

Основы Python

Олег Булыгин

Нетология



Булыгин Олег

- Преподаватель на курсах “Основы языка программирования Python”, “Продвинутый Python”, “Python для анализа данных” в Нетологии
- Начальник бюро планирования и управления в АО “НПО автоматики”



o.bulygin@netology.ru



<https://www.facebook.com/obulygin91>



О чём мы поговорим сегодня

1. Почему Python?
2. Интерактивная оболочка Jupyter Notebook
3. Переменные
4. Система контроля версий Git



Python 3

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.





Простой синтаксис

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




Простой синтаксис

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

JAVA

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



Простой синтаксис

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

C

```
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    printf("Hello, world!\n");
    return EXIT_SUCCESS;
}
```



Простой синтаксис

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

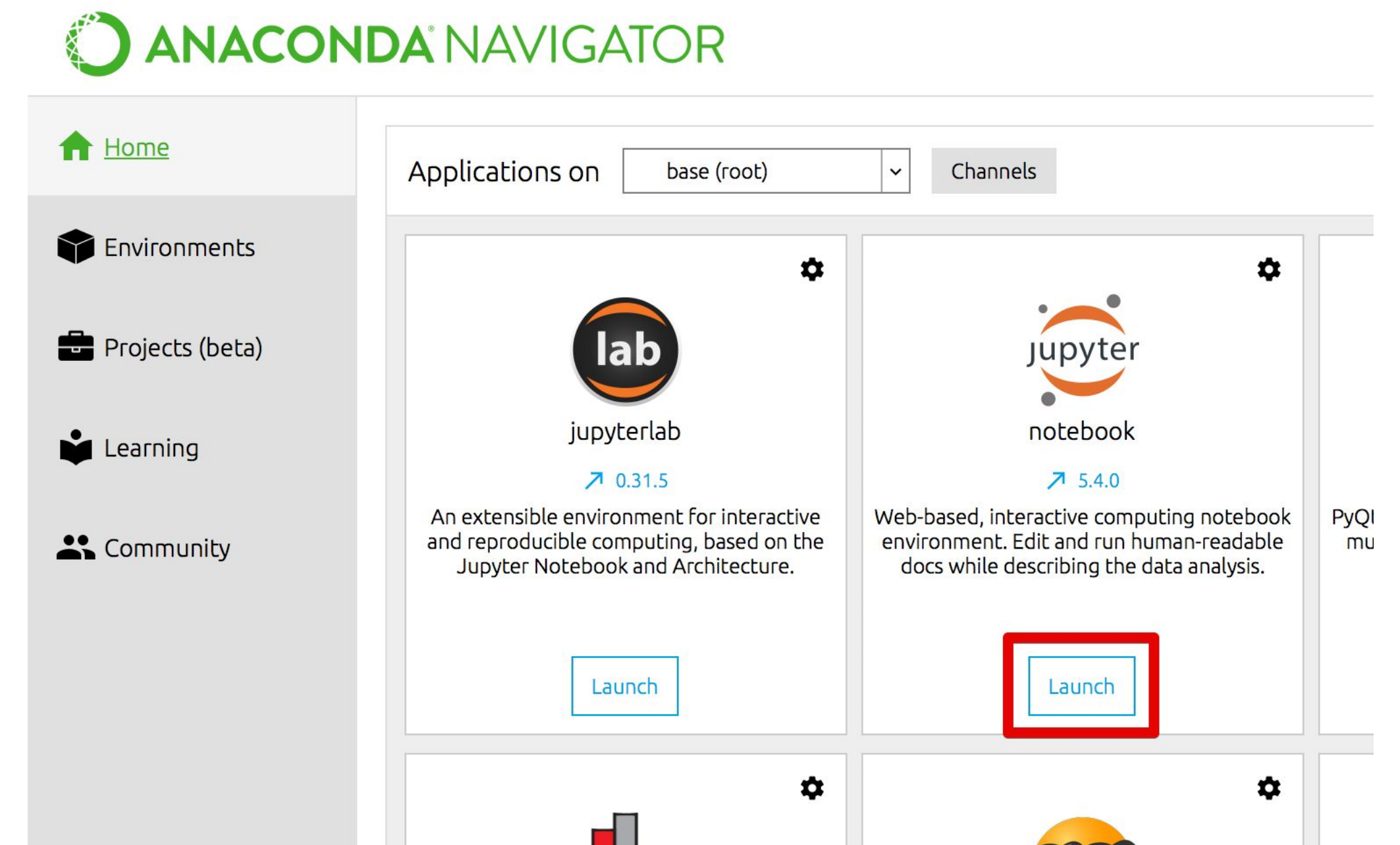
Brainfuck

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




Как запустить Jupyter Notebook

- через Anaconda Navigator;
- в Anaconda prompt;
- в командной строке.





Выполнить команду и перейти на следующую ячейку

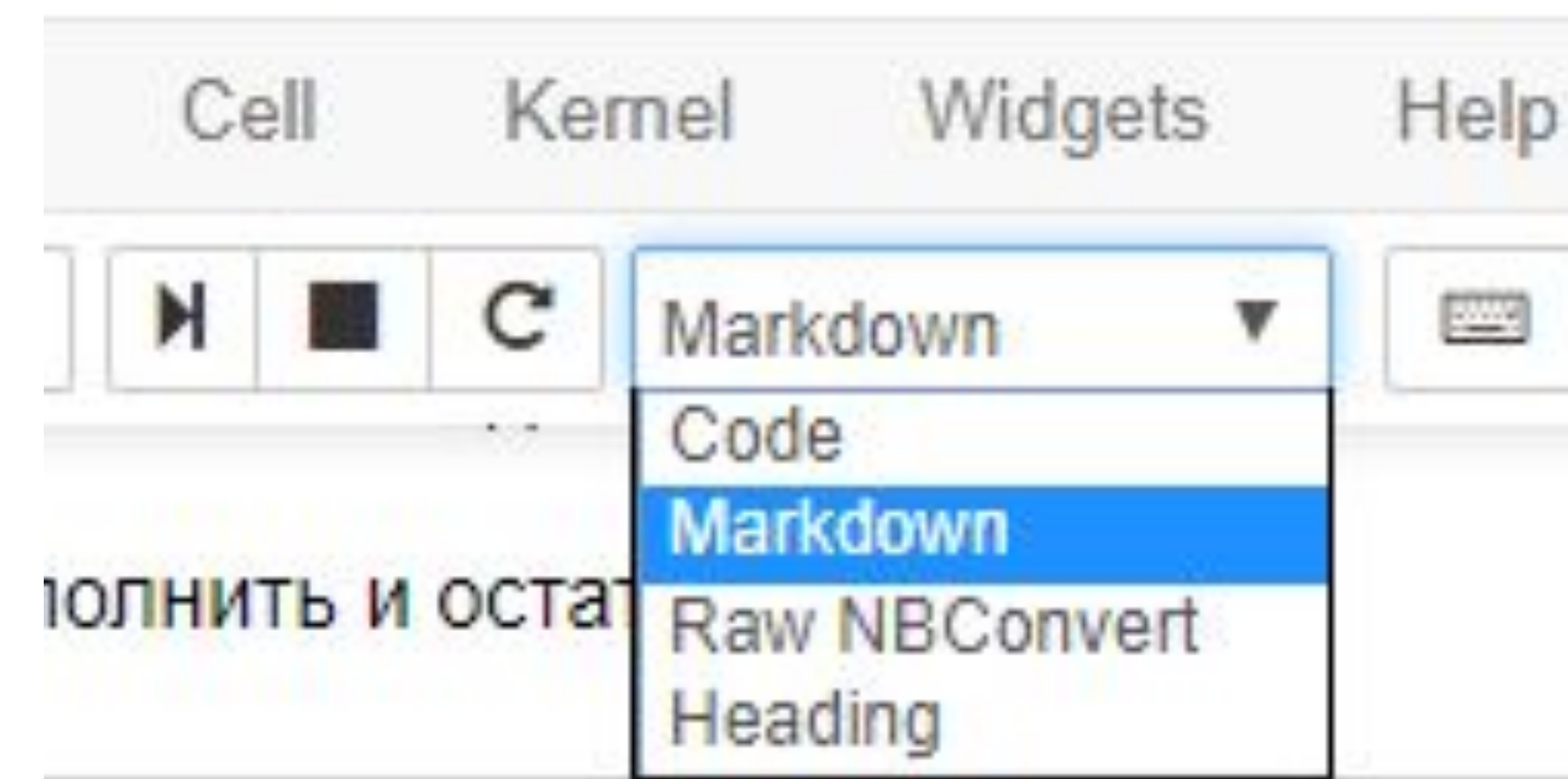
Shift + Enter

Выполнить команду и остаться в текущей ячейке

Ctrl + Enter

Комментарии и формулы

Markdown





Комментарии

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

Написание комментария начинается с символа **#**.



Арифметические операции

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

Приоритет операции аналогичен стандартным математическим правилам, можно использовать скобки.



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

Правила именования:

- имя переменной может состоять только из цифр, букв и знаков подчеркивания.
- имя переменной не может начинаться с цифры.

Рекомендации именования:

- имя переменной должно описывать ее суть;
- лучше использовать snake_case (слова писать с маленькой буквы и разделять нижним подчеркиванием).

Помните: Python – регистрозависимый язык.



Операторы сравнения

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

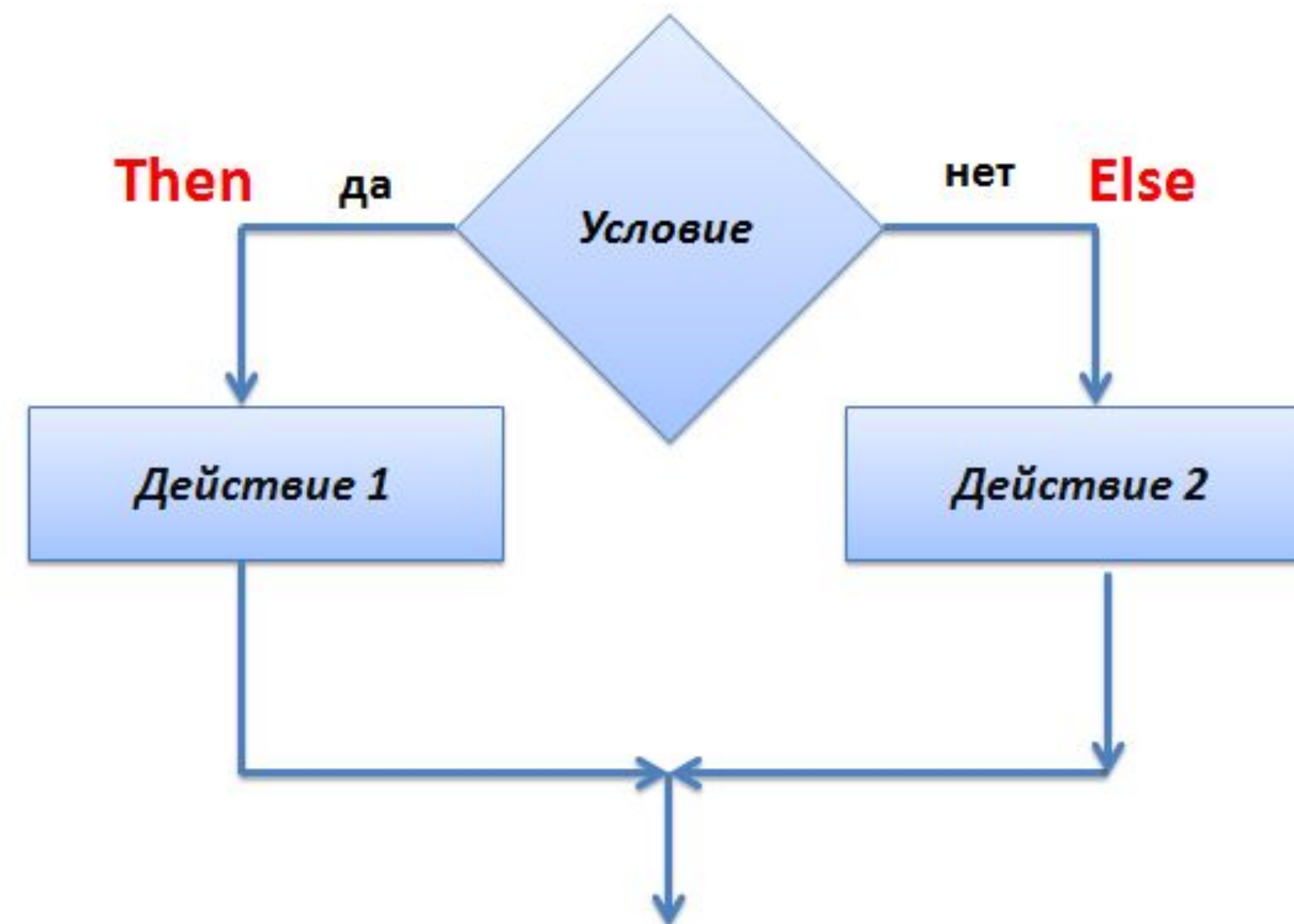
В результате операций сравнения возвращается булево значение (True / False).

Сравнения могут быть записаны в цепочку.



Условия

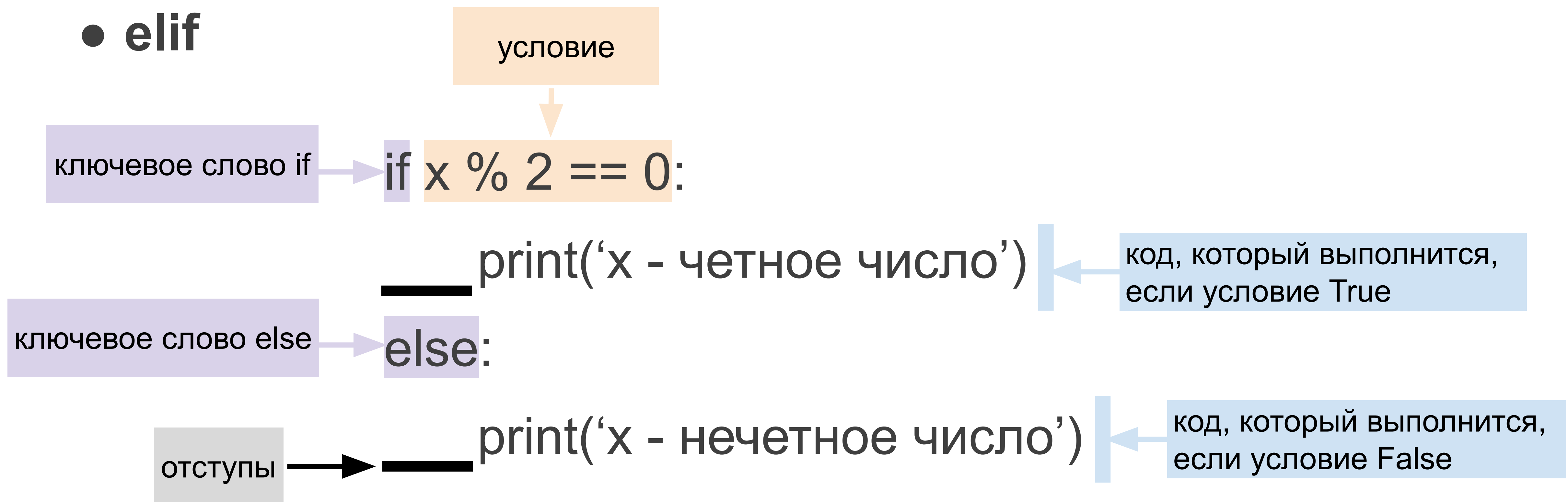
- это способ управлять выполнением программы;
- это способ запрограммировать принятие решений;
- это логическое выражение после которого пишутся команды, которые выполняются, если условие истинно.





Условные конструкции

- **if**
- **else**
- **elif**





Логические операторы

- **and** (логическое И) – возвращает True, только когда оба операнда True
- **or** (логическое ИЛИ) – возвращает True, когда хотя бы один операнд True
- **not** (логическое НЕ) – возвращает булево значение, противоположное операнду



Логические значения

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

```
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

- любое ненулевое число – **это истина**
- любая непустая строка – **это истина**
- 0, "", [], {}, (), False – **это ложь**



НЕТОЛОГИЯ
групп

Спасибо за внимание!

ОЛЕГ БУЛЫГИН



<https://www.facebook.com/obulygin91>



o.bulygin@netology.ru



<https://vk.com/obulygin91>



<https://www.linkedin.com/in/obulygin>