

новые<sup>10</sup>. Это особенность числового типа данных – объекты этого типа являются **неизменяемыми**.

У начинающих программистов часто возникает недоумение при виде следующих вычислений:

```
>>> num = 20
>>> num = num * 3 # сокращенно: num *= 3
>>> num
60
>>>
```

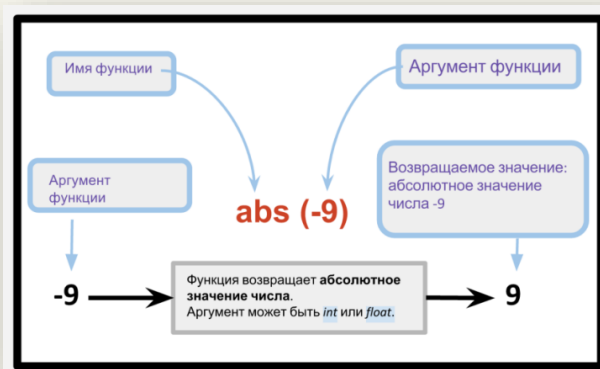
Если вспомнить, что сначала вычисляется правая часть, то все легко объясняется.

Функция в Python является основой при написании программ. С чем можно сравнить функцию? Напрашивается аналогия с «черным ящиком», когда мы знаем, что поступает на вход и что при этом получается на выходе, а внутренности «черного ящика» часто бывают от нас скрыты. Примером является банкомат.

На вход банкомата поступает пластиковая карточка (пин-код, денежная сумма), на выходе мы ожидаем получить запрашиваемую сумму. Нас не очень сильно интересует принцип работы банкомата до тех пор, пока он работает без сбоев.



Рассмотрим функцию с именем `abs`, принимающую на вход один аргумент – объект числового типа и возвращающую абсолютное значение для этого объекта.



Пример вызова функции `abs` с аргументом `-9` имеет вид:

```
>>> abs(-9)
9
>>> d = 1
>>> n = 3
>>> abs(d - n)
2
```

<sup>10</sup> **Информация для опытных программистов.** Для экономии ресурсов при работе с небольшими целыми значениями Python ссылается на уже существующие в памяти объекты