Python Quick Tips
Defining Functions

Python Quick Tips #5

Defining Functions

Built-In

So far, most functions we’ve used are **built-in**

Defined

But we can actually define **our own!**

Why?

Simplify our script

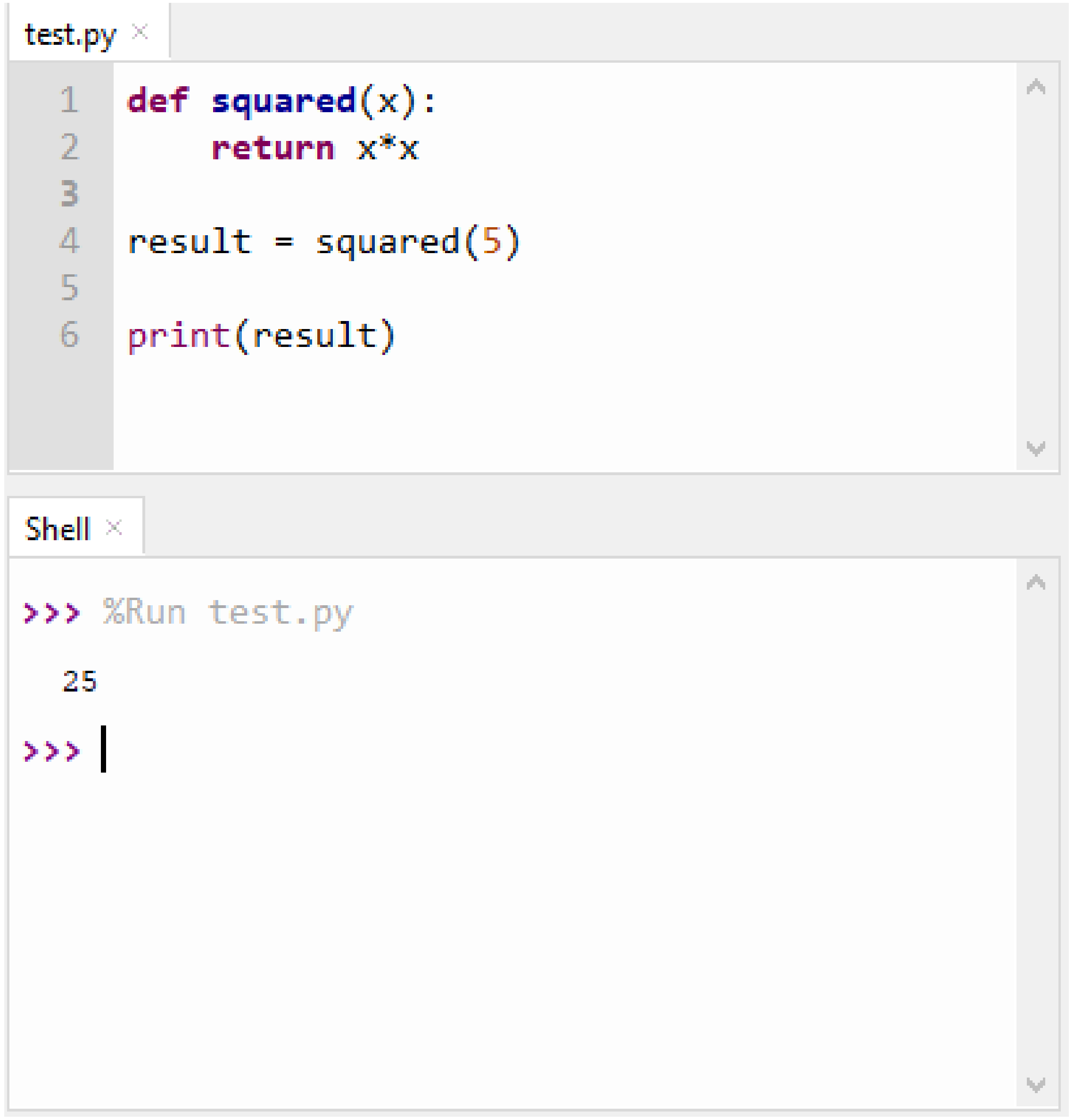
Consolidate **repetition**

Define (def)

Syntax

```
def name(var):
(tab)code
(tab)etc.
return = result
```

Define (def)

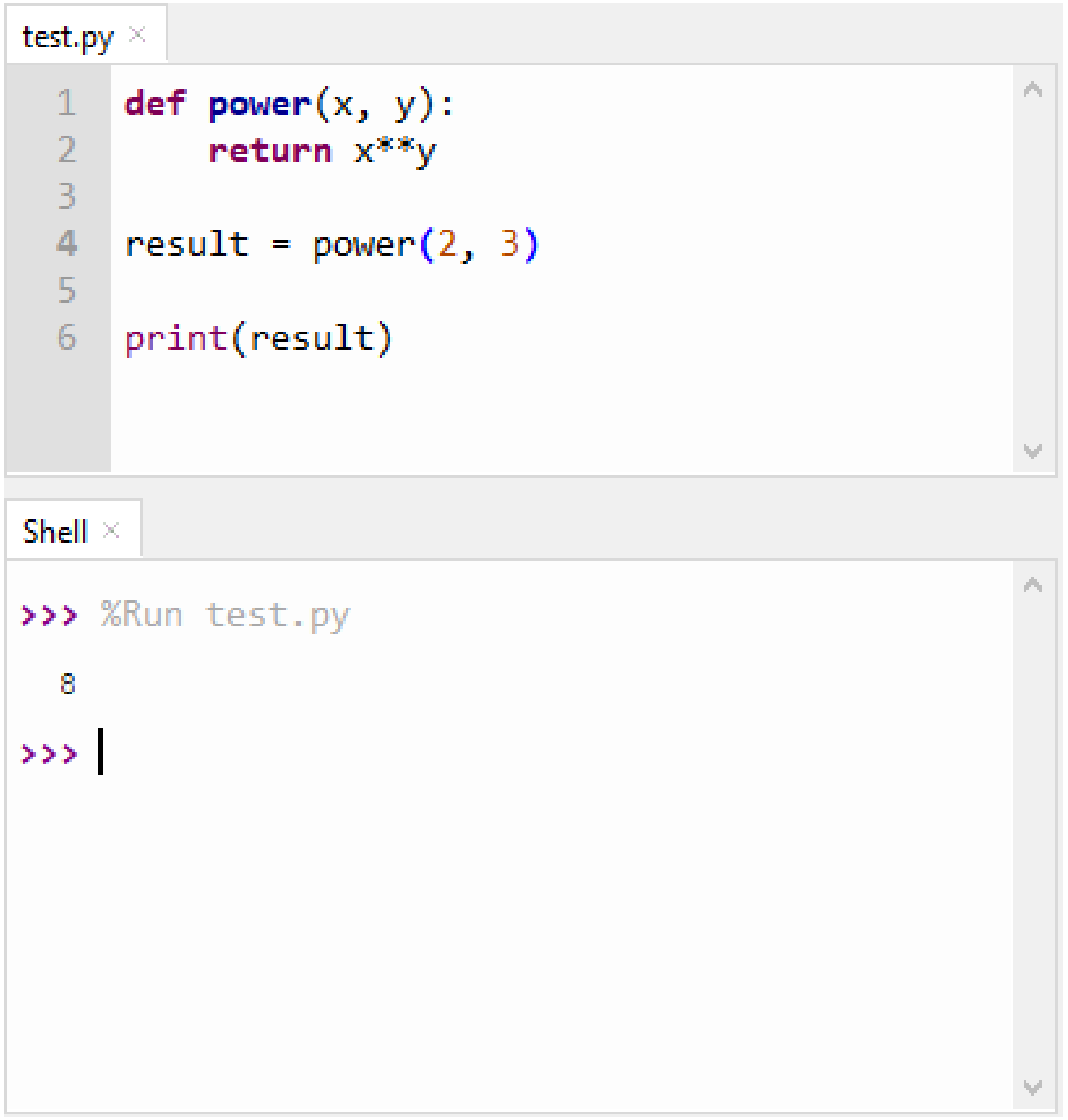

test.py

```python
def squared(x):
    return x*x

result = squared(5)

print(result)
```

Shell

```
>>> %Run test.py
  25
>>> 
```

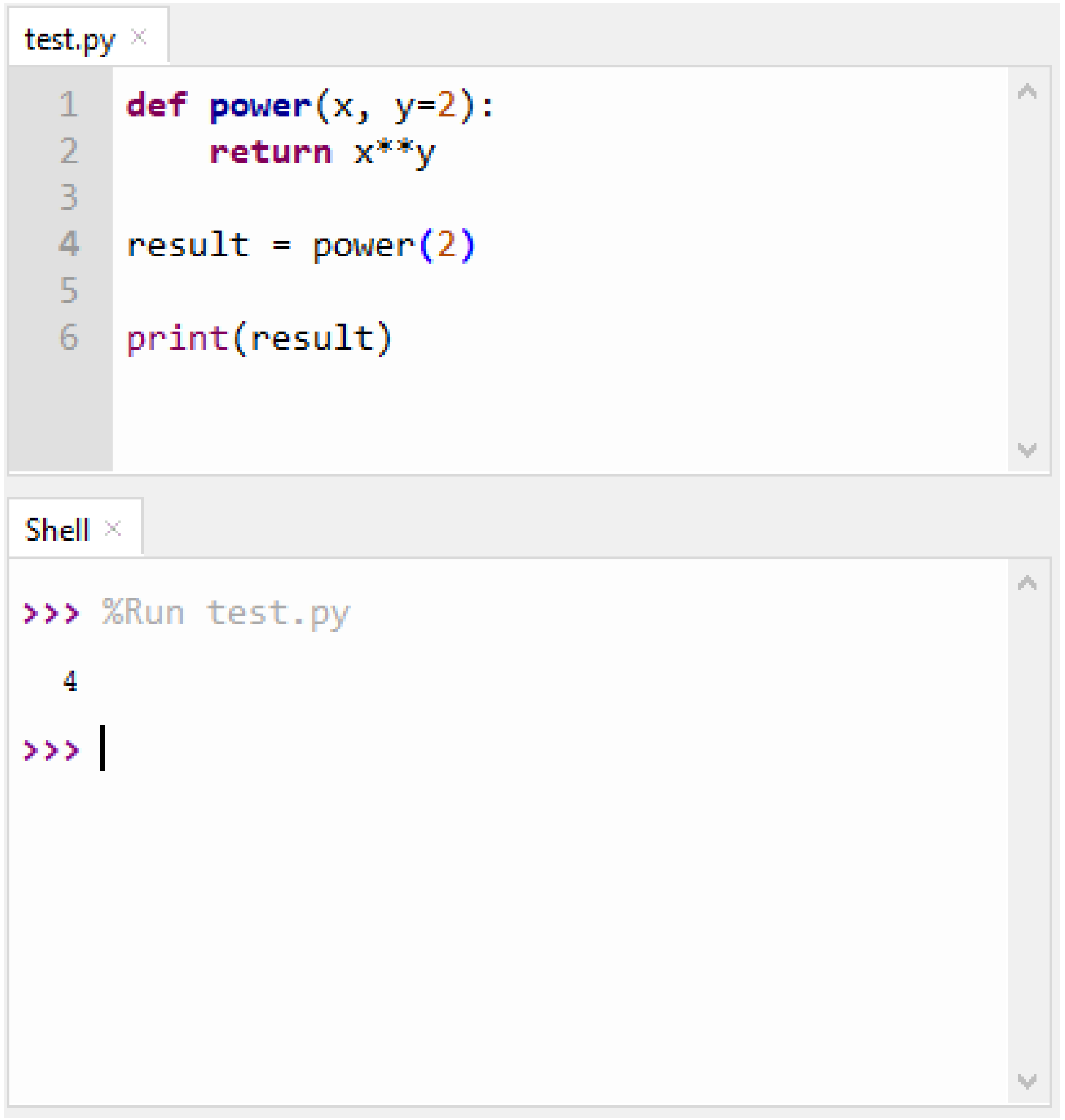
Multiple Variables



```
def power(x, y):
    return x**y

result = power(2, 3)

print(result)
```

```
>>> %Run test.py
  8
>>> 
```

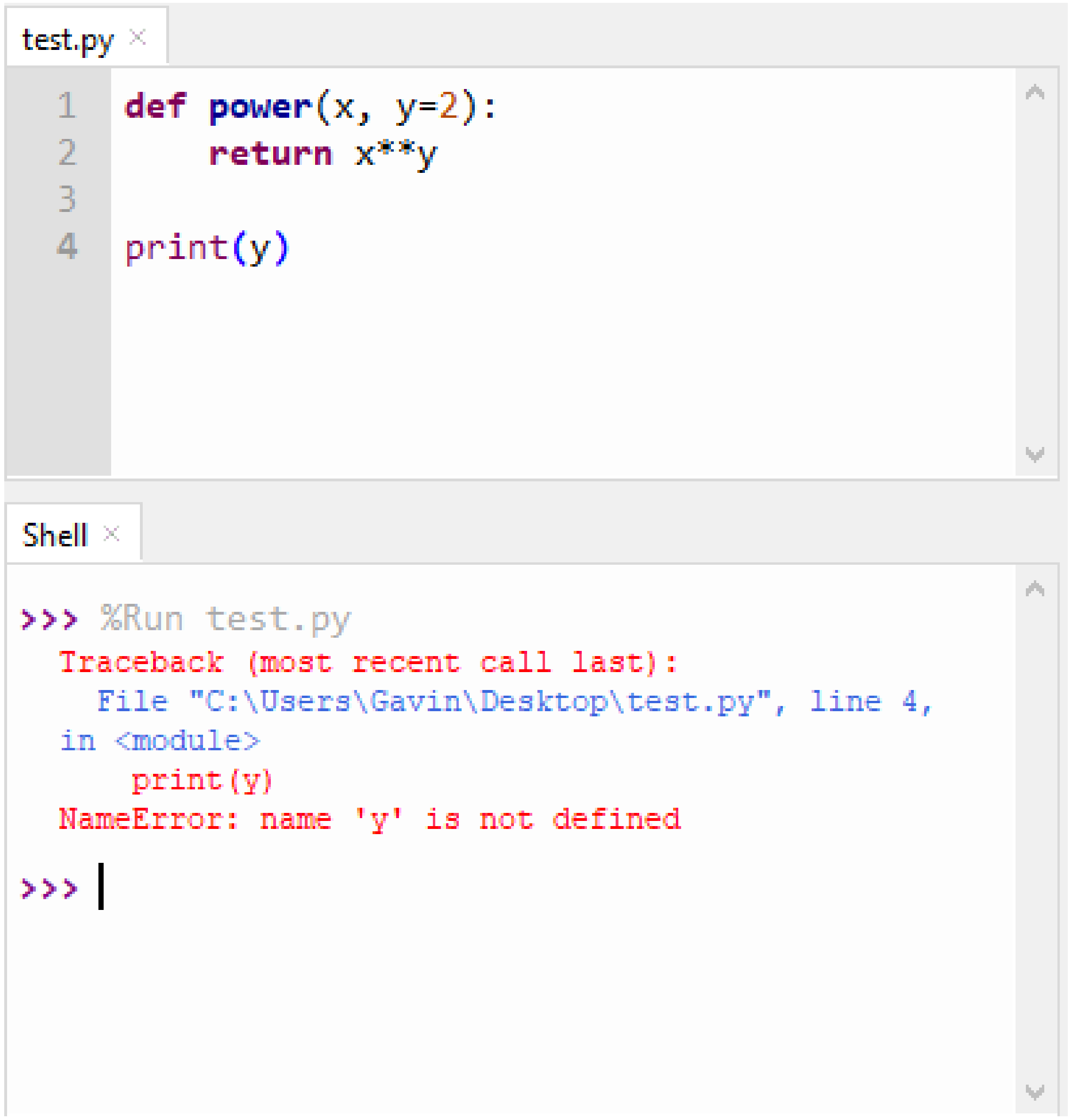
Default Values



```python
def power(x, y=2):
    return x**y

result = power(2)

print(result)
```

```
>>> %Run test.py
  4
>>> 
```

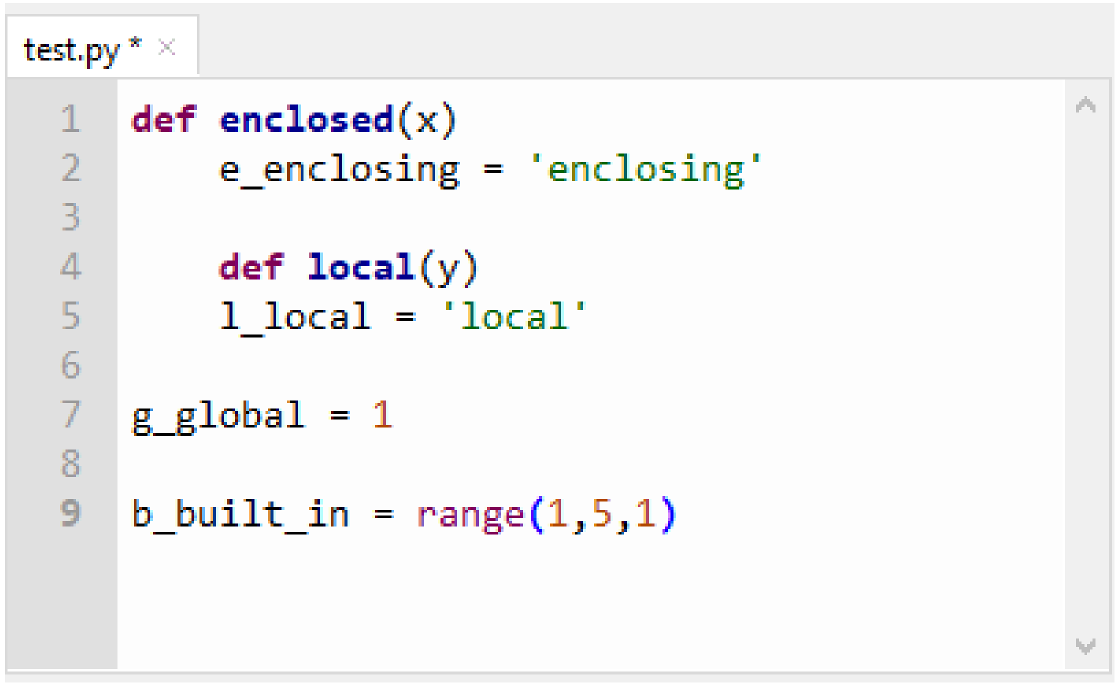
Local Variables



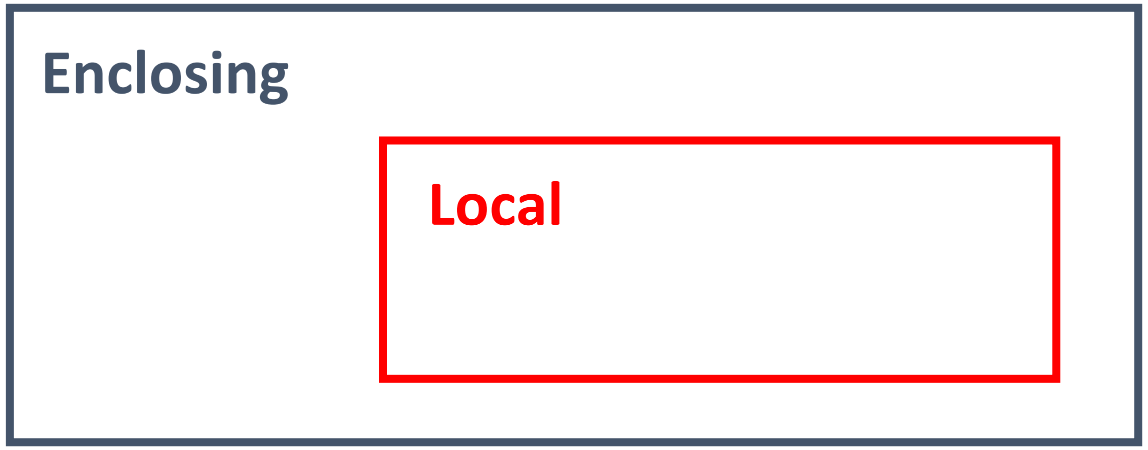
```python
def power(x, y=2):
    return x**y

print(y)
```

```
>>> %Run test.py
  Traceback (most recent call last):
    File "C:\Users\Gavin\Desktop\test.py", line 4,
  in <module>
      print(y)
  NameError: name 'y' is not defined

>>> 
```

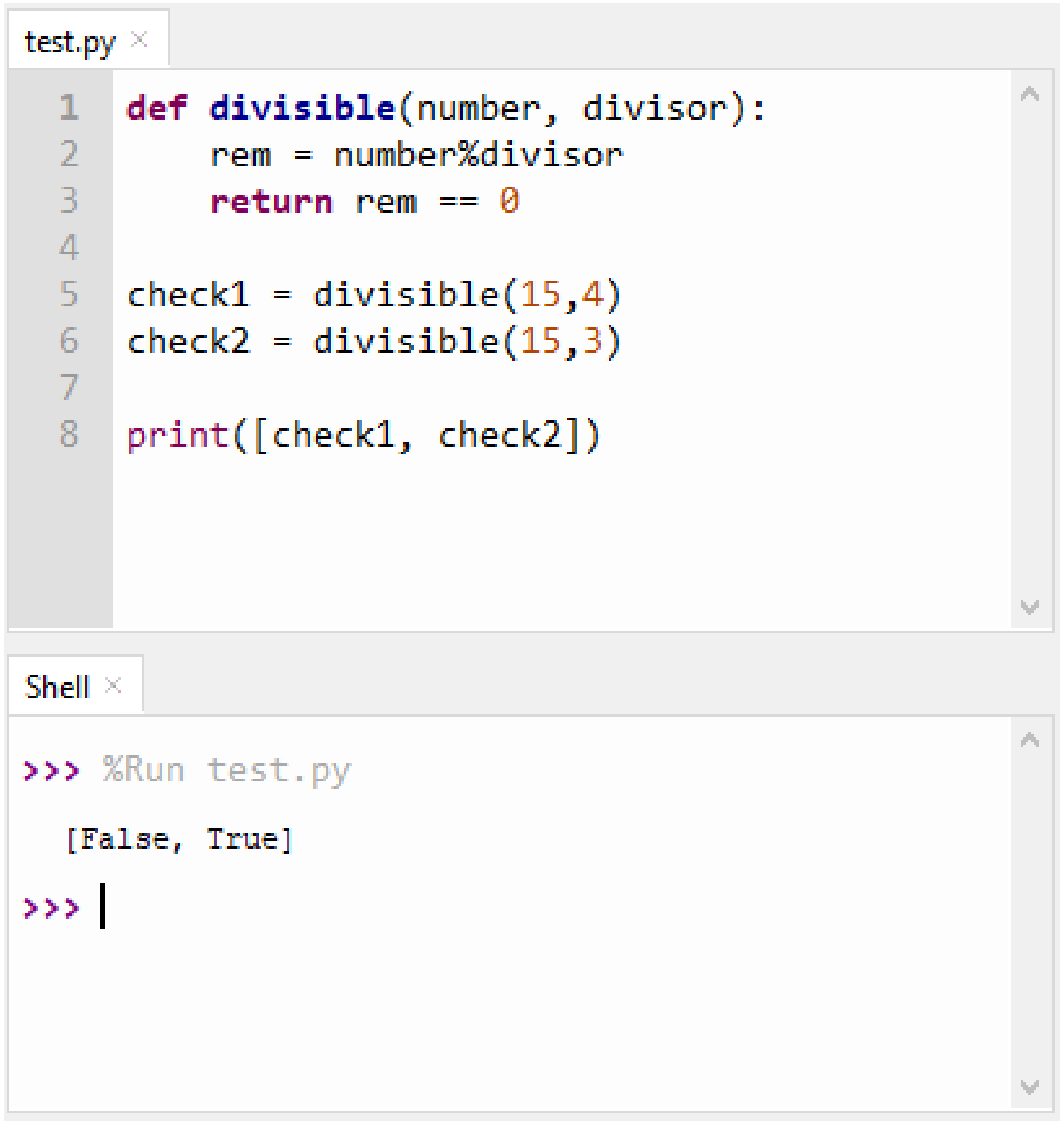
LEGB



```python
def enclosed(x)
    e_enclosing = 'enclosing'

    def local(y)
    l_local = 'local'

g_global = 1

b_built_in = range(1,5,1)
```



Another Example



```python
def divisible(number, divisor):
    rem = number%divisor
    return rem == 0

check1 = divisible(15,4)
check2 = divisible(15,3)

print([check1, check2])
```

```
>>> %Run test.py
  [False, True]
>>> 
```

Next on #6

If Statements

5

Python Quick Tips

Defining Functions