Skipping Tests

Summary: in this tutorial, you'll learn how to skip tests using the Python unittest module.

The unittest (https://www.pythontutorial.net/python-unit-testing/python-unittest/) module allows you to skip a test method or a test class. To skip a test, you have three available options:

- Use the `@unittest.skip()` decorator.
- Call the `skipTest()` method of the `TestCase` class.
- Raise the `SkipTest` exception.

Skipping a test method examples

The following example uses the `@unittest.skip()` decorator (https://www.pythontutorial.net/advanced-python/python-decorators/) to skip the `test_case_2()` method unconditionally:

```
import unittest


class TestDemo(unittest.TestCase):
    def test_case_1(self):
        self.assertEqual(1+1, 2)
    @unittest.skip('Work in progress')
    def test_case_2(self):
        pass
```

Run the test:

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

Output:

```
test_case_1 (test_skipping.TestDemo) ... ok
test_case_2 (test_skipping.TestDemo) ... skipped 'Work in progress'

----------------------------------------------------------------------
Ran 2 tests in 0.001s

OK (skipped=1)
```

The output shows that two tests were executed and one was skipped. For the skipped test, the output shows the message that we passed to the `@unittest.skip()` decorator.

Similarly, you can call the `skipTest()` in a test method to skip it:

```
import unittest


class TestDemo(unittest.TestCase):
    def test_case_1(self):
        self.assertEqual(1+1, 2)

    def test_case_2(self):
        self.skipTest('Work in progress')
```

Run the test:

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

Output:

```
test_case_1 (test_skipping.TestDemo) ... ok
test_case_2 (test_skipping.TestDemo) ... skipped 'Work in progress'

----------------------------------------------------------------------
Ran 2 tests in 0.001s

OK (skipped=1)
```

Also, you can raise the `SkipTest` exception in a test method to skip it.

```
import unittest


class TestDemo(unittest.TestCase):
    def test_case_1(self):
        self.assertEqual(1+1, 2)

    def test_case_2(self):
        raise unittest.SkipTest('Work in progress')
```

Run the test:

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

Output:

```
test_case_1 (test_skipping.TestDemo) ... ok
test_case_2 (test_skipping.TestDemo) ... skipped 'Work in progress'

----------------------------------------------------------------------
```

```
Ran 2 tests in 0.001s

OK (skipped=1)
```

Skipping a test class examples

To skip a test class, you use the `@unittest.skip()` decorator at the class level.

The following example uses the `@unittest.skip()` decorator at the class level. Hence, all tests of the `TestDemo` class are skipped:

```
import unittest


@unittest.skip('Work in progress')
class TestDemo(unittest.TestCase):
    def test_case_1(self):
        self.assertEqual(1+1, 2)

    def test_case_2(self):
        self.assertIsNotNone([])
```

Run the test:

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

Output:

```
test_case_1 (test_skipping.TestDemo) ... skipped 'Work in progress'
test_case_2 (test_skipping.TestDemo) ... skipped 'Work in progress'

----------------------------------------------------------------------
Ran 2 tests in 0.000s
```

```
OK (skipped=2)
```

Skipping a test with a condition

Sometimes, you want to skip a test conditionally. For example, you may want to skip a test if the current platform, where the test is running, is windows.

To do that you use the `@unittest.skipIf()` decorator:

```
@unittest.skipIf(condition, reason)
```

This will skip the test if the condition is true. Also, it'll display the reason for skipping the test in the test result.

For example:

```
import unittest

from sys import platform


class TestDemo(unittest.TestCase):
    def test_case_1(self):
        self.assertEqual(1+1, 2)

    @unittest.skipIf(platform.startswith("win"), "Do not run on Windows")
    def test_case_2(self):
        self.assertIsNotNone([])
```

In this example, we skip the `test_case_2()` method if the current platform is windows. To get the current platform where the test is running, we use the `sys.platform` property.

Run the test:

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

Output:

```
test_case_1 (test_skipping.TestDemo) ... ok
test_case_2 (test_skipping.TestDemo) ... skipped 'Do not run on Windows'

----------------------------------------------------------------------
Ran 2 tests in 0.001s

OK (skipped=1)
```

Unlike the `@unittest.skipIf` decorator, the `@unittest.skipUnless` skips a test uncles a condition is true:

```
@unittest.skipUnless(condition, reason)
```

Summary

- Use the `@unittest.skip()` decorator, `skipTest()` method, or raise `SkipTest` exception to skip a test.
- Use the `@unittest.skipIf()` or `@unittest.skipUnless()` to skip a test conditionally.