▼ Transcripted Summary

YouTube 🖵 🖸

Let's add a second test case to our "test_math.py" module. However, let's add it with a twist. We will code it to deliberately fail.

Create a new Python test function named "test_one_plus_two" and implement it as follows:

3:44 / 3:44

```
def test_one_plus_two():
    a = 1
    b = 2
    c = 0
    assert a + b == c
```

Clearly, one plus two does not equal zero. This test case should fail when we run it. You might be wondering why we are writing this test case to use variables instead of simply writing assert 1 + 2 == 0. You will see why in just a moment. Let's run the new test. Pytest will discover and run both test cases, and then both will run very quickly.

```
python -m pytest
```

```
tau-intro-to-pytest — -bash — 101×28
sterling2:tau-intro-to-pytest andylpk247$ python -m pytest
             platform darwin -- Python 3.8.1, pytest-5.4.3, py-1.8.1, pluggy-0.13.1
rootdir: /Users/andylpk247/Programming/automation-panda/tau-intro-to-pytest
plugins: metadata-1.9.0, bdd-3.4.0, cov-2.10.0, html-2.1.1, forked-1.1.3, xdist-1.32.0
collected 2 items
                                                                                         [100%]
tests/test_math.py .F
                                     ===== FAILURES =====
                                       test_one_plus_two
   def test_one_plus_two():
      a = 1
      b = 2
     c = 0
     assert a + b == c
     assert (1 + 2) == 0
 tests/test_math.py:22: AssertionError
                                  === short test summary info ==========
FAILED tests/test_math.py::test_one_plus_two - assert (1 + 2) == 0
                                 == 1 failed, 1 passed in 0.07s =====
sterling2:tau-intro-to-pytest andylpk247$
```

Notice how the pytest output prints a "." for the previous test that passes and an "F" for the new test that fails. Pytest also dumps a code snippet for each failing test at the point of failure.

Take a closer look at the line identified as the failure, "assert a + b == c".

Immediately beneath it, pytest prints another line that shows the values of each variable used in the assertion. This feature is called assertion introspection.

It is very helpful when trying to figure out why tests fail because you don't need to re-run the test with debugging to see the variables' values.

Under the failing line, pytest will also print the type of exception that caused the test to fail (in this case, "AssertionError").

When a test fails due to a failed assertion, the exception will be an assertion error. However, tests will fail for _any _unhandled exception type.

For example, if a test tries to access the ninth index in a list of only five elements, then the exception will be an index error instead of an assertion error. Pay

attention to the exception types to help determine if test failures are due to legitimate defects or automation bugs.

At the bottom of the report, pytest lists the paths and names of each failing test with their failures. It then concludes by printing the test result totals. Pytest's

basic test report is both concise and helpful.

Failing tests are bad, so let's go back and fix that.

```
def test_one_plus_two():

a = 1

b = 2

c = 3

assert a + b == c
```

Change c = 0 to c = 3, then re-run the tests.

```
tau-intro-to-pytest — -bash — 101×28
                              ====== FAILURES =========
                                 test_one_plus_two __
   def test_one_plus_two():
    a = 1
    b = 2
    c = 0
    assert a + b == c
    assert (1 + 2) == 0
tests/test_math.py:22: AssertionError
                      FAILED tests/test_math.py::test_one_plus_two - assert (1 + 2) == 0
                          === 1 failed, 1 passed in 0.07s ======
sterling2:tau-intro-to-pytest andylpk247$
sterling2:tau-intro-to-pytest andylpk247$
sterling2:tau-intro-to-pytest andylpk247$ python -m pytest
platform darwin -- Python 3.8.1, pytest-5.4.3, py-1.8.1, pluggy-0.13.1
rootdir: /Users/andylpk247/Programming/automation-panda/tau-intro-to-pytest
plugins: metadata-1.9.0, bdd-3.4.0, cov-2.10.0, html-2.1.1, forked-1.1.3, xdist-1.32.0
collected 2 items
                                                                           [100%]
tests/test_math.py ...
                  sterling2:tau-intro-to-pytest andylpk247$
```

Pytest should now report two passing tests. Nice. If you'd like to learn more about how pytest handles assertions, check out the page, "The writing and reporting of assertions in tests " on pytest.org.

Resources • Assertions in pytest 🗗

```
Hide my answers
 1. One Python test module can contain more than one test case function.
  true
  false
 2. What symbol does pytest use in its reports to denote a passing test?
  !
   р
   0
 3. What symbol does pytest use in its reports to denote a failing test?
  _
 4. What types of exceptions will make a pytest test case fail?
   any type of exception
   only AssertionError
   any exception other than AssertionError
 5. By default, pytest will print test code snippets, failure reasons, and test result tallies
 for failed test cases.
   true
  false
        Note: 100 credits is for successful completion on the first try; 50 credits for the second try, and 25 credits thereafter
Prev Chapter
                                                                                             Next Chapter
                                                     in Share
                                           Share
```