



# Python assertTrue()

If this Python Tutorial saves you hours of work, please **whitelist it in your ad blocker** 🙏 and

Donate Now

(<https://www.pythontutorial.net/donation/>)

to help us ❤️ pay for the web hosting fee and CDN to keep the

website running.

**Summary:** in this tutorial, you'll learn how to use Python `assertTrue()` to test if an expression is True and `assertFalse()` method to test if an expression is False.

## Introduction to the Python assertTrue() method

The `assertTrue()` is a method of the `TestCase` class in the `unittest` (<https://www.pythontutorial.net/python-unit-testing/python-unittest/>) module. The `assertTrue()` method tests if an expression is `True` :

```
assertTrue(expr, msg=None)
```

If the `expr` is `True` , the test passes. Otherwise, the test fails.

The `msg` is optional. If you pass the `msg` parameter, it'll be displayed when the test fails.

The `assertTrue()` method is equivalent to the following expression:

```
bool(expr) is True
```

It is not equivalent to the following:

```
expr is True
```

In practice, you should use the `assertTrue()` method to test the boolean of a variable. Also, you should not use the `assertTrue()` method if more specific methods are available.

For example, instead of using the `assertTrue()` method for the following case:

```
assertTrue(a == b)
```

you should use the `assertEqual()` method instead:

```
assertEqual(a,b)
```

Because the `assertEqual()` method provides a more clear error message in case the test fails.

## Python assertTrue() method example

The following example uses the `assertTrue()` method to test if all characters in a string are digits:

```
import unittest

class TestBool(unittest.TestCase):
    def test_is_digit(self):
        str = '123'
        self.assertTrue(str.isdigit())
```

Run the test:

```
python -m unittest -v
```

Output:

```
test_is_digit (test_bool.TestBool) ... ok
```

```
-----  
Ran 1 test in 0.000s  
  
OK
```

## Python assertFalse() method

The `assertFalse()` method tests if an expression is `False`. For example:

```
import unittest  
  
class TestBool(unittest.TestCase):  
    def test_empty_string_is_digit(self):  
        str = ''  
        self.assertFalse(str.isdigit())  
  
    def test_alpha_is_digit(self):  
        str = 'a00'  
        self.assertFalse(str.isdigit())
```

Run the test:

```
python -m unittest -v
```

Output:

```
test_alpha_is_digit (test_bool.TestBool) ... ok  
test_empty_string_is_digit (test_bool.TestBool) ... ok  
test_is_digit (test_bool.TestBool) ... ok  
  
-----  
Ran 3 tests in 0.001s
```

OK

## Summary

- Use the `assertTrue()` method to test if an expression is True
- Use the `assertFalse()` method to test if an expression is False