Python TUTORIAL

# Python Unittest Assert Methods

**Summary**: in this tutorial, you'll learn the overview of the Python `unittest` assert methods to perform unit testing.

## Introduction to Python unittest assert methods

The `TestCase` class of the unittest (https://www.pythontutorial.net/python-unit-testing/python-unittest/) module provides you with a large number of assert methods to test. The following table shows the most commonly used assert methods:

| **Method** | **Checks that** |
|---|---|
| `assertEqual(x, y, msg=None)` (https://www.pythontutorial.net/python-unit-testing/python-assertequal/) | `x == y` |
| `assertNotEqual(x,y,msg=None)` (https://www.pythontutorial.net/python-unit-testing/python-assertequal/) | `x != y` |

| Method | Checks that |
|---|---|
| `assertTrue(x, msg=None)` (https://www.pythontutorial.net/python-unit-testing/python-asserttrue/) | `bool(x) is True` |
| `assertFalse(x, msg=None)` (https://www.pythontutorial.net/python-unit-testing/python-asserttrue/) | `bool(x) is False` |
| `assertIs(x, y , msg=None)` (https://www.pythontutorial.net/python-unit-testing/python-assertis/) | `x is y` |
| `assertIsNot(x, y, msg=None)` (https://www.pythontutorial.net/python-unit-testing/python-assertis/) | `x is not y` |
| `assertIsNone(x, msg=None)` (https://www.pythontutorial.net/python-unit-testing/python-assertisnone/) | `x is None` |
| `assertIsNotNone(x , msg=None)` (https://www.pythontutorial.net/python-unit-testing/python-assertisnone/) | `x is not None` |
| `assertIn(x, y, msg=None)` (https://www.pythontutorial.net/python-unit-testing/python-assertin/) | `x in y` |
| `assertNotIn(x, y, msg=None)` (https://www.pythontutorial.net/python-unit-testing/python-assertin/) | `x not in y` |
| `assertIsInstance(x, y, msg=None)` (https://www.pythontutorial.net/python-unit-testing/python-assertisinstance/) | `isinstance(x, y)` |

| Method | Checks that |
|---|---|
| `assertNotIsInstance(x, y, msg=None)` (https://www.pythontutorial.net/python-unit-testing/python-assertisinstance/) | `not isinstance(x, y)` |

All of these methods have an optional `msg` parameter whose type is a string. The `msg` will be displayed in the test result if the test fails.

The following assert methods check the exceptions, warnings, and log messages:

| Method | Checks that |
|---|---|
| `assertRaises(exc, fun, *args, **kwds)` | `fun(*args, **kwds)` raises *exc* |
| `assertRaisesRegex(exc, r, fun, *args, **kwds)` | `fun(*args, **kwds)` raises *exc* and the message matches regex *r* |
| `assertWarns(warn, fun, *args, **kwds)` | `fun(*args, **kwds)` raises *warn* |
| `assertWarnsRegex(warn, r, fun, *args, **kwds)` | `fun(*args, **kwds)` raises *warn* and the message matches regex *r* |
| `assertLogs(logger, level)` | The `with` block logs on *logger* with a minimum *level* |
| `assertNoLogs(logger, level)` | The `with` block does not log on *logger* with a minimum *level* |

The following table shows the assert methods that perform more specific checks:

| Method | Checks that |
|---|---|
| `assertAlmostEqual(x, y)` (https://www.pythontutorial.net/python-unit-testing/python-assertalmostequal/) | `round(x-y, 7) == 0` |

| Method | Checks that |
|---|---|
| `assertNotAlmostEqual(x, y)` (https://www.pythontutorial.net/python-unit-testing/python-assertalmostequal/) | `round(x-y, 7) != 0` |
| `assertGreater(x, y)` | `x > y` |
| `assertGreaterEqual(x, y)` | `x >= y` |
| `assertLess(x, y)` | `x < y` |
| `assertLessEqual(x, y)` | `x <= y` |
| `assertRegex(s, r)` | `r.search(s)` |
| `assertNotRegex(s, r)` | `not r.search(s)` |
| `assertCountEqual(x, y)` | *x* and *y* have the same number of elements in the same number. |

In the next tutorials, you'll learn about the `unittest` assert methods in more detail and how to use them effectively.