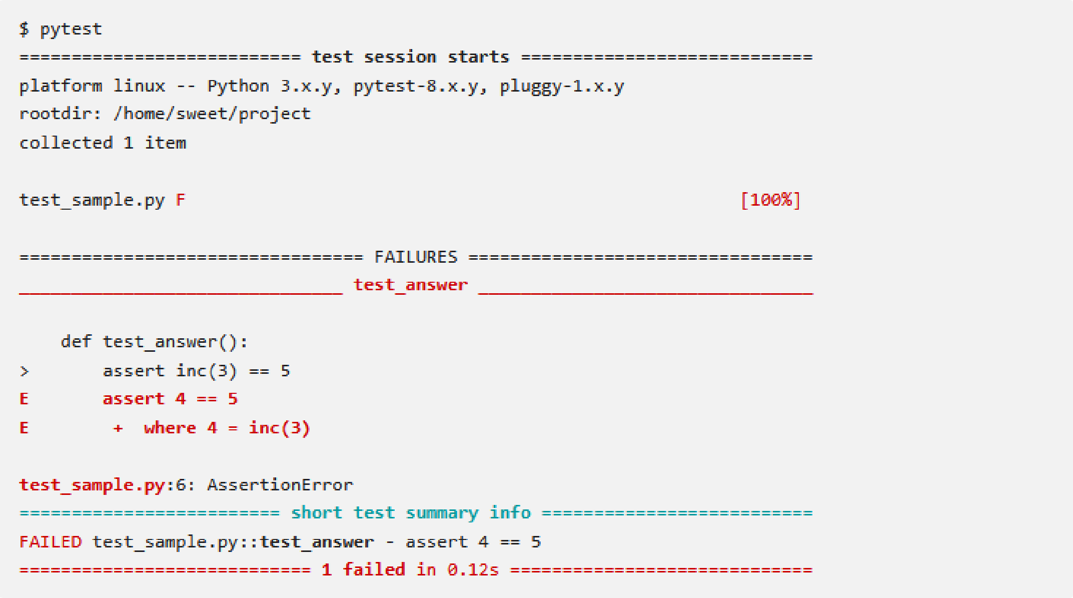
Pytest and Coverage.py

Sechang Jang

SoC KAIST

Pytest

- A simple, powerful, and modern testing framework for Python.
  - https://docs.pytest.org/en/stable/
  - https://pypi.org/project/pytest/



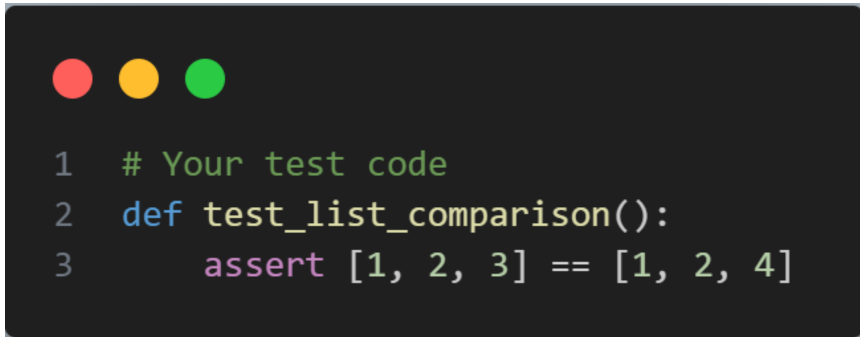
```
$ pytest
=========================== test session starts ============================
platform linux -- Python 3.x.y, pytest-8.x.y, pluggy-1.x.y
rootdir: /home/sweet/project
collected 1 item

test_sample.py F                                                     [100%]

================================= FAILURES =================================
_______________________________ test_answer ________________________________

    def test_answer():
>       assert inc(3) == 5
E       assert 4 == 5
E        +  where 4 = inc(3)

test_sample.py:6: AssertionError
========================= short test summary info ==========================
FAILED test_sample.py::test_answer - assert 4 == 5
============================ 1 failed in 0.12s =============================
```

Image from https://docs.pytest.org/en/stable/

Getting Started

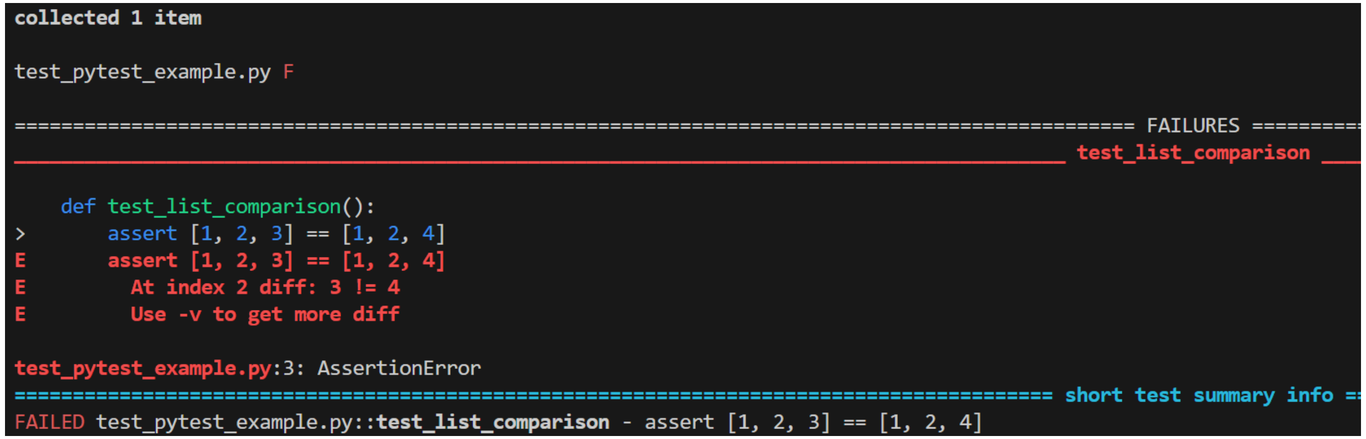
- **0. Use python 3.14 or higher version**
- **1. Install it:**
  - `pip install pytest`
- **2. How pytest Finds Tests:**
  - File Names: `test_*.py` or `*_test.py`
  - Function Names: `def test_...():`
- **3. How to Run it:**
  - Type **`pytest`** in your terminal on project root directory.
    It will automatically find and run all your tests.

Example



```
1  # Your test code
2  def test_list_comparison():
3      assert [1, 2, 3] == [1, 2, 4]
```

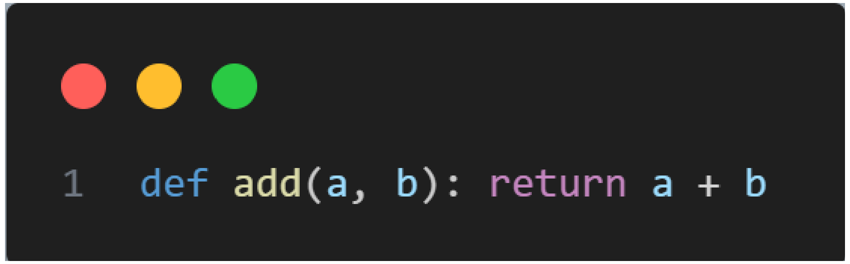
test_pytest_example.py



```
collected 1 item

test_pytest_example.py F

============================================================================================= FAILURES =========
                                                                                           test_list_comparison ____

    def test_list_comparison():
>       assert [1, 2, 3] == [1, 2, 4]
E       assert [1, 2, 3] == [1, 2, 4]
E         At index 2 diff: 3 != 4
E         Use -v to get more diff

test_pytest_example.py:3: AssertionError
========================================================================================= short test summary info ==
FAILED test_pytest_example.py::test_list_comparison - assert [1, 2, 3] == [1, 2, 4]
```

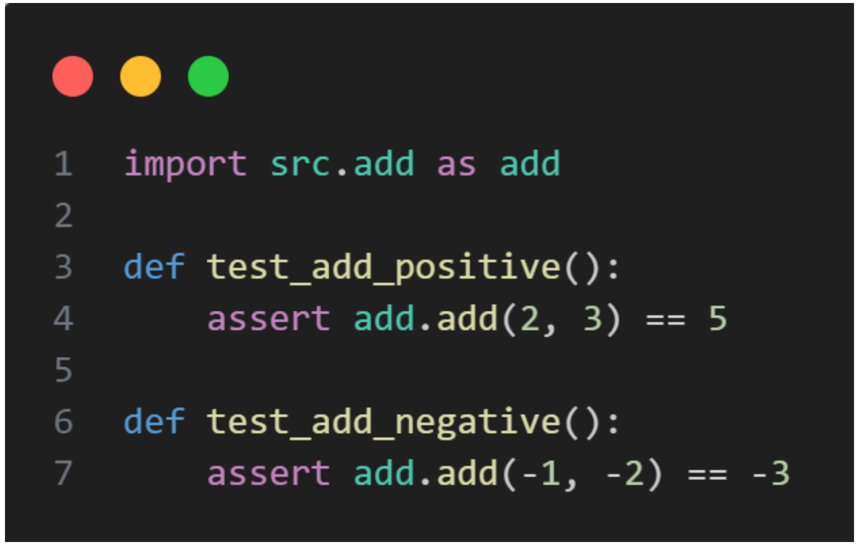
pytest result

Example



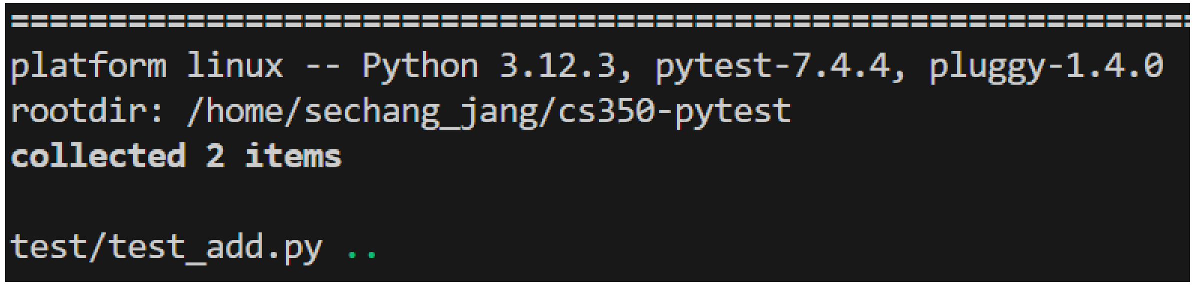
```python
def add(a, b): return a + b
```

Source Code



```python
import src.add as add

def test_add_positive():
    assert add.add(2, 3) == 5

def test_add_negative():
    assert add.add(-1, -2) == -3
```

Test Code

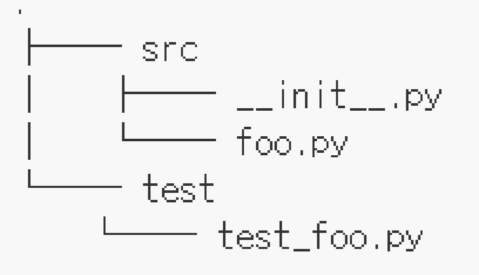


```
============================================================
platform linux -- Python 3.12.3, pytest-7.4.4, pluggy-1.4.0
rootdir: /home/sechang_jang/cs350-pytest
collected 2 items

test/test_add.py ..
```

Pytest Result

Tips

- Pytest might raise “module not found error” in such scenario
  - A. Create __init__.py file in a directory where test_*.py is located
  - B. Or Run “python -m pytest” instead of pytest



```
.
├──── src
│     ├──── __init__.py
│     └──── foo.py
└──── test
    └──── test_foo.py
```

Directory Layout

Coverage

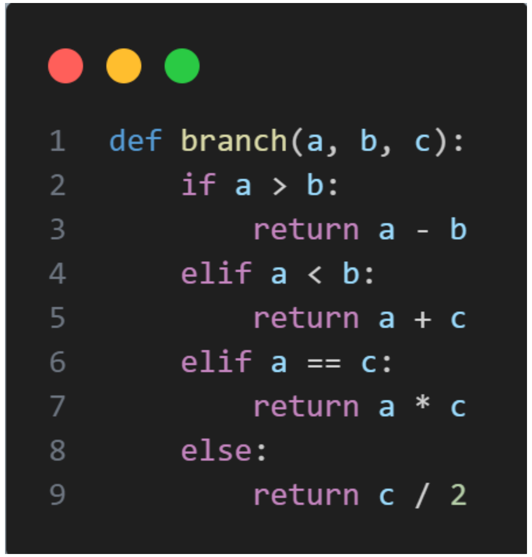
- Coverage.py is a tool for measuring code coverage of Python programs.
    - Code Coverage: A metric that measures which lines and branches of your source code were executed by your test suite.
    - https://coverage.readthedocs.io/en/7.11.3/
    - There must be test case to measure the coverage!
        - Coverage.py offers integration with pytest using “-m pytest” option

Getting Started

- **0. Use python 3.14 or higher version**
- **1. Install it:**
  - `pip install coverage`
- **2. Run it**
  - `coverage run --branch -m pytest`
- **3. See the Report**
  - `coverage report –m`

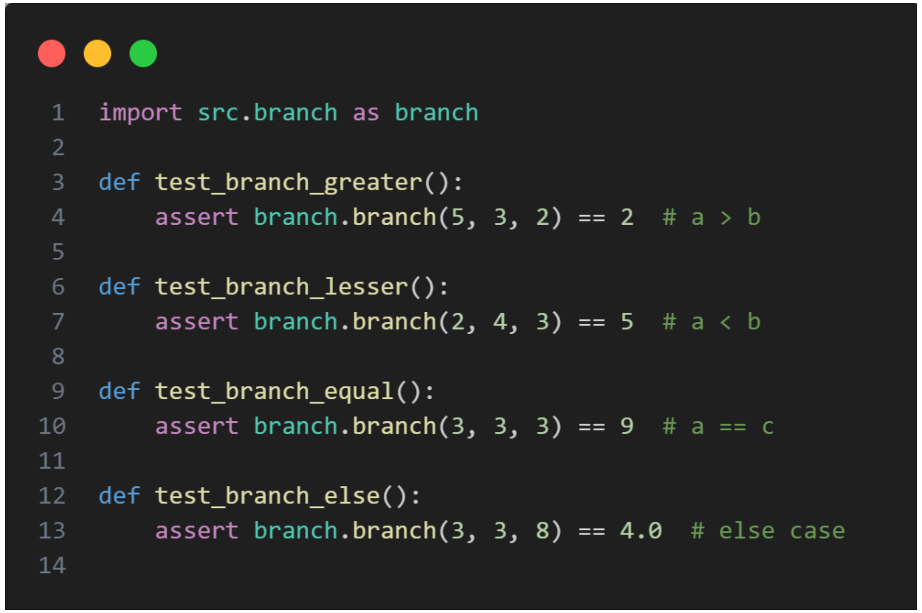
    (Show a screenshot of the terminal output: Name | Stmts | Miss | Cover | Missing)
  - `coverage html (or xml)`
    - (This creates an htmlcov/ folder. Open index.html in your browser.)

Example



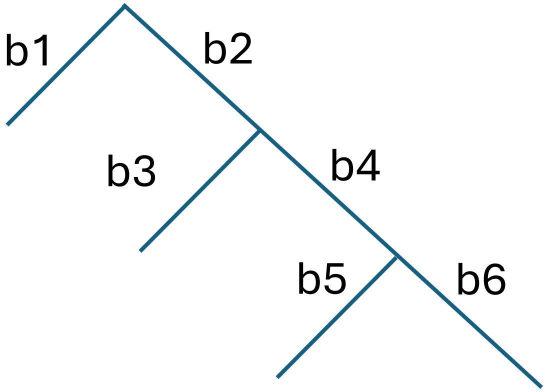
```python
def branch(a, b, c):
    if a > b:
        return a - b
    elif a < b:
        return a + c
    elif a == c:
        return a * c
    else:
        return c / 2
```

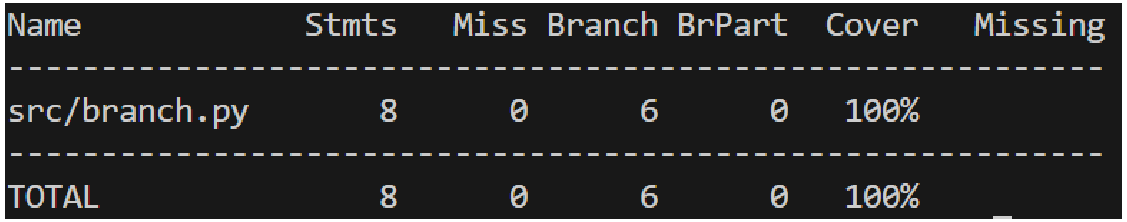
Source Code



```python
import src.branch as branch

def test_branch_greater():
    assert branch.branch(5, 3, 2) == 2  # a > b

def test_branch_lesser():
    assert branch.branch(2, 4, 3) == 5  # a < b

def test_branch_equal():
    assert branch.branch(3, 3, 3) == 9  # a == c

def test_branch_else():
    assert branch.branch(3, 3, 8) == 4.0  # else case

```

Test Code

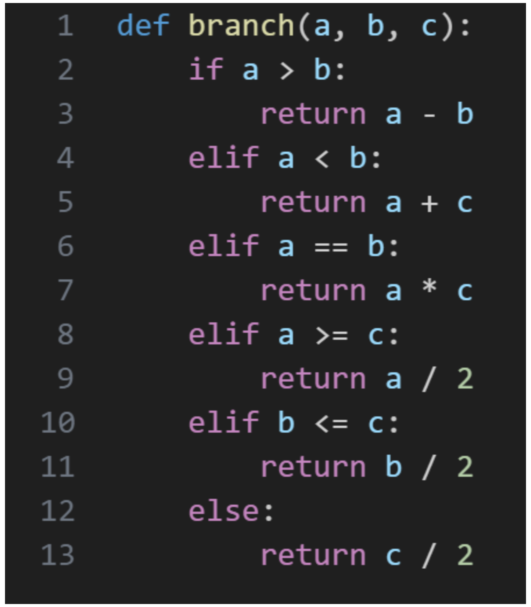




```
Name           Stmts   Miss Branch BrPart  Cover   Missing
----------------------------------------------------------
src/branch.py      8      0      6      0   100%
----------------------------------------------------------
TOTAL              8      0      6      0   100%
```

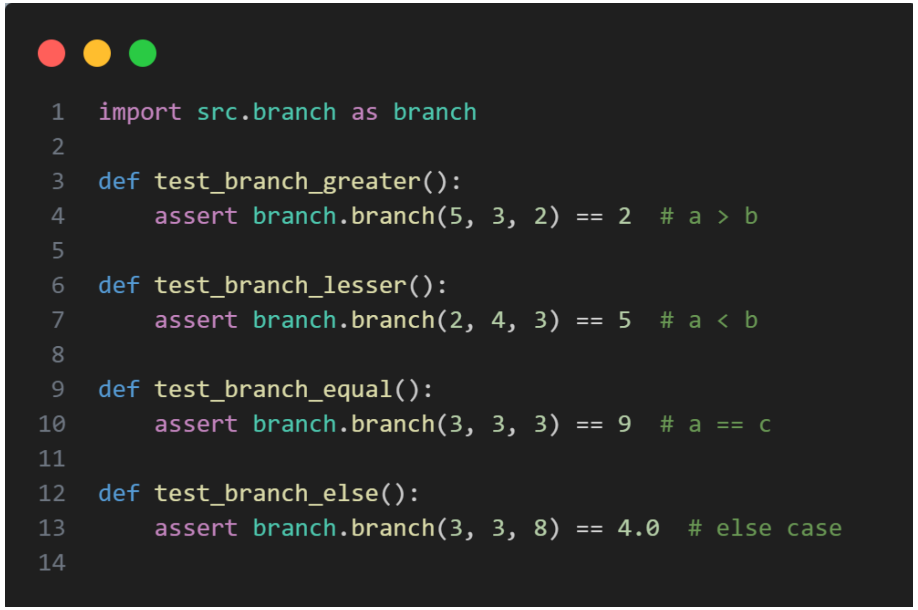
Coverage Report

Example



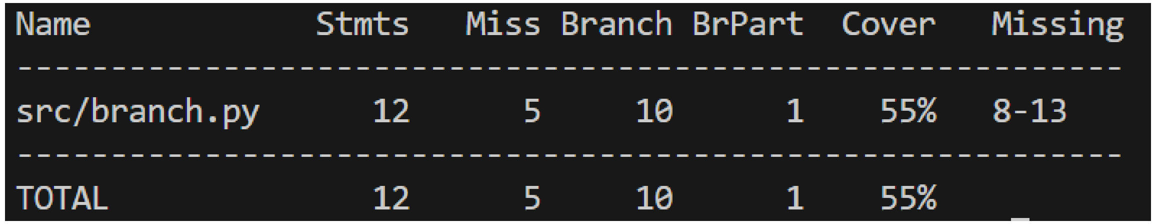
```
 1  def branch(a, b, c):
 2      if a > b:
 3          return a - b
 4      elif a < b:
 5          return a + c
 6      elif a == b:
 7          return a * c
 8      elif a >= c:
 9          return a / 2
10      elif b <= c:
11          return b / 2
12      else:
13          return c / 2
```

Source Code



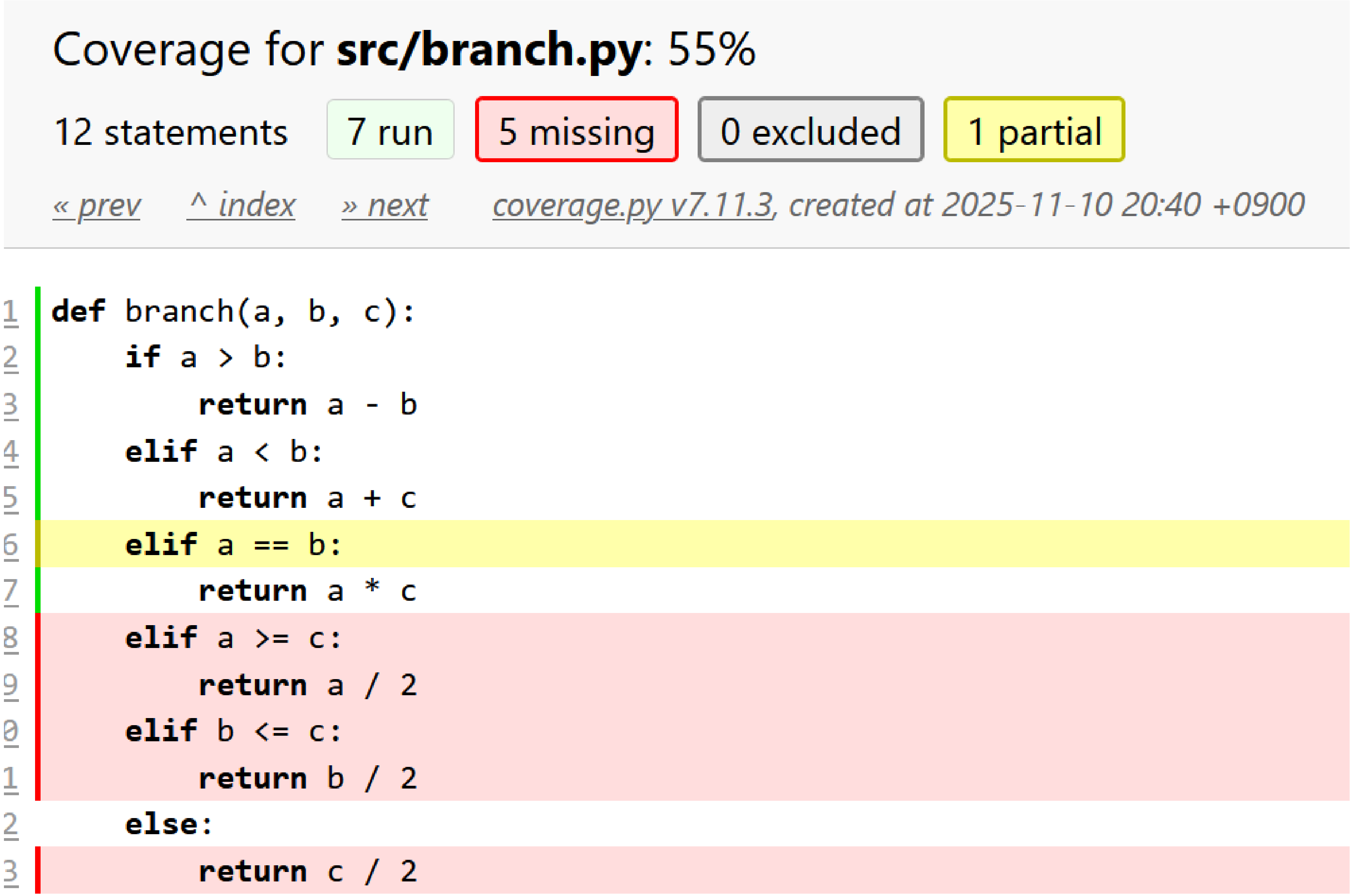
```
import src.branch as branch

def test_branch_greater():
    assert branch.branch(5, 3, 2) == 2  # a > b

def test_branch_lesser():
    assert branch.branch(2, 4, 3) == 5  # a < b

def test_branch_equal():
    assert branch.branch(3, 3, 3) == 9  # a == c

def test_branch_else():
    assert branch.branch(3, 3, 8) == 4.0  # else case

```

Test Code



```
Name             Stmts   Miss Branch BrPart  Cover   Missing
------------------------------------------------------------
src/branch.py       12      5     10      1    55%   8-13
------------------------------------------------------------
TOTAL               12      5     10      1    55%
```

Coverage Report



Coverage for **src/branch.py**: 55%

12 statements 7 run 5 missing 0 excluded 1 partial

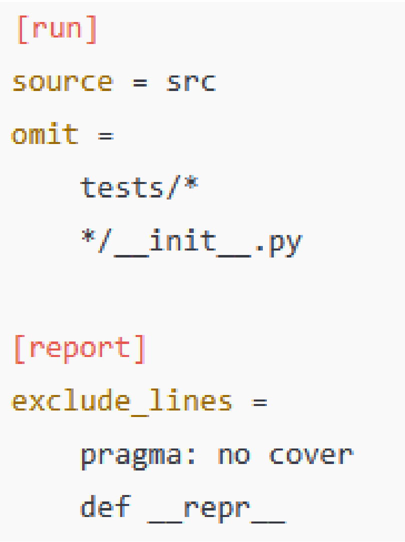
coverage.py v7.11.3, created at 2025-11-10 20:40 +0900

```
def branch(a, b, c):
    if a > b:
        return a - b
    elif a < b:
        return a + c
    elif a == b:
        return a * c
    elif a >= c:
        return a / 2
    elif b <= c:
        return b / 2
    else:
        return c / 2
```

html Coverage Report

Tips

- .coverage file
  - It lets you control what code is measured,
    which files to ignore,
    and how reports are displayed.
  - Use “exclude_lines”
  - Detailed Branch coverage can be extracted using xml report



```
[run]
source = src
omit =
    tests/*
    */__init__.py

[report]
exclude_lines =
    pragma: no cover
    def __repr__
```

- **You can use pytest-cov as well!**
  - pytest-cov is a pytest plugin that integrates coverage.py
  - integrates coverage collection into pytest seamlessly