Лекция 6
Изменяемые и неизменяемые типы

Изменяемые и неизменяемые типы в Python

Все переменные в Python делятся на два типа: изменяемые и неизменяемые.

Изменяемый тип переменных - связан с тем, что, как только вы меняете значение переменной, то в памяти создается новый объект, в котором и будет храниться новое значение вашей переменной.

Неизменяемый тип переменных — при изменении значения в памяти нового объекта не создается, но меняются свойства уже существующего.

Как работают переменные в Python

Для того, чтобы понять идею изменяемых и неизменяемых типов, рассмотрим, как работают переменные в Python.

Когда вы создаете переменную, то вы по сути создаете ссылку на область оперативной памяти, в которой хранится объект. Именно внутри этого объекта и хранится информация о типе переменной, о значении и всем наборе свойств, связанных с этим объектом.

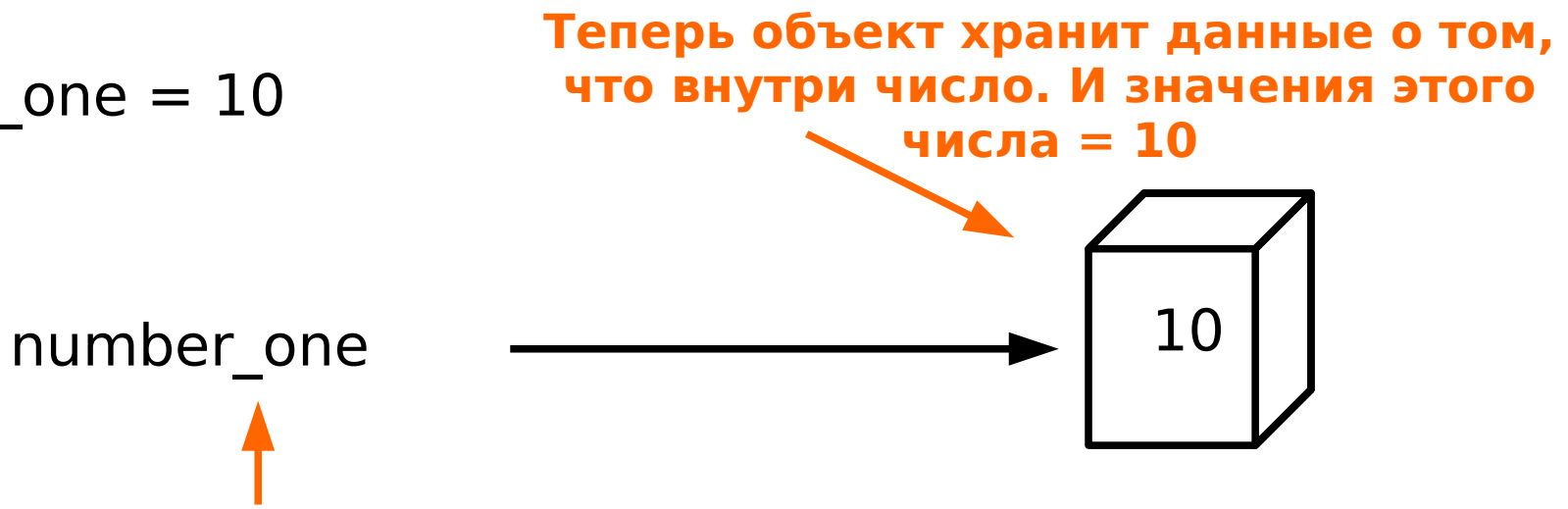
Т.е. тип переменной в Python определяется типом объекта, а не типом ссылки.

python

Как устроены переменные в Python

Предположим, что вы объявили обычную переменную. Например:

```
number_one = 10
```

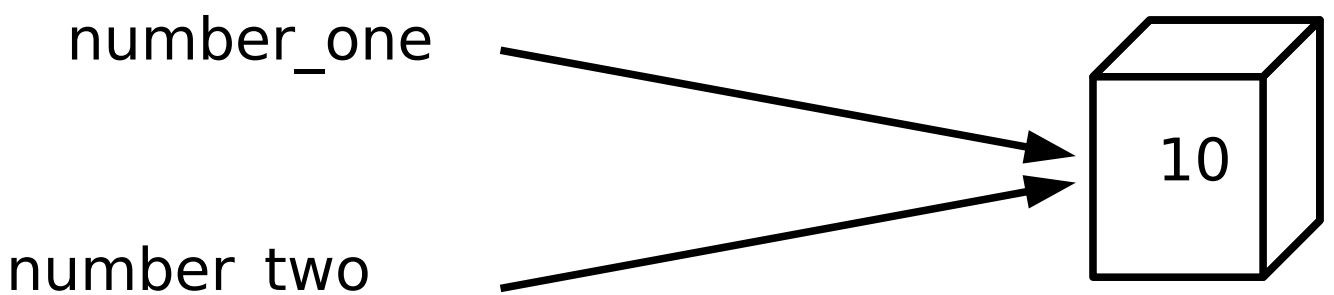


python

Что происходит, когда вы присваиваете одной переменной значение другой

Предположим, что вы присвоили одной переменной значение другой. Например:

```
number_one = 10
number_two = number_one
```

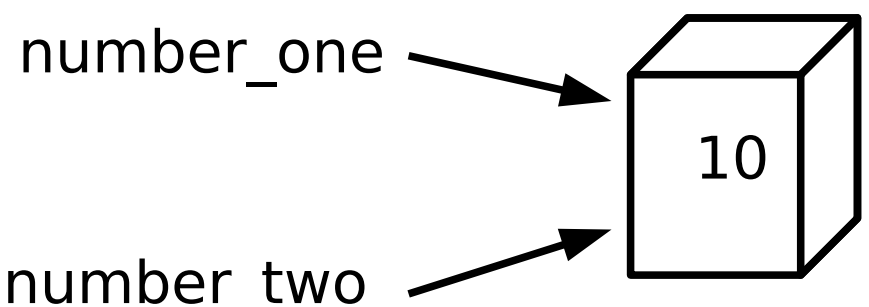


В таком случае вы получаете две ссылки на один и тот же объект.

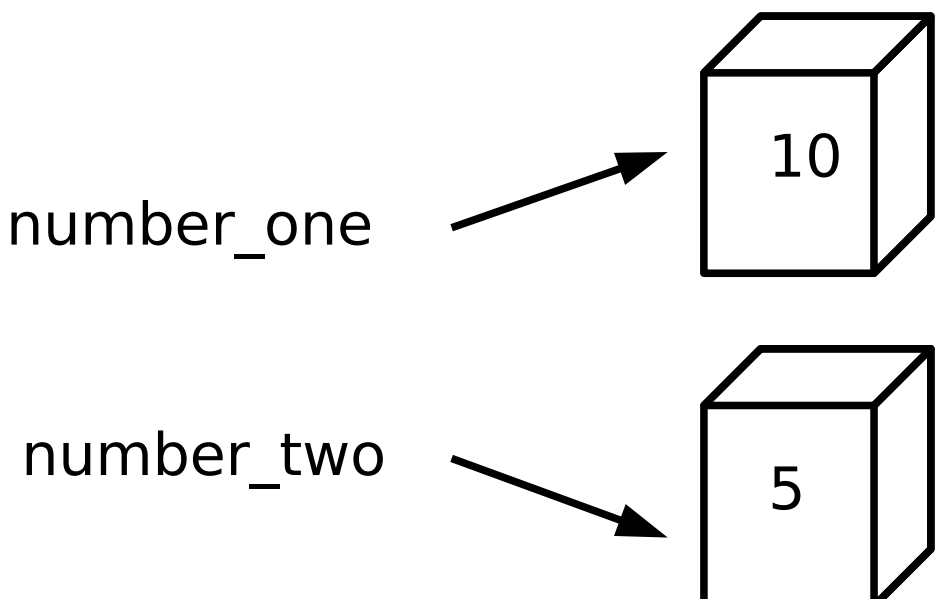
Работа с переменными неизменяемого типа

Теперь попробуем после присвоения одной переменной значения другой, изменить значение другой. Как это сработает для переменных неизменяемого типа (например, для чисел)?

```
number_one = 10
number_two = number_one
number_two = 5
```



До изменения второй переменной



После изменения второй переменной

Работа с переменными неизменяемого типа

Таким образом, при попытке изменения значения переменной неизменяемого типа, в памяти **создается новый объект**, и ссылка уже указывает на него.

Из-за этого, после того, как вы присвоите одной переменной значения другой, вы тут же получите две ссылки на один объект. Если после этого вы измените значение одной из этих переменных, то получите две ссылки на два разных объекта.

Как следствие, изменение значения одной переменной никак **не влияет** на значение другой.

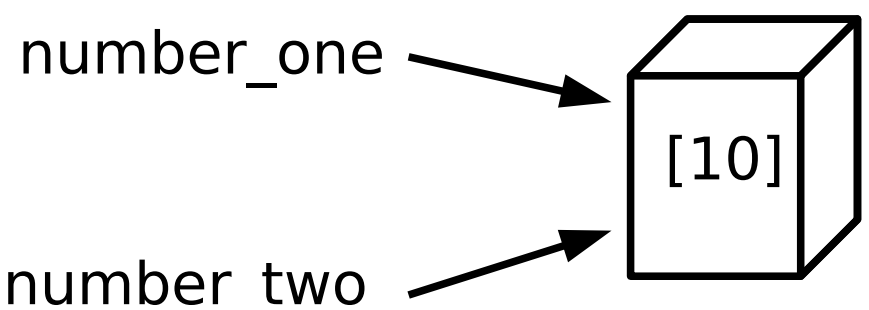
К переменным неизменяемого типа относят числа, строки, кортежи и т.д.

python

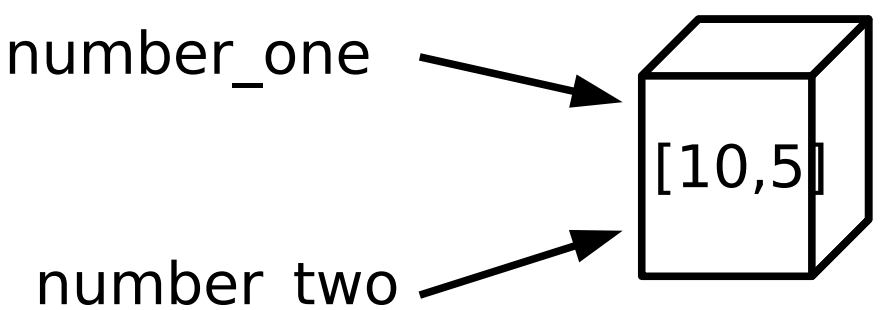
Работа с переменными изменяемого типа

Теперь попробуем после присвоения одной переменной значения другой, изменить значение другой. Как это сработает для переменных изменяемого типа (например, для списков)?

```
number_one = [10]
number_two = number_one
number_two.append(5)
```



До изменения второй переменной



После изменения второй переменной

Работа с переменными изменяемого типа

Таким образом, при попытке изменения значения переменной изменяемого типа, **новый объект не создается**, а просто меняется значение основного.

Из-за этого, после того, как вы присвоите одной переменной значения другой, вы опять, как и в случае с переменными неизменяемого типа, получите две ссылки на один объект. Но, если после этого вы измените значение одной из этих переменных, то вторая переменная также будет указывать на тот же измененный объект. **Как следствие, изменение значения одной переменной влияет на значение другой**.

К переменным изменяемого типа относят списки, словари, и т.д.

Список использованной литературы

1) Лутц М. Изучаем Python, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011 - 194 - 205 с.
2) Саммерфилд М. Программирование на Python 3. Подробное руководство. - Пер. с англ. - СПб.:Символ-Плюс, 2009. - 29 - 33 с.