



Основы Python



Олег Булыгин



Олег Булыгин

- Преподаватель на курсах “Основы языка программирования Python”, “Продвинутый Python”, “Python для анализа данных” в Нетологии
- IT-аудитор в ПАО “Сбербанк”

О чём мы поговорим сегодня

3

1. Почему Python?
2. Интерактивная оболочка Jupyter Notebook
3. Переменные
4. Логические операторы
5. Условные конструкции

Почему Python?

Python 3

5

Python – интерпретируемый язык с динамической типизацией.

Плюсы Python:

- полностью open-source;
- расширяемость за счет множества библиотек (в т.ч. для анализа данных и машинного обучения);
- очень широкая сфера применения;
- приятный синтаксис.

Сфера применения:

- анализ данных и машинное обучение;
- разработка веб-приложений (Django);
- системное администрирование;
- разработка десктопных приложений;
- gamedev.



К слову, Python — это не змея

это комик-группа из Великобритании

Как вывести фразу “Hello world” на экран

7

```
print("Hello world!")
```

Как вывести фразу “Hello world” на экран

8

```
print("Hello world!")
```

JAVA

```
class HelloWorldApp {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```


Как вывести фразу “Hello world” на экран

9

```
print("Hello world!")
```

C

```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    printf("Hello, world!\n");
    return EXIT_SUCCESS;
}
```

Как вывести фразу “Hello world” на экран

10

```
print("Hello world!")
```

Brainfuck

```
>+++++++[<++++++>-]<.>+++++++[<++++>-]  
<+.+++++++..+++. [-]>+++++++[<++++>-]  
<.>+++++++[<++++>-]  
<. >+++++++[<+++>-]<.+..-----.-. [-]>  
+++++++[<++++>-]<+. [-]+++++++>  
+++++++[<++++++>-]<.>+++++++[<++++>-]  
<+.+++++++..+++. [-]>+++++++[<++++>-]  
<.#>+++++++[<++++>-]<. >+++++++  
[<+++>-]<.+..-----.-. [-]>+++++++[<++++>-]  
<+. [-]+++++++.
```

Арифметические операции

- сложение (+)
- вычитание (-)
- умножение (*)
- деление (/)
- целочисленное деление (//)
- возведение в степень (**)
- взятие остатка от деления (%)

Приоритет операции аналогичен стандартным математическим правилам, можно использовать скобки.

Комментарии

это текст, который присутствует в коде программы, но игнорируется интерпретатором. Используются для того, чтобы добавить объяснение для определенного блока кода.

Написание комментария начинается с символа #.

Переменные

Переменная – это объект, которому дано имя. Необходимы, в том числе, для хранения данных и промежуточных результатов вычислений.

Объект – это:

- число;
- строка;
- практически, что угодно в Python.

$$a = 10 + 20$$

$$b = a * 30$$

$$c = a / b$$



Именованние переменных

14

имя переменной может состоять только из цифр, букв и знаков подчеркивания

имя переменной не может начинаться с цифры

имя переменной должно описывать ее суть

лучше использовать `snake_case` (слова писать со строчной буквы и разделять нижним подчеркиванием)



Помните: Python – регистрозависимый язык

Операторы сравнения

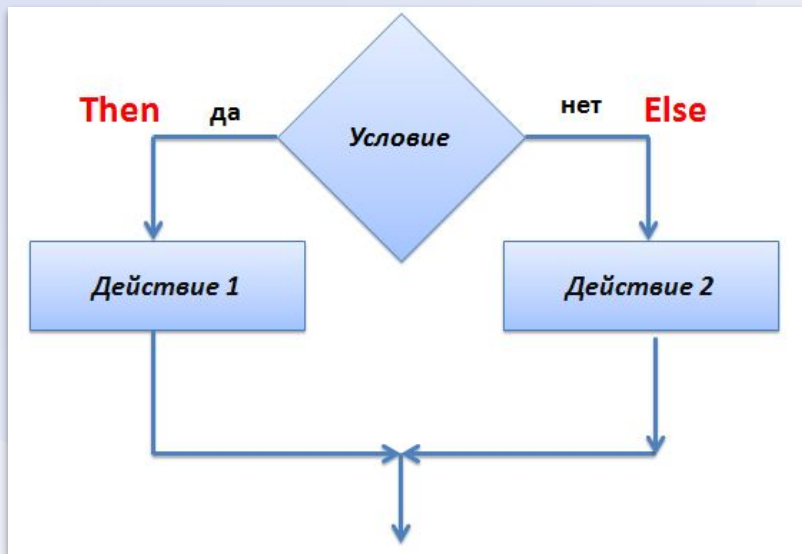
- >
- <
- == (не путать с =)
- >=
- <=
- !=

В результате операций сравнения возвращается булево значение (True / False).

Сравнения могут быть записаны в цепочку.

Условные конструкции

16

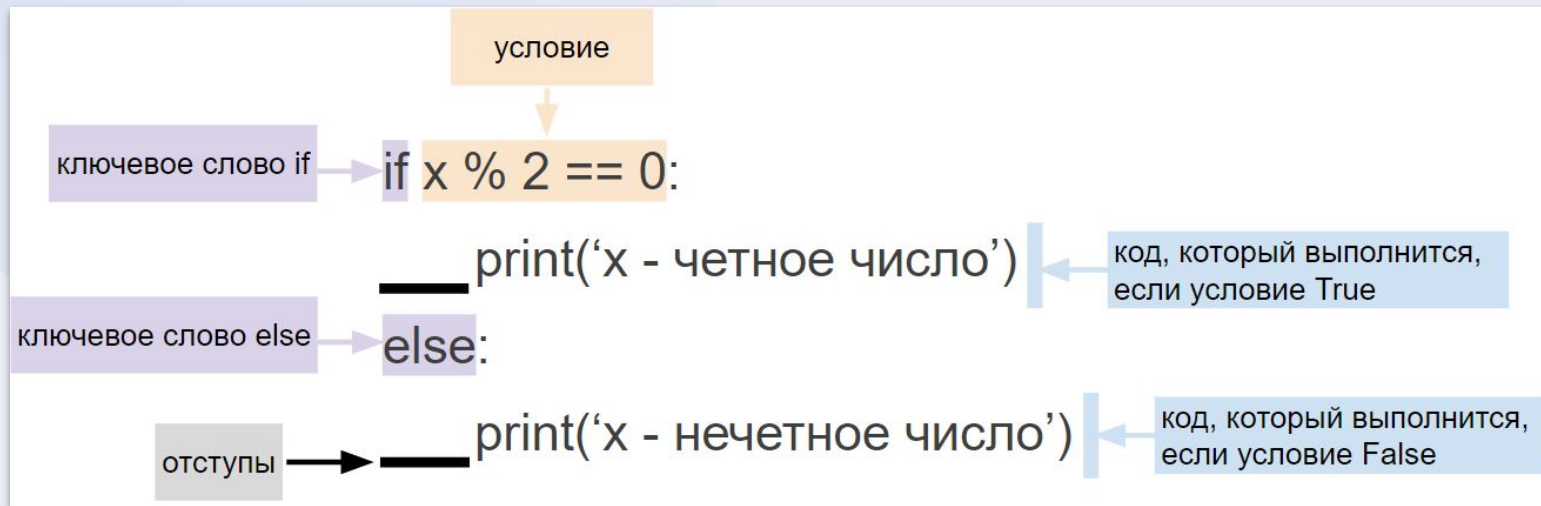


- это способ управлять выполнением программы;
- это способ запрограммировать принятие решений;
- это логическое выражение после которого пишутся команды, которые выполняются, если условие истинно.

01 if

02 elif

03 else



Логические операторы

18

01

AND

логическое И – возвращает True, только когда оба операнда True

02

OR

логическое ИЛИ – возвращает True, когда хотя бы один операнд True

03

NOT

логическое НЕ – возвращает булево значение, противоположное операнду

```
should_notify = account_amout \  
< loan_amount and \  
((days_to_pay < 10 \  
and not is_holiday) \  
or (days_to_pay == 15))
```

Оповещаем человека о необходимости вернуть кредит если:

- на его счету меньше денег, чем нужно для оплаты кредита
- и дней до оплаты кредита меньше 10, но сейчас не выходной
- или ровно за 15 дней до оплаты

- логические значения можно записывать в переменные;
- в логических выражениях можно использовать скобки;
- логические выражения могут быть сложными.

TRUE

Любое ненулевое число

TRUE

Любая непустая строка

FALSE

0, "", [], {}, (), False



Основы Python

Вопросы?

Олег Булыгин

Соцсеть
fb.com/obulygin91

Почта
obulygin91@ya.ru