

Создадим файл `myprog.py`, содержащий следующий код (тело функции должно отделяться четырьмя пробелами)¹³:

```
def f(x):  
    x = 2 * x  
    return x
```

Запустим программу с помощью F5. Увидим, что в интерактивном режиме программа выполнялась, но ничего не вывела на экран. Правильно, ведь мы не вызвали функцию!

```
===== RESTART: C:/Python35-32/myprog.py =====  
>>>
```

После запуска программы в интерактивном режиме вызовем функцию `f` с различными аргументами:

```
>>> f(4)  
8  
>>> f(56)  
112  
>>>
```

Все работает! Теперь вызовем функцию `f` в файле, но не забываем про `print`. Обновленная версия файла `myprog.py` будет иметь вид:

```
def f(x):  
    x = 2 * x  
    return x  
  
print(f(4))    # комментарии игнорируются Python  
print(f(56))
```

Запустим программу с помощью F5 и увидим, что в интерактивном режиме отобразился результат!

```
===== RESTART: C:/Python35-32/myprog.py =====  
8  
112  
>>>
```

Теперь поговорим об области видимости переменных. Ранее мы сказали, что переменная является локальной (видна только внутри функции), если значение ей присваивается внутри функций, в ином случае – переменная глобальная, т.е. видна (к ней можно обратиться) во всей программе, в том числе и внутри функции.

¹³ Разработчики на языке Python очень трепетно относятся к оформлению исходного кода, поэтому следуют ряду правил, собранных в документе под названием PEP8.