



Лекция 2

Переменные

KIEV 2019



Определение переменной

Переменная - поименованная, либо адресуемая иным способом область памяти, адрес которой можно использовать для осуществления доступа к данным.

Данные, находящиеся в переменной (то есть по данному адресу памяти), называются **значением** этой переменной.

Если тип данных определяется на этапе компиляции, имеет место **статическая типизация**, а если на этапе выполнения программы - **динамическая**.

В Python используется динамическая типизация!!!




Как объявить переменную в Python


Для того чтобы объявить переменную, сначала нужно придумать для нее имя (главное, чтобы не совпадало с зарезервированными словами Python). После этого поставить знак « = » и указать значение этой переменной.

Например:

`my_age = 35`



Имя переменной



Значение



Особенности объявления переменных в Python

Сколько переменных я могу объявить?

Вы можете объявить произвольное количество переменных (в пределах размера ОЗУ). Главное придумать им имена и присвоить начальные значения.

Могу ли я использовать любые имена (кроме зарезервированных)?

Да, любое имя, которое придет вам в голову.

Есть ли рекомендации по написанию имен переменных?

Да есть - это официальная рекомендация **PEP 8 -- Style Guide for Python Code**

Как называется такой стиль оформления переменных?

Такой стиль для именования переменных называется «Snake case». Подробнее можно прочесть например тут: https://ru.wikipedia.org/wiki/Snake_case



Как вывести значение переменной на экран

Для этого достаточно написать `print(имя переменной)`.

```
my_age = 36  
print(my_age)
```

Объявление переменной



```
my_name = "Alexander"  
print(my_name)
```



Вывод ее на экран



Список встроенных типов в Python

Type	Тип	Пример
Number	Число	123, 0.5, 1+2j, Decimal(), Fraction()
String	Строка	"Hello world" , 'Hello'
List	Список	[1,4,5]
Dictionary	Словарь	{1:"one", 2:"two"}
Tuple	Кортеж	(1,4,5)
File	Файл	open("a.txt")
Set	Множество	{1,4,5,6,7}
....	И другие..	



Как узнать тип переменной в Python

У меня есть переменная, но я пока не понимаю какого она типа. Как его узнать?

Для того чтобы узнать тип переменной, используется оператор `type`. Например, если вы хотите вывести тип переменной на экран, то наберите:

```
print( type( variable_name))
```



Сюда подставьте имя переменной, тип которой хотите узнать.



Как узнать тип переменной в Python (пример кода)

```
my_age = 35
```

```
print(type(my_age))
```

```
my_name = "Alexander"
```

```
print(type(my_name))
```

Результат работы программы

```
<type 'int'>
```

```
<type 'str'>
```

Переменная с именем `my_age` — это число, а с именем `my_name` — это строка.



Числовые типы данных в Python

Целые числа — числа, используемые для представления штучного количества предметов. Т.е. у них нет дробной части. В Python представлены типом **int**. Могут содержать произвольное количество знаков (в пределах размера ОЗУ).

Как задать переменную для хранения целого числа?

Для этого придумайте имя переменной. После имени переменной поставьте знак = и укажите числовое значение.

Внимание! Дробной части быть не должно.

Например:

```
my_age = 35
```



Основные арифметические действия с числами в Python

Вид оператора	Его математический эквивалент
+	Операция сложения
-	Операция вычитания
*	Операция умножения
/	Операция деления
//	Деление с округлением до минимального целого
%	Остаток от деления
()	Изменение приоритета операций
**	Возведение в степень

Приоритет арифметических операторов в Python такой же, как и в математике!!!



Некоторые пояснения к арифметическим операторам

Что такое деление с округлением до минимального целого?

Это означает, что сначала вы должны выполнить обычное деление, и полученный результат округлить к ближайшему минимальному целому числу.

Например:

$5/2 = 2.5$ Ближайшее минимальное целое число равно 2.

Следовательно:

$$5//2 = 2$$

$5/(-2) = -2.5$ Ближайшее минимальное целое равно -3.

Следовательно:

$$5//(-2) = -3$$



Пример использования целых чисел в Python

```
a = 5  
b = 10  
c = a + b * 3  
print(c)
```

Описание работы примера:

В первой и второй строке объявлены две переменных (обе целочисленные). В третьей строке объявлена переменная *c*, значение которой вычисляется на основании значений переменных *a* и *b*.

Важно!!! Все переменные должны быть объявлены до того, как они используются.

Переменная *c* в этом случае равна 35. И в четвертой строке ее значение выводится на экран.



Интересные аспекты при присвоении переменных в Python

Что означает такой пример кода? Интересует именно вторая строка.

```
my_age = 35  
my_age = my_age + 1  
print(my_age)
```

В случае, если имя переменной встречается и справа, и слева от оператора равно, это означает, что новое значение переменной (слева от оператора =) вычисляется на основании старого значения (справа от оператора =).

Так, во второй строке новое значение переменной `my_age` вычисляется на основании старого значения этой переменной, к которому прибавили 1. Так как старое значение равнялось 35, то новое значение стало равно 36.



Вещественные числа в Python

Вещественные числа используются для указания непрерывных по своей природе величин. Т.е. содержащие дробную часть. В Python представлены типом **float**.

Этот тип в Python имеет ограниченный размер. Диапазон возможных значений $[-1.8E308, 1.8E308]$.

Также нужно знать, что эти переменные обладают точностью только до 16 знаков после запятой.

Для того, чтобы задать переменную вещественного типа, укажите ее имя, поставьте знак `=` и значение вещественного типа.

Внимание! Для разделения целой и дробной части используется точка. Например: `orange_price = 15.75`



Выражения содержащие и целые, и вещественные числа

Каков будет тип результата, если справа от оператора = в выражении используются целые и вещественные числа одновременно? ?

Если справа от оператора = в выражении используются и целые, и вещественные числа, то результирующим типом будет вещественное число.

Например:

```
orange_price = 15.75
```

```
number_of_orange = 12
```

```
total_price = orange_price * number_of_orange
```

```
print(total_price)
```

```
print(type(total_price))
```



Дополнительные математические методы из модуля `math`

Метод	Его математический эквивалент
<code>math.fabs(x)</code>	Модуль x
<code>math.sqrt(x)</code>	Квадратный корень из x
<code>math.cos(x)</code>	Косинус x
<code>math.sin(x)</code>	Синус x
<code>math.exp(x)</code>	Экспонента x
<code>math.log(x, base)</code>	Логарифм x , по основанию <code>base</code> . Если <code>base</code> не указано, тогда натуральный
<code>math.pow(x, y)</code>	Возведение x в степень y
<code>math.pi</code>	Значение числа π

И много других математических функций в этом модуле.

Перед использованием, его необходимо импортировать.



Импорт дополнительных модулей в Python

Для подключения дополнительных модулей в свое приложение, их необходимо импортировать.

Выполняется это с помощью команды **import**, за которой следует имя модуля, который вы хотите импортировать. Выполнять импорт дополнительных модулей следует в самом начале вашего приложения.

Например, импорт модуля `math` выглядел бы так:

```
import math
```



Пример использования дополнительных математических функций модуля **math**

Импорт модуля math

`import math`

Число π

`x = math.sqrt(7) * math.cos(math.pi * 3/4)`

Квадратный корень из 7 Вычисление косинуса

`print(x)`



Комплексные числа (complex) в Python

Комплексные числа используются для указания комплексных величин. Используются в основном в наукоемких приложениях. В Python представлены типом `complex`.

Этот тип в Python имеет ограниченный размер. Диапазон возможных значений целой и мнимой части `[-1.8E308, 1.8E308]`.

Также нужно знать, что эти переменные обладают точностью только до 16 знаков после запятой.

Для того, чтобы задать переменную комплексного типа, укажите ее имя, поставьте знак `=` и напишите слово `complex`(действительная часть, мнимая часть).

Например: `x = complex(2,3)`



Преобразование типов чисел в Python

У меня есть вещественное число, а я хочу сделать из него целое. Это возможно ?

Да. Это называется преобразованием типа. Для этого необходимо написать тип данных, к которому вы хотите привести ваше число, потом открыть круглые скобки и в них вставить значение вашего числа, которое вы хотите привести к вашему типу данных.

Например:

Какое значение хочу привести

```
x = int(7.5)
```

К какому типу хочу привести



Ввод данных с клавиатуры

Иногда значение переменной нужно считать с клавиатуры. Для этого используется метод **`input("Сообщение пользователю")`**.

Правда, результатом считывания будет строка.

Для преобразования к требуемому типу значения нужно использовать оператор преобразования типов.

Как считать с клавиатуры:

Целое число -	<code>x = int(input("Input some number"))</code>
Вещественное число -	<code>x = float(input("Input some number"))</code>
Строку -	<code>x = input("Input some text")</code>



Пример ввода данных с клавиатуры

```
a = int(input("Input a"))  
b = int(input("Input b"))
```

← Считывание целого числа с клавиатуры
← Считывание целого числа с клавиатуры

```
sum_1 = a + b  
difference_1 = a - b  
product_1 = a * b  
quotient_1 = a / b
```

← Арифметические действия над введенными переменными

```
print("a+b =", sum_1)  
print("a-b =", difference_1)  
print("a*b =", product_1)  
print("a/b =", quotient_1)
```

← Вывод результата на экран



Список использованной литературы

- 1) Лутц М. Изучаем Python, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. - 121-126 с.
- 2) Саммерфилд М. Программирование на Python 3. Подробное руководство. - Пер. с англ. - СПб.:Символ-Плюс, 2009. - 69-76 с.



Домашнее задание

1) Написать программу, которая считывает 5-тизначное число с клавиатуры и выводит цифры, из которого оно СОСТОИТ.

Например : Считывается число 54698

Выводится:

5
4
6
9
8

2) Написать программу, которая вычислит и выведет на экран площадь треугольника, если известны его стороны.

3) Написать программу, которая вычислит и выведет на экран длину окружности, если ее радиус считывается с клавиатуры.