

## Язык программирования Python.

### *Переменные, объекты и ссылки*

Имена переменных могут быть любыми. Однако есть несколько общих правил их написания:

1. Желательно давать переменным осмысленные имена, говорящие о назначении данных, на которые они ссылаются.
2. Имя переменной не должно совпадать с командами языка (зарезервированными ключевыми словами).

Проверить является ли переменная ключевым словом можно так:

Раздаточный материал № 3

```
import keyword
...
print(keyword.iskeyword("NumberInt"))
False
```

3. Имя переменной должно начинаться с буквы или символа подчеркивания (`_`), но не с цифры.
4. Имя переменной не должно содержать пробелы.

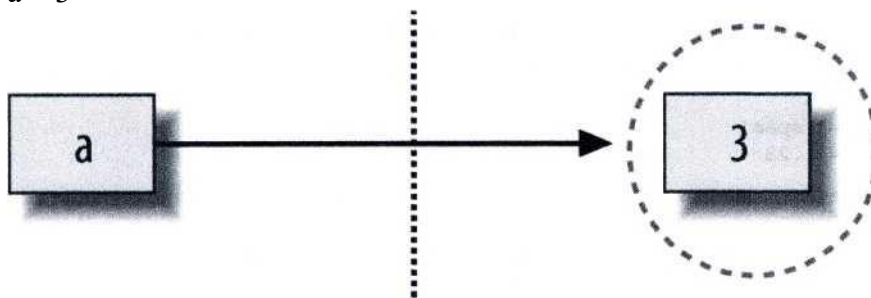
Переменная (или имя) создается при первом присваивании ей значения в коде. Последующие присваивания изменяют значение уже созданного имени. Переменная никогда не располагает какой-либо информацией о типе или о связанных с ним ограничениях. Понятие типа связано с объектом, но не с именем. Переменные всегда просто ссылаются на определенный объект в конкретный момент времени. Когда переменная встречается в выражении, она тотчас же заменяется объектом, на который в текущий момент ссылается. Более того, все переменные должны быть присвоены, прежде чем их можно будет применять; ссылка на не присвоенные переменные приводит к ошибкам.

Важно: необходимо инициализировать имена, прежде чем их можно обновлять. Например, счетчик должен быть инициализирован нулем до того, как его можно будет увеличивать.

Такая модель называется динамической типизацией.

Раздаточный материал № 4

`a = 3`



Переменные всегда ссылаются на объекты и никогда на другие переменные. Такие связи от переменных к объектам в Python называются ссылками. Всякий раз, когда переменная используется в более позднее время (т.е. на нее производится ссылка), Python автоматически следует по связи от переменной к объекту.

Каждый раз, когда генерируется новое значение, Python создает новый объект (т.е. участок памяти) для представления этого значения. Объекты имеют более сложную структуру, чем только пространство, достаточное для представления их значений. Каждый объект также содержит следующие стандартные атрибуты:

- обозначение типа, применяемое для пометки типа объекта,
- счетчик ссылок, используемый для определения, когда можно освободить память, которую занимает объект.
- идентификатор – это уникальный признак объекта, позволяющий отличать объекты друг от друга.

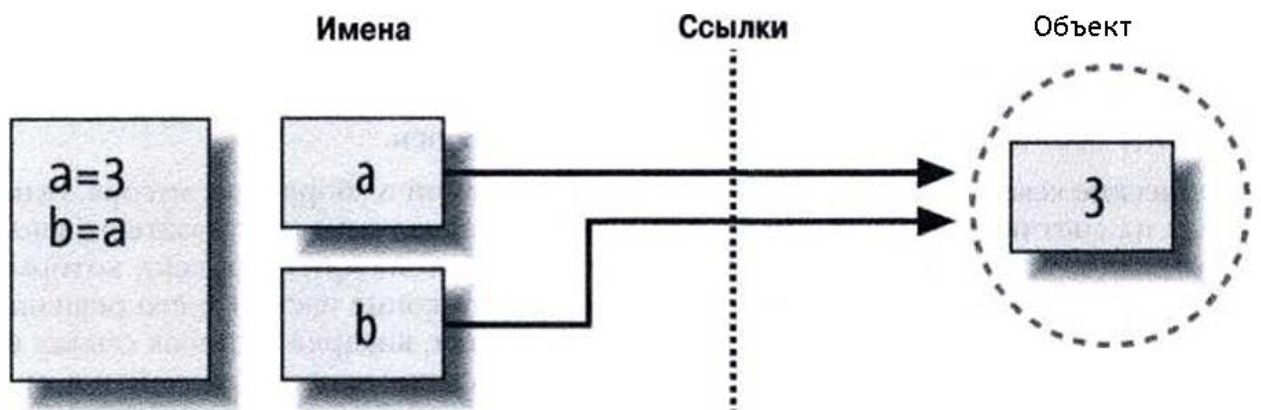
Для того, чтобы посмотреть на объект с каким идентификатором ссылается данная переменная, можно использовать функцию `id()`.

Раздаточный материал № 5

```
#1
print(id(Username))
11083840
```

### Разделяемые ссылки

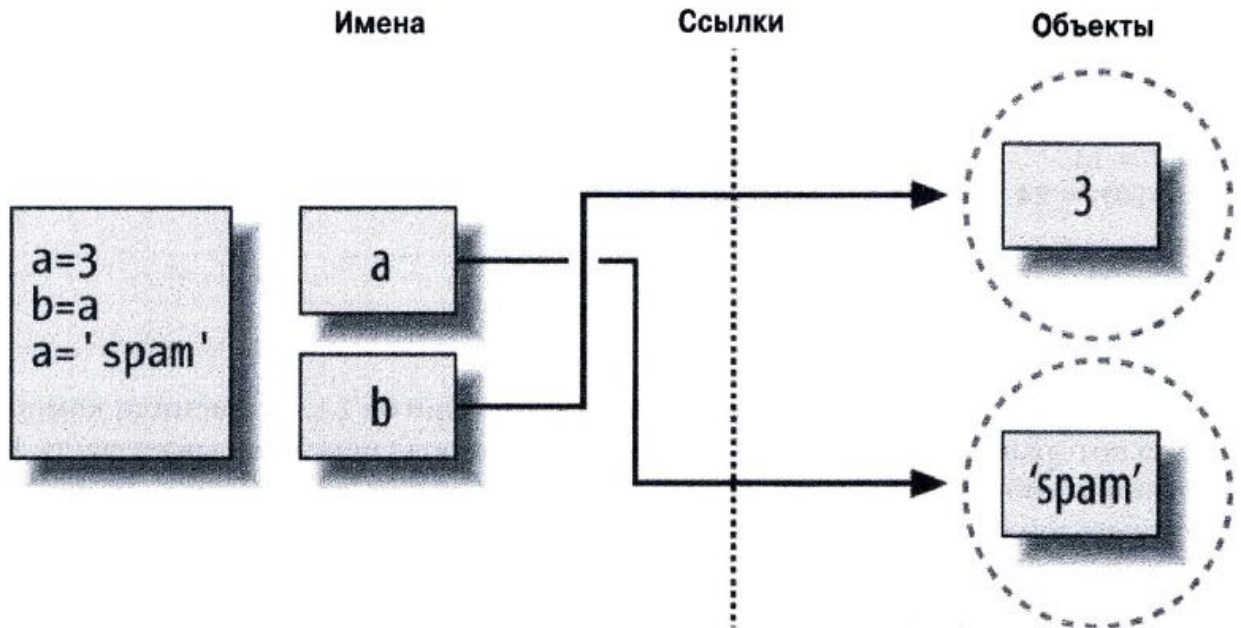
Раздаточный материал № 6



Python создает переменную *b*; переменная *a* здесь используется и не присваивается, поэтому она заменяется объектом, на который ссылается (3), а в *b* обеспечивается ссылка на данный объект. В итоге переменные *a* и *b* ссылаются на тот же самый объект (т.е. указывают на один и тот же участок памяти)

Такой сценарий в Python с множеством имен, ссылающихся на тот же самый объект, обычно называется разделяемой ссылкой (а иногда просто разделяемым объектом). Имена *a* и *b* не связаны друг с другом напрямую; на самом деле в Python вообще нет способа связать одну переменную с другой. Наоборот, обе переменные указывают на один и тот же объект через свои ссылки.

## Раздаточный материал № 7



## Изменяемые и неизменяемые типы.

К неизменяемым (immutable) типам относятся: целые числа (int), числа с плавающей точкой (float), комплексные числа (complex), логические переменные (bool), кортежи (tuple), строки (str) и неизменяемые множества (frozen set).

## Раздаточный материал № 8

```
r = 15
print(id(r)) → 2012981392
r += 5.1
print(id(r)) → 1719872
```

Создан новый объект с новым ID.

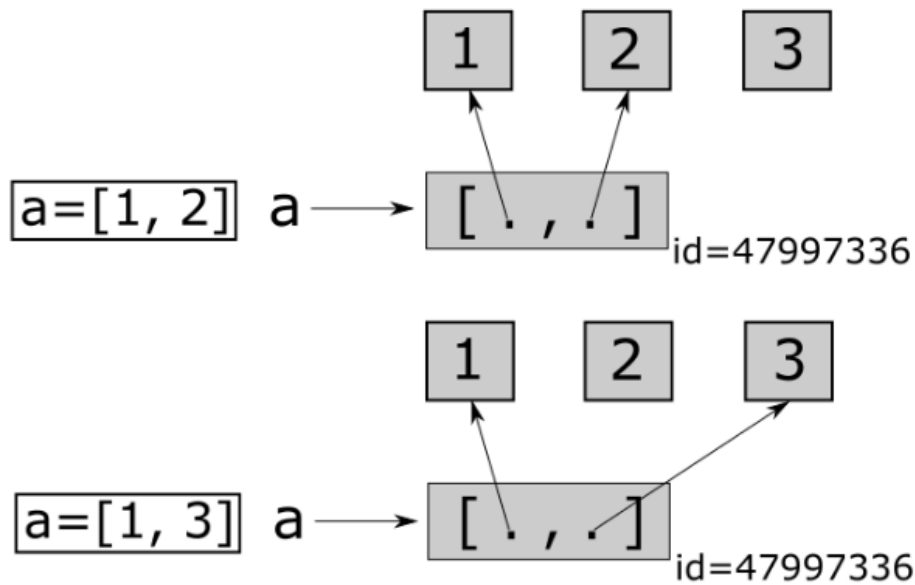
К изменяемым (mutable) типам относятся: списки (list), множества (set), словари (dict). Если тип данных изменяемый, то можно менять значение объекта. Например,

## Раздаточный материал № 9

создадим список `[1, 2]`, а потом заменим второй элемент на `3`.

```
>>> a = [1, 2]
>>> id(a)
47997336
>>> a[1] = 3
>>> a
[1, 3]
>>> id(a)
47997336
```

Объект, на который ссылается переменная `a`, был изменен. Это можно проиллюстрировать следующим рисунком.



В рассмотренном случае, в качестве данных списка, выступают не объекты, а отношения между объектами. Т.е. в переменной `a` хранятся ссылки на объекты содержащие числа 1 и 3, а не непосредственно сами эти числа.

Фронтальный опрос:

- ✓ Правила составления переменных.
- ✓ Динамическая типизация в Python.
- ✓ Понятие ссылки в Python
- ✓ Атрибуты объекта в Python
- ✓ Понятие разделяемой ссылки в Python.
- ✓ Перечислить изменяемые типы.
- ✓ Перечислить неизменяемые типы.