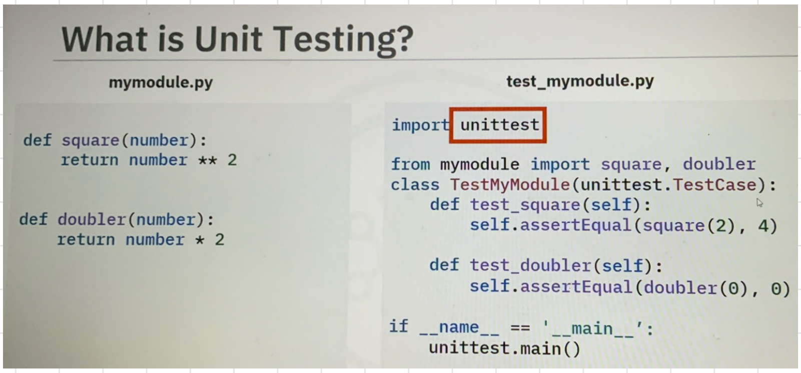


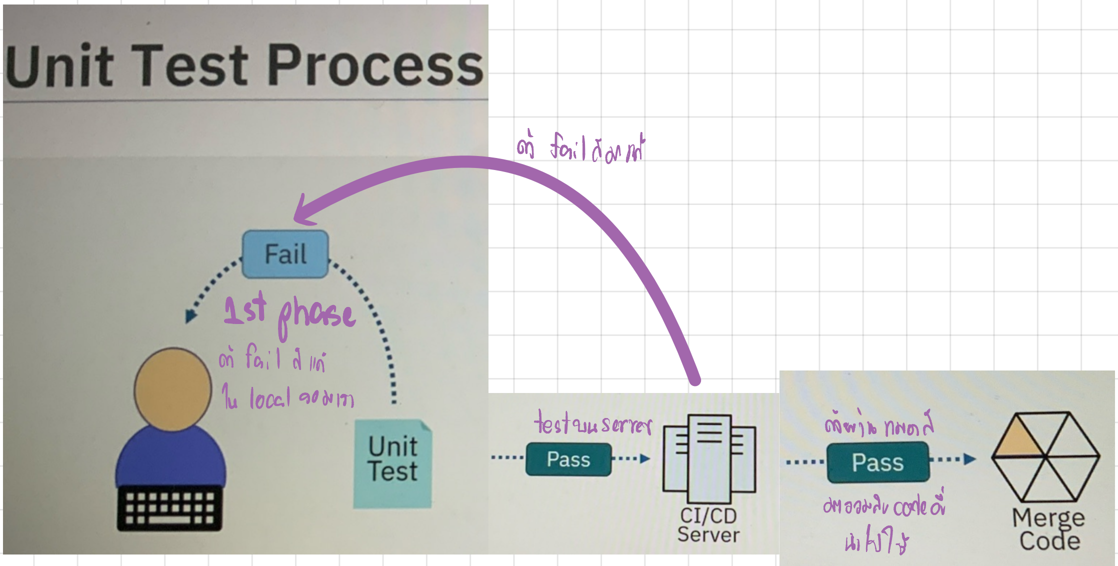
## What is Unit Testing?

mymodule.py

```
def square(number):
    return number ** 2

def doubler(number):
    return number * 2
```

test_mymodule.py

```
import unittest

from mymodule import square, doubler
class TestMyModule(unittest.TestCase):
    def test_square(self):
        self.assertEqual(square(2), 4)

    def test_doubler(self):
        self.assertEqual(doubler(0), 0)

if __name__ == '__main__':
    unittest.main()
```



## Unit Test Process

ถ้า fail ต้องแก้

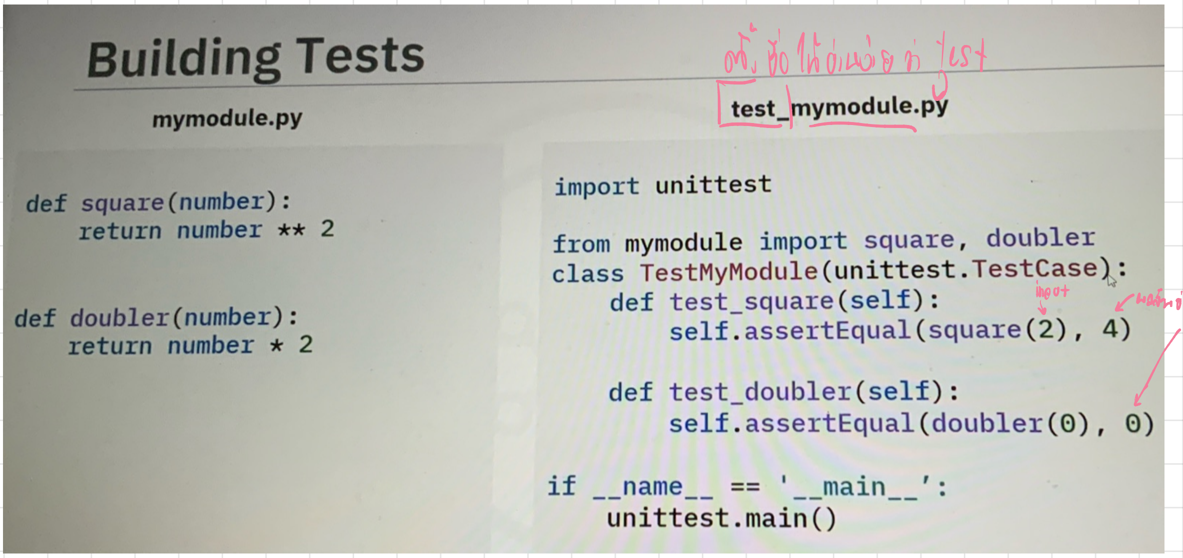
1st phase

ถ้า fail ก็แก้ ใน local ของเรา

test บน server

ถ้าผ่าน ทดสอบ

ต้องรวม code ที่ แก้ไข



## Building Tests

ตั้งชื่อให้ขึ้นต้นว่า test

mymodule.py

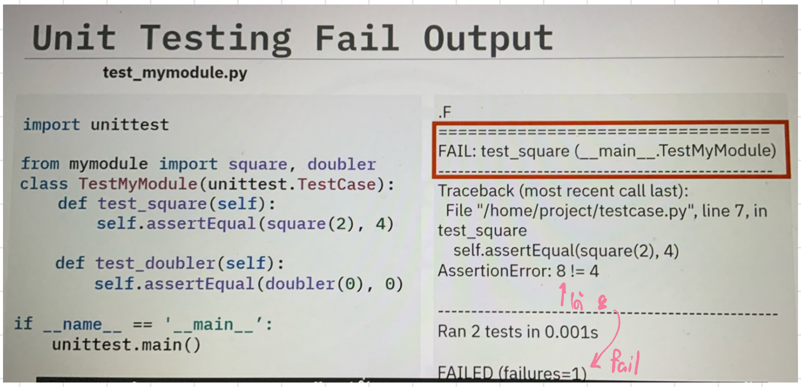
```
def square(number):
    return number ** 2

def doubler(number):
    return number * 2
```

test_mymodule.py

```
import unittest

from mymodule import square, doubler
class TestMyModule(unittest.TestCase):
    def test_square(self):
        self.assertEqual(square(2), 4)

    def test_doubler(self):
        self.assertEqual(doubler(0), 0)

if __name__ == '__main__':
    unittest.main()
```

input

ผลลัพธ์ที่ต้องการได้



## Unit Testing Fail Output

test_mymodule.py

```
import unittest

from mymodule import square, doubler
class TestMyModule(unittest.TestCase):
    def test_square(self):
        self.assertEqual(square(2), 4)

    def test_doubler(self):
        self.assertEqual(doubler(0), 0)

if __name__ == '__main__':
    unittest.main()
```

```
.F
=================================
FAIL: test_square (__main__.TestMyModule)
---------------------------------------------
Traceback (most recent call last):
  File "/home/project/testcase.py", line 7, in
test_square
    self.assertEqual(square(2), 4)
AssertionError: 8 != 4

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

FAILED (failures=1)
```

ได้ 8

fail