



Python Basics

- **Компьютерная программа**

- – это набор инструкций, выполняемых вычислительной техникой (компьютерами). В большинстве случаев программы могут быть представлены в виде объектного (бинарного) и исходного кода.

- **Для пользователя одним из основных отличий является характер программы:**

- **Прикладные программы**

- — это программы, с которыми может взаимодействовать непосредственно пользователь. К ним относятся «текстовые редакторы», «браузер», «веб-сайты», «игры».

- **Системные программы**

- – это программы, которые обеспечивают взаимодействие пользователя с компьютером и создают среду, в которой выполняются прикладные программы. К ним можно отнести «драйвера» и «операционные системы».

- Синонимами слова программа являются: «приложение», «ПО», «программное обеспечение», «software». Таким образом, компьютер – это совокупность «программного» и «аппаратного» обеспечения.

Преимущества Python

- Интерпретируемый
- Динамически типизированный
- Объектно-ориентированный
- Лаконичный и простой
- Имеет большое сообщество

Как работает Python?

Интерпретатор

Интерпретатор - это такой модуль (программа), которая выполняет другие программы.

Когда вы пишете программу на языке Python, интерпретатор читает вашу программу и выполняет содержащиеся в ней инструкции.

В действительности, интерпретатор - это слой программной логики между вашим программным кодом и аппаратурой вашего компьютера.

Выполнение программы

Что стоит за словами «**написать и запустить программу на языке Python**» зависит от того, как вы смотрите на эту

задачу – как **программист** или как **интерпретатор Python**. Обе точки зрения определяют свой взгляд на программирование.

С точки зрения программиста

Программа на языке Python, в самой простой форме, – это обычный текстовый файл, содержащий инструкции Python.

Например, следующий файл, с именем script0.py, – это один из простейших сценариев на языке Python, который только можно придумать, но его официально можно назвать программой на языке Python:

```
print('hello world')
```

Типы Данных

- В Python все объекты делятся на два вида:

Изменяемые (mutable)

Списки(`list`), Словари(`dict`),
Множества(`set`)

Неизменяемые (immutable)

Числа(`int`),
Числа с плавающей точкой
(`float`),
Строки(`str`), Кортежи(`tuple`),
Логические переменные(`bool`)
и все остальные...

Изменяемые (mutable)

Списки(list) – тип данных, предназначенный для хранения набора или последовательности разных элементов.
Список в Python – это массив указателей на элементы, размещенные в памяти

```
[1, 33, 6, 9] # литерал списка в Python
```

Словари(dict) – это тип данных, представляющий собой неупорядоченный набор пар ключ:значение.
(при этом каждый ключ, в рамках одного словаря, является уникальным).

```
# литерал словаря в Python, где first_key и second_key – ключи,  
# а 1 и 2 – соответственно ассоциированные с ними значения
```

```
{'first_key': 1, 'second_key': 2}
```

Множества(set) – интуитивно понятный математический термин, который часто используется в **обыденной речи и означает набор или совокупность неких элементов, что обладают каким-то общим свойством.**

```
# множество натуральных чисел от 1 до 10  
natural num set = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

```
# множество персонажей Братства Кольца  
the fellowship of the ring set = {'Гэндальф', 'Арагорн', 'Фродо', 'Сэм', 'Боромир', 'Леголас', 'Гимли', 'Мерри',  
'Пиппин'}
```

```
# множество приближений math.sqrt(2)  
sqrt approximation set = {1.4142135623, 1.414213562, 1.41421356, 1.4142135, 1.414213}
```

```
# множество результатов какого-то голосования  
vote result set = {'P', 76.69}, ('G', 11.77), ('J', 5.65), ('S', 1.68), ('Y', 1.05), ('T', 0.76)}
```

Неизменяемые (immutable)

Числа(int). Любое целое число состоит из массива цифр переменной длины, поэтому в Python 3 в переменную типа `int` может быть записано число неограниченной длины. Единственное ограничение длины – это размер оперативной памяти.

```
>>> 134523345234252523523478777 ** 2
18096530413013891133013347014216107772438771969415729
```

Вещественные числа(float). Еще такие числа называют числами с плавающей точкой. Это числа, содержащие точку (десятичный разделитель) или знак экспоненты.

Числа типа float – неточны (из-за представления чисел с плавающей запятой в компьютере).

```
>>> 0.3 + 0.3 + 0.3 + 0.1
0.9999999999999999
```

**Строки(str), Кортежи(tuple), Логические переменные(bool)
и все остальные...**

Операторы

Оператор	Описание	Примеры	Оператор	Описание	Примеры
+	Сложение - Суммирует значения слева и справа от оператора	15 + 5 в результате будет 20 20 + -3 в результате будет 17 13.4 + 7 в результате будет 20.4	==	Проверяет равны ли оба операнда. Если ДА , то условие становится истинным.	5 == 5 в результате будет True True == False в результате будет False "hello" == "hello" в результате будет True
-	Вычитание - Вычитает правый операнд из левого	15 - 5 в результате будет 10 20 - -3 в результате будет 23 13.4 - 7 в результате будет 6.4	!=	Проверяет равны ли оба операнда. Если НЕТ , то условие становится истинным.	12 != 5 в результате будет True False != False в результате будет False "hi" != "Hi" в результате будет True
*	Умножение - Перемножает операнды	5 * 5 в результате будет 25 7 * 3.2 в результате будет 22.4 -3 * 12 в результате будет -36	<>	Проверяет равны ли оба операнда. Если нет, то условие становится истинным.	12 <> 5 в результате будет True. Похоже на оператор !=
/	Деление - Делит левый операнд на правый	15 / 5 в результате будет 3 5 / 2 в результате будет 2 (В Python 2.x версии при делении двух целых чисел результат будет целое число) 5.0 / 2 в результате будет 2.5 (Чтобы получить "правильный" результат хотя бы один операнд должен быть float)	>	Проверяет больше ли значение левого операнда, чем значение правого. Если да, то условие становится истинным.	5 > 2 в результате будет True. True > False в результате будет True. "A" > "B" в результате будет False.
%	Деление по модулю - Делит левый операнд на правый и возвращает остаток.	6 % 2 в результате будет 0 7 % 2 в результате будет 1 13.2 % 5 в результате 3.2	<	Проверяет меньше ли значение левого операнда, чем значение правого. Если да, то условие становится истинным.	3 < 5 в результате будет True. True < False в результате будет False. "A" < "B" в результате будет True.
**	Возведение в степень - возводит левый операнд в степень правого	5 ** 2 в результате будет 25 2 ** 3 в результате будет 8 -3 ** 2 в результате будет -9	>=	Проверяет больше или равно значение левого операнда, чем значение правого. Если да, то условие становится истинным.	1 >= 1 в результате будет True. 23 >= 3.2 в результате будет True. "C" >= "D" в результате будет False.
//	Целочисленное деление - Деление в котором возвращается только целая часть результата. Часть после запятой отбрасывается.	12 // 5 в результате будет 2 4 // 3 в результате будет 1 25 // 6 в результате будет 4	<=	Проверяет меньше или равно значение левого операнда, чем значение правого. Если да, то условие становится истинным.	4 <= 5 в результате будет True. 0 <= 0.0 в результате будет True. -0.001 <= -36 в результате будет False.