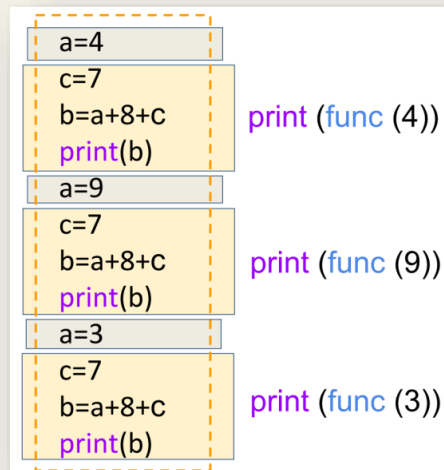


Часто функции используются для сокращения кода программы, например, объявив функцию вида:

```
def func(x):  
    c = 7  
    return x + 8 + c
```

Следующий код может быть заменен на три вызова функции с различными аргументами:



В файле не забываем вызывать функцию print.

Бывают случаи, когда наша функция ничего не принимает на вход и ничего не возвращает<sup>16</sup> (не используется ключевое слово return). Пример подобной функции:

```
def print_hello():  
    print('Привет')  
    print('Hello')  
    print('Hi')
```

Видим, что внутри функции происходит вызов print, поэтому в момент вызова функции print\_hello еще раз вызывать print не требуется. Следующий пример демонстрирует, что множество вызовов print можно заменить тремя вызовами функции print\_hello:

---

<sup>16</sup> На самом деле, если не указать return, то Python вернет объект None:

```
>>> def f(x):  
    print(x**2 + 1)  
  
>>> f(2)  
5  
>>> 3*f(2)+1 # f(2) вернет объект None типа NoneType  
5  
Traceback (most recent call last):  
  File "<pyshell#14>", line 1, in <module>  
    3*f(2)+1  
TypeError: unsupported operand type(s) for *: 'int' and 'NoneType'  
>>>
```