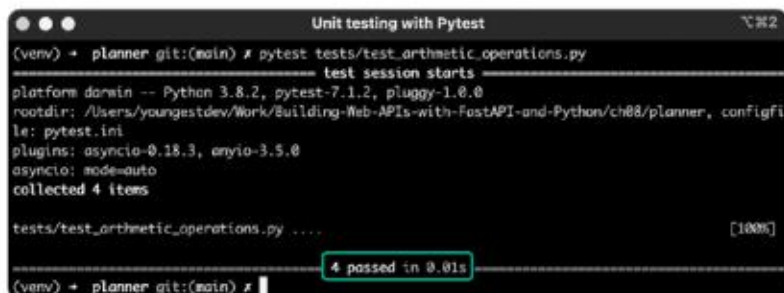


```
(venv)$ pytest test_arithmetic_operations.py
```

테스트가 모두 성공하면 통과된 테스트는 터미널 창에 초록색으로 표시된다.



```
Unit testing with Pytest
(venv) + planner git:(main) x pytest tests/test_arithmetic_operations.py
test session starts
platform darwin -- Python 3.8.2, pytest-7.1.2, pluggy-1.0.0
rootdir: /Users/youngestdev/Work/Building-Web-APIs-with-FastAPI-and-Python/ch08/planner, configfile: pytest.ini
plugins: asyncio-0.18.3, anyio-3.5.0
asyncio: mode=auto
collected 4 items

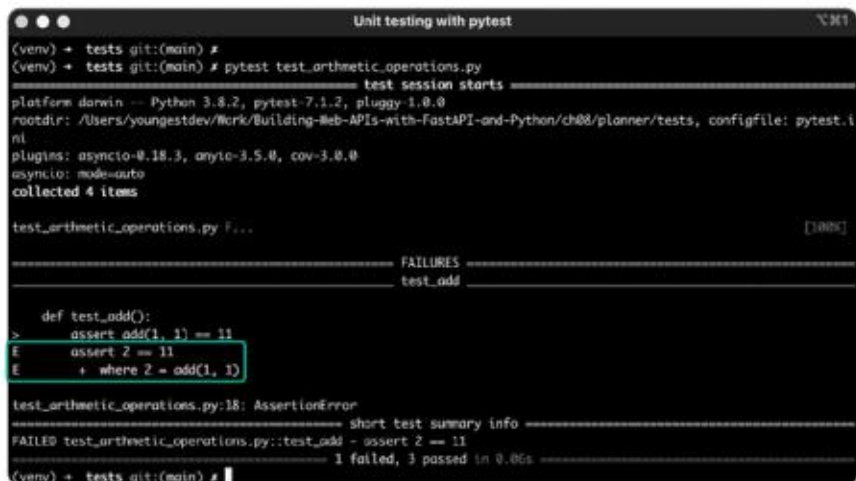
tests/test_arithmetic_operations.py .... [100%]

4 passed in 0.01s
(venv) + planner git:(main) x
```

그림 8-1 사칙연산 결과를 확인하는 단위 테스트(이 그림에는 색상이 표현되지 않았지만 상자 안의 텍스트가 초록색으로 표시된다)

테스트가 실패하면 터미널 창에 실패한 위치가 빨간색으로 표시된다. 예를 들어 `test_add()` 함수를 다음과 같이 변경하면 테스트에 실패한다.

```
def test_add() -> None:
    assert add(1, 1) == 11
```



```
Unit testing with pytest
(venv) + tests git:(main) x
(venv) + tests git:(main) x pytest test_arithmetic_operations.py
test session starts
platform darwin -- Python 3.8.2, pytest-7.1.2, pluggy-1.0.0
rootdir: /Users/youngestdev/Work/Building-Web-APIs-with-FastAPI-and-Python/ch08/planner/tests, configfile: pytest.ini
plugins: asyncio-0.18.3, anyio-3.5.0, cov-3.0.0
asyncio: mode=auto
collected 4 items

test_arithmetic_operations.py F... [100%]

===== FAILURES =====
test_add

def test_add():
> assert add(1, 1) == 11
E assert 2 == 11
E + where 2 = add(1, 1)

test_arithmetic_operations.py:18: AssertionError
===== short test summary info =====
FAILED test_arithmetic_operations.py::test_add - assert 2 == 11
1 failed, 3 passed in 0.00s
(venv) + tests git:(main) x
```

그림 8-2 실패한 테스트(이 그림에는 색상이 표현되지 않았지만 상자 안의 텍스트가 빨간색으로 표시된다)