Ссылки на объекты

Для правильного понимания присваивания в Python всегда сначала читайте правую часть, ту, где объект создается или извлекается. Уже после этого переменная в левой части связывается с объектом — как приклеенная к нему этикетка.

Поскольку переменные — это просто этикетки, ничего не мешает наклеить на объект несколько этикеток. В этом случае образуются *синонимы*.

Выбор между `==` и `is`

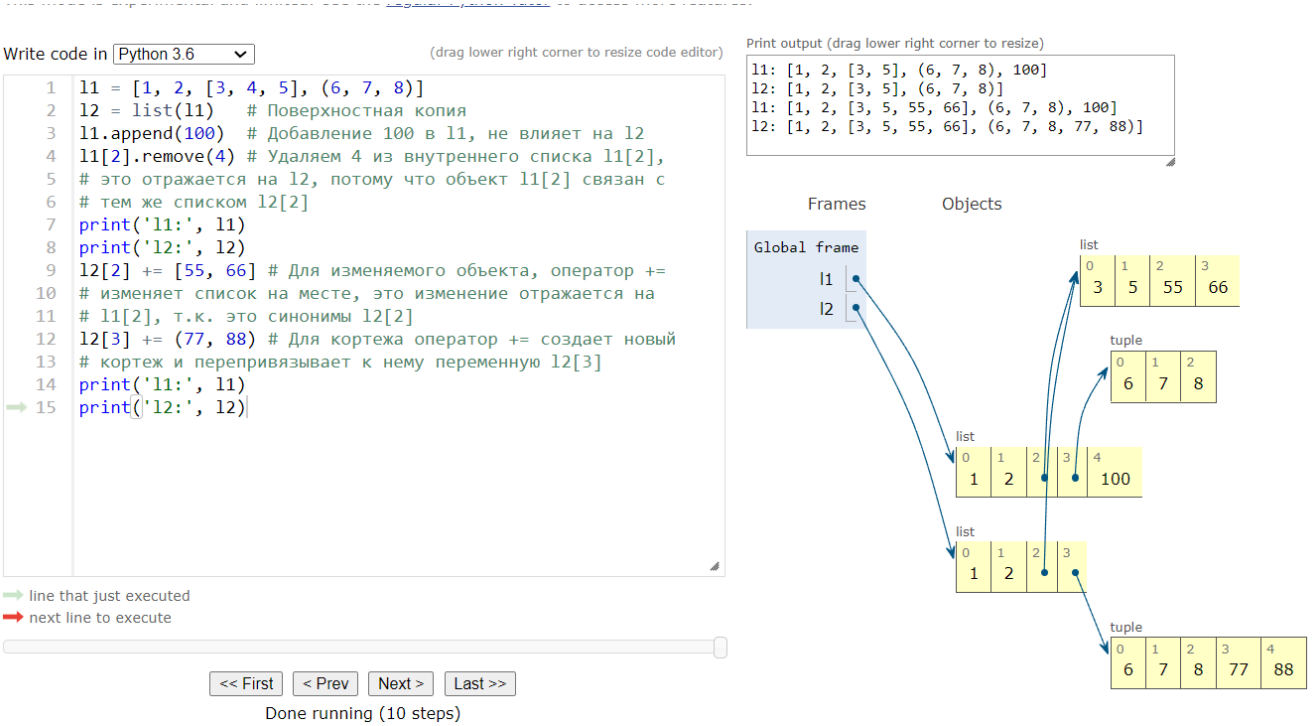
Оператор `==` сравнивает значение объектов (хранящихся в нем данных), а оператор `is` — их `id`.

Относительная неизменность кортежей

Кортежи, как и большинство коллекций в Python, — списки, словари, множества и т.д. — хранят ссылки на объекты. Если элементы, на которые указывают ссылки, изменяемы, то их можно модифицировать, хотя сам кортеж останется неизменяемый.

По умолчанию копирование поверхностное

Создание поверхностной копии списка, содержащий другой список;



Теперь должно быть понятно, что создать поверхностную копию легко, но это не всегда то что нужно.

Для значений по умолчанию (`def func( a=default )`) необходимо устанавливать значение None, а не изменяемые пустые объекты типо `[]`. Это может привести к изменению объекта установленного по умолчанию и во вновь созданных объектах будут фантомные объекты.

Ошибка при назначении по умолчанию изменяемого объекта

Write code in Python 3.6 (drag lower right corner to resize code editor)

```
1  class HauntedBuss:
2      def __init__(self, passangers=[]):
3          self.passangers = passangers
4
5      def pick(self, name):
6          self.passangers.append(name)
7
8      def drop(self, name):
9          self.passangers.remove(name)
10
11 bus1 = HauntedBuss(['Олеся', 'Артём'])
12 print(bus1.passangers)
13 bus1.pick('Антон')
14 bus1.drop('Олеся')
15 print(bus1.passangers)
16 bus2 = HauntedBuss()
17 bus2.pick('Оксана')
18 print(bus2.passangers)
19 bus3 = HauntedBuss()    # Объект, новый но уже
20 # содержит в списке пассажира из bus2
21 print(bus3.passangers)
```

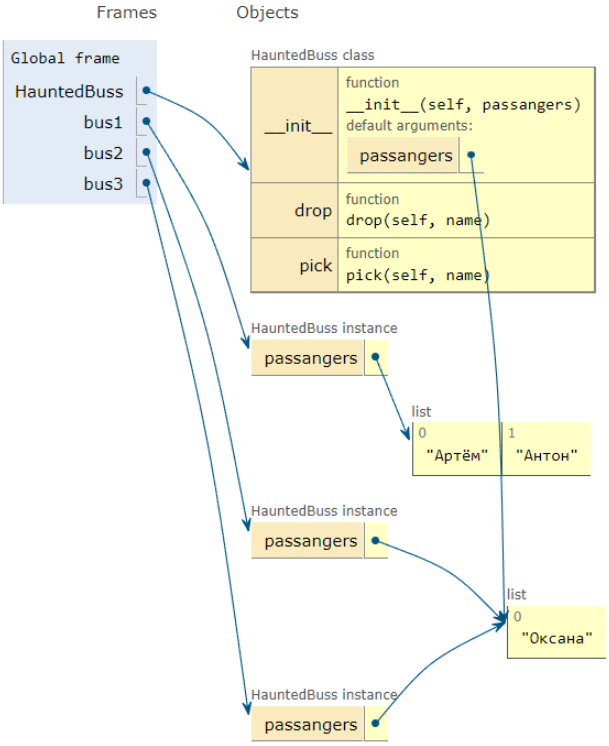
line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Done running (29 steps)

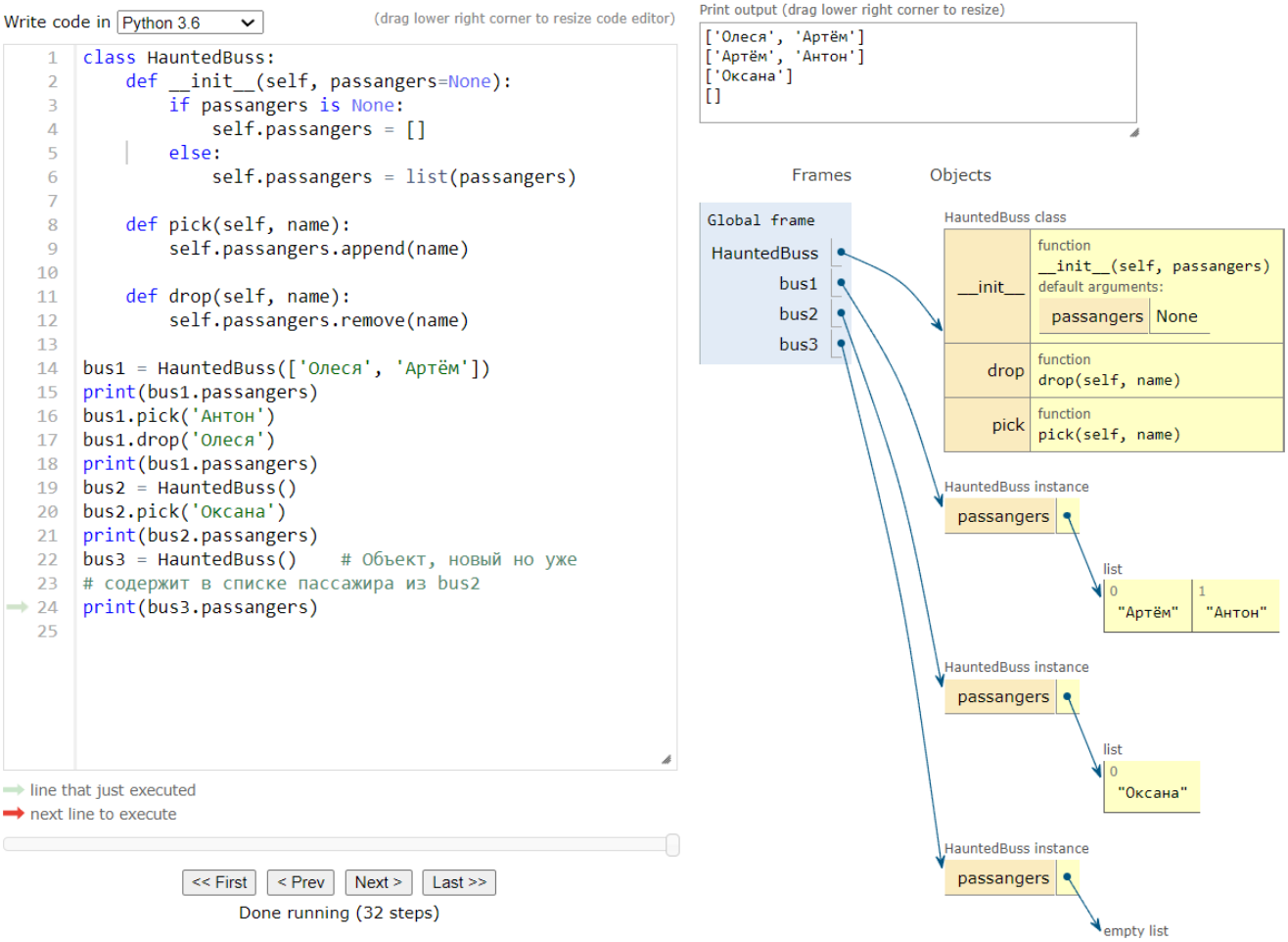
Print output (drag lower right corner to resize)

```
['Олеся', 'Артём']
['Артём', 'Антон']
['Оксана']
['Оксана']
```



При написании функции, принимающий изменяемый параметр, необходимо тщательно обдумывать, ожидает ли принимающая сторона, что аргумент может быть изменён.

Пример правильной реализации



Важное замечание. В объекте автобуса при инициализации создается копия списка. Это исключает ошибки связанной с изменением объекта переданного при инициализации.

```
class HauntedBuss:
    def __init__(self, passangers=None):
        if passangers is None:
            self.passangers = []
        else:
            self.passangers = list(passangers)
            # Если тут сделать просто self.passangers = passangers,
            # эти объекты будут синонимами.

    def pick(self, name):
        self.passangers.append(name)

    def drop(self, name):
        self.passangers.remove(name)
```

doctest для проверки корректного автобуса

```
>>> from source.buss import HauntedBuss
>>> bus1 = HauntedBuss(['Олеся', 'Артём'])
>>> bus1.passangers
['Олеся', 'Артём']
>>> bus1.pick('Антон')
>>> bus1.drop('Олеся')
>>> bus1.passangers
['Артём', 'Антон']
>>> bus2 = HauntedBuss()
>>> bus2.pick('Оксана')
>>> bus2.passangers
['Оксана']
>>> bus3 = HauntedBuss()
>>> bus3.passangers
[]
>>> footboal_team = ['Артем', 'Гриша', 'Пётр', 'Антон', 'Лакки']
>>> bus4 = HauntedBuss(footboal_team)
>>> bus4.passangers
['Артем', 'Гриша', 'Пётр', 'Антон', 'Лакки']
>>> bus4.drop('Антон')
>>> footboal_team
['Артем', 'Гриша', 'Пётр', 'Антон', 'Лакки']
>>> bus4.passangers
['Артем', 'Гриша', 'Пётр', 'Лакки']
```

Если сделать вот так:

```
self.passangers = passangers
```

То футбольная команда потеряет одного игрока вместе с выходом из автобуса.

```
doctest.testfile('./source/doctest/buss_correct.txt')
**********************************************************************
File ".\source\doctest\buss_correct.txt", line 21, in buss_correct.txt
Failed example:
    footboal_team
Expected:
    ['Артем', 'Гриша', 'Пётр', 'Антон', 'Лакки']
Got:
    ['Артем', 'Гриша', 'Пётр', 'Лакки']
**********************************************************************
1 items had failures:
   1 of  17 in buss_correct.txt
***Test Failed*** 1 failures.
TestResults(failed=1, attempted=17)
```