driven Development, 133  
Blacklist, 143, 182

## **C**

classHasAttribute(),  
classHasStaticAttribute(),  
Code Coverage,,, 139, 143, 174, 182  
Configuration,,  
Constant, 183  
contains(),  
containsOnly(),  
containsOnlyInstancesOf(),

---

**D**

Data-Driven Tests, 167  
Defect Localization, 11  
Depended-On Component, 112  
Design-by-Contract, 128  
Documenting Assumptions, 145  
Domain-Driven Design, 133

**E**

equalTo(),  
Error, 77  
Error Handler, 19  
expects(), 113, 114, 114, 115, 115, 116, 116, 117  
Extreme Programming, 128, 133, 145

**F**

Failure, 77  
fileExists(),  
Fixture, 82  
Fluent Interface, 112

**G**

getMock(), 113, 114, 115, 115, 116, 116, 117  
getMockBuilder(), 114  
getMockForAbstractClass(), 120  
getMockFromWSDL(), 121  
Global Variable, 86, 183  
greaterThan(),  
greaterThanOrEqual(),

**H**

hasAttribute(),

**I**

identicalTo(),  
include\_path,  
Incomplete Test, 108, 147  
isFalse(),  
isInstanceOf(),  
isNull(),  
isTrue(),

---

isType(),

## J

JSON,

## L

lessThan(),

lessThanOrEqual(),

Logfile,,

Logging, 160, 182

logicalAnd(),

logicalNot(),

logicalOr(),

logicalXor(),

## M

matchesRegularExpression(),

method(), 113, 114, 114, 115, 115, 116, 116, 117

Mock Object, 117, 119

## O

onConsecutiveCalls(), 116

onNotSuccessfulTest(), 83

## P

PHP Error, 19

PHP Notice, 19

PHP Warning, 19

php.ini, 183

PHPUnit\_Extensions\_RepeatedTest, 166

PHPUnit\_Extensions\_Selenium2TestCase, 152

PHPUnit\_Extensions\_SeleniumTestCase, 153

PHPUnit\_Extensions\_Story\_TestCase, 133

PHPUnit\_Extensions\_TestDecorator, 166

PHPUnit\_Extensions\_TestSetup, 166

PHPUnit\_Framework\_Assert, 131

PHPUnit\_Framework\_Error, 19

PHPUnit\_Framework\_Error\_Notice, 20

PHPUnit\_Framework\_Error\_Warning, 20

PHPUnit\_Framework\_IncompleteTest, 108

PHPUnit\_Framework\_IncompleteTestError, 108

PHPUnit\_Framework\_Test, 167  
PHPUnit\_Framework\_TestCase, 10, 164  
PHPUnit\_Framework\_TestListener,, 165, 183  
PHPUnit\_Runner\_TestSuiteLoader,  
PHPUnit\_Util\_Printer,  
Process Isolation,

## **R**

Refactoring, 126  
Report,  
returnArgument(), 114  
returnCallback(), 116  
returnSelf(), 115  
returnValue(), 113, 114  
returnValueMap(), 115

## **S**

Selenium RC, 184  
Selenium Server, 152  
setUp(), 82, 83, 83  
setUpBeforeClass, 85  
setUpBeforeClass(), 83, 83  
Skeleton Generator, 147  
stringContains(),  
stringEndsWith(),  
stringStartsWith(),  
Stub, 112  
Stubs, 146  
System Under Test, 112

## **T**

tearDown(), 82, 83, 83  
tearDownAfterClass, 85  
tearDownAfterClass(), 83, 83  
Template Method, 82, 83, 83, 83  
Test Dependencies, 10  
Test Double, 112  
Test Groups,,,, 181  
Test Isolation,,,, 86  
Test Listener, 183

---



Test Suite, 87, 181  
Test-Driven Development, 128, 133  
Test-First Programming, 128  
TestDox, 145, 179  
throwException(), 117

## U

Unit Test, 1, 128

## W

Whitelist, 143, 182  
will(), 113, 114, 114, 115, 115, 116, 116, 117

## X

Xdebug, 139  
XML Configuration, 88

# PHPUnitManual:E

---

Appendix E.

[Astels2003] Test Driven Development. David Astels. Copyright © 2003. Prentice Hall. ISBN 0131016490.

[Astels2006] A New Look at Test-Driven Development. David Astels. Copyright © 2006. [http://blog.daveastels.com/files/BDD\\_Intro.pdf](http://blog.daveastels.com/files/BDD_Intro.pdf).

[Beck2002] Test Driven Development by Example. Kent Beck. Copyright © 2002. Addison-Wesley. ISBN 0-321-14653-0.

[Meszaros2007] xUnit Test Patterns: Refactoring Test Code. Gerard Meszaros. Copyright © 2007. Addison-Wesley. ISBN 978-0131495050.

# PHPUnitManual:F

---

Appendix F.

Copyright (c) 2005-2012 Sebastian Bergmann.

Creative Commons Attribution License

<http://creativecommons.org/licenses/by/2.0/>

Creative Commons, 559, Nathan Abbott Way, Stanford, California 943.6, USA

---

- - , , ,
- - , ,
- ,
- 가

```
=====
Creative Commons Legal Code
Attribution 3.0 Unported
```

```
CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE
LEGAL SERVICES. DISTRIBUTION OF THIS LICENSE DOES NOT CREATE AN
ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS
INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO
WARRANTIES REGARDING THE INFORMATION PROVIDED, AND DISCLAIMS
LIABILITY FOR DAMAGES RESULTING FROM ITS USE.
```

```
License
```

```
THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS
CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS
PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE
WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW
IS PROHIBITED.
```

```
BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND
AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS
LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU
```

---

THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

- a. "Adaptation" means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.
- b. "Collection" means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined above) for the purposes of this License.
- c. "Distribute" means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership.
- d. "Licensor" means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.
- e. "Original Author" means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or

expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.

- f. "Work" means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.
- g. "You" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.
- h. "Publicly Perform" means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.
- i. "Reproduce" means to make copies of the Work by any means including without limitation by sound or visual recordings and

the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

2. Fair Dealing Rights. Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.
3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:
  - a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;
  - b. to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The original work was translated from English to Spanish," or a modification could indicate "The original work has been modified.";
  - c. to Distribute and Publicly Perform the Work including as incorporated in Collections; and,
  - d. to Distribute and Publicly Perform Adaptations.
  - e. For the avoidance of doubt:
    - i. Non-waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;
    - ii. Waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the

rights granted under this License; and,

- iii. Voluntary License Schemes. The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

- a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(b), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(b), as requested.
- b. If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or

means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and (iv), consistent with Section 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). The credit required by this Section 4 (b) may be implemented in any reasonable manner; provided, however, that in the case of a Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.

- c. Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action prejudicial to the Original Author's honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the applicable national law, to enable You to reasonably exercise Your right

under Section 3(b) of this License (right to make Adaptations) but not otherwise.

#### 5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### 7. Termination

- a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.
- b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

#### 8. Miscellaneous

- a. Each time You Distribute or Publicly Perform the Work or a



Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

- b. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.
- c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.
- d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.
- e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.
- f. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.

Creative Commons is not a party to this License, and makes no

warranty whatsoever in connection with the Work. Creative Commons will not be liable to You or any party on any legal theory for any damages whatsoever, including without limitation any general, special, incidental or consequential damages arising in connection to this license. Notwithstanding the foregoing two (2) sentences, if Creative Commons has expressly identified itself as the Licensor hereunder, it shall have all rights and obligations of Licensor.

Except for the limited purpose of indicating to the public that the Work is licensed under the CCPL, Creative Commons does not authorize the use by either party of the trademark "Creative Commons" or any related trademark or logo of Creative Commons without the prior written consent of Creative Commons. Any permitted use will be in compliance with Creative Commons' then-current trademark usage guidelines, as may be published on its website or otherwise made available upon request from time to time. For the avoidance of doubt, this trademark restriction does not form part of this License.

Creative Commons may be contacted at <http://creativecommons.org/>.

=====

**Notes**





# Image Sources, Licenses and Contributors

**image:PHPUnit\_14.1.png** *Source:* <http://trans.onionmixer.net/mediawiki/index.php?title=> :PHPUnit\_14.1.png *License:* : Onionmixer  
**image:PHPUnit\_14.2.png** *Source:* <http://trans.onionmixer.net/mediawiki/index.php?title=> :PHPUnit\_14.2.png *License:* : Onionmixer  
**image:PHPUnit\_18.1.png** *Source:* <http://trans.onionmixer.net/mediawiki/index.php?title=> :PHPUnit\_18.1.png *License:* : Onionmixer

---