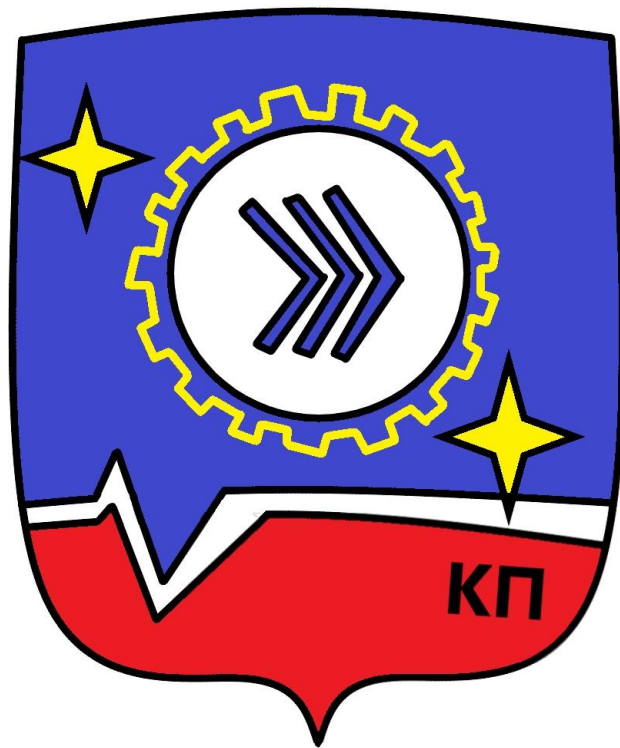


КЛУБ ПРОГРАММИСТОВ Г.КОРОЛЁВА



КУРС “Программирование на PYTHON”



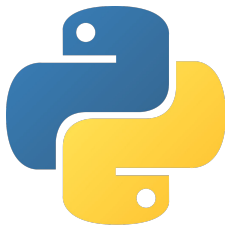


РЕГЛАМЕНТ



**1. Вопросы можно задавать
в любое время**





РЕГЛАМЕНТ



1. Вопросы можно задавать в любое время

**2. Встречаемся 1 раз в неделю
(или реже) по субботам**





РЕГЛАМЕНТ



1. Вопросы можно задавать в любое время
2. Встречаемся 1 раз в неделю (или реже) по субботам

**3. Две лекции по 45 минут
+ дополнительная лекция**





РЕГЛАМЕНТ



1. Вопросы можно задавать в любое время
2. Встречаемся 1 раз в неделю (или реже) по субботам
3. Две лекции по 45 минут + дополнительная лекция

**4. Домашние задания – ДА,
аттестация – ДА,
сертификаты – ДА**



**KEEP
CALM
AND
DO YOUR
HOMEWORK**



РЕГЛАМЕНТ



1. Вопросы можно задавать в любое время
2. Встречаемся 1 раз в неделю (или реже) по субботам
3. Две лекции по 45 минут + дополнительная лекция
4. Домашние задания - ДА, аттестация - ДА, сертификаты - ДА

5. Площадка для общения



https://vk.com/python_korolev



+7 (909) 820-56-18



Я.П.



Java
JavaScript
Visual Basic
Python
C or C++
C#
Objective-C
Perl
Ruby
PHP
R



Java
JavaScript
C#
PHP
Python
C++
Swift
Ruby
Go
TypeScript



JavaScript
HTML
CSS
SQL
Java
Bash/Shell
Python
C#
PHP
C++



ГВИДО

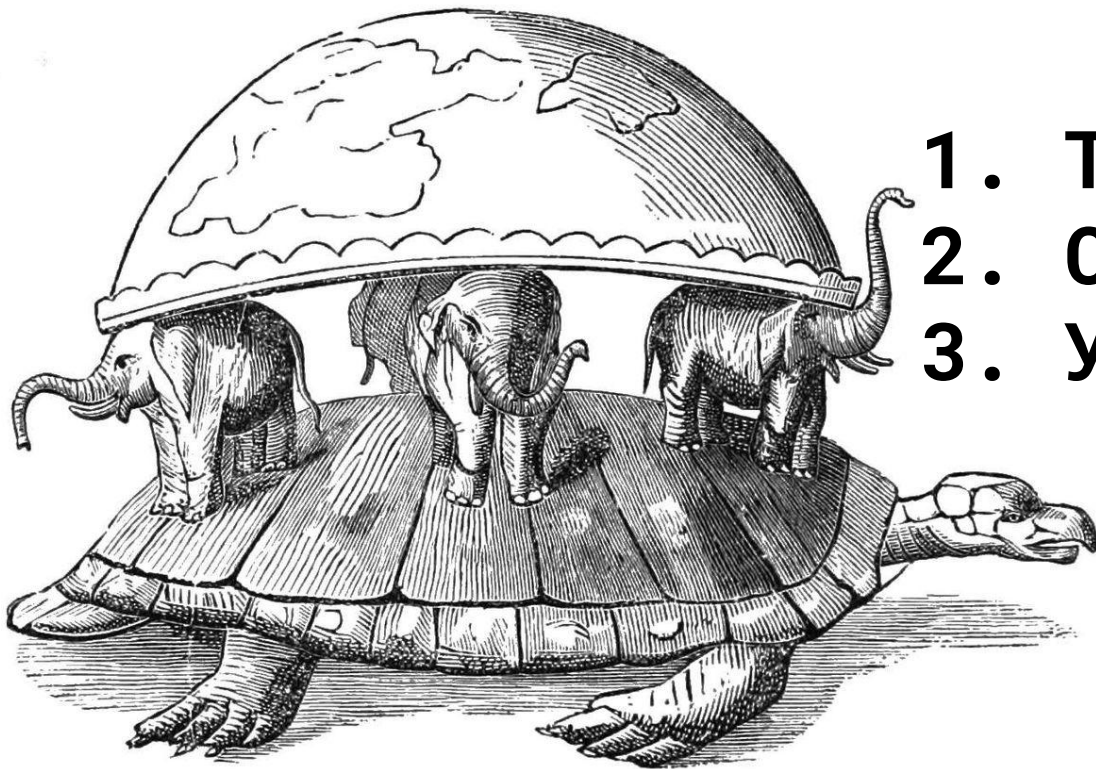


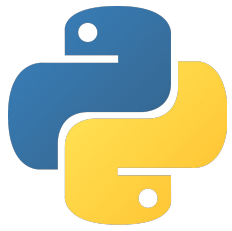


ЧТО ТАКОЕ ЯП



1. Типы данных
2. Структуры
3. Управляющие конструкции





MY NAME IS




python



ДОМ: python.org



[Python](#)[PSF](#)[Docs](#)[PyPI](#)[Jobs](#)[Community](#)


 **python**TM

[GO](#) [Socialize](#)

[About](#)[Downloads](#)[Documentation](#)[Community](#)[Success Stories](#)[News](#)[Events](#)

```
# Python 3: Simple output (with Unicode)
>>> print("Hello, I'm Python!")
Hello, I'm Python!

# Input, assignment
>>> name = input('What is your name?\n')
>>> print('Hi, %s.' % name)
What is your name?
Python
Hi, Python.
```



Quick & Easy to Learn

Experienced programmers in any other language can pick up Python very quickly, and beginners find the clean syntax and indentation structure easy to learn. [Whet your appetite](#) with our Python 3 overview.

[1](#)[2](#)[3](#)[4](#)[5](#)



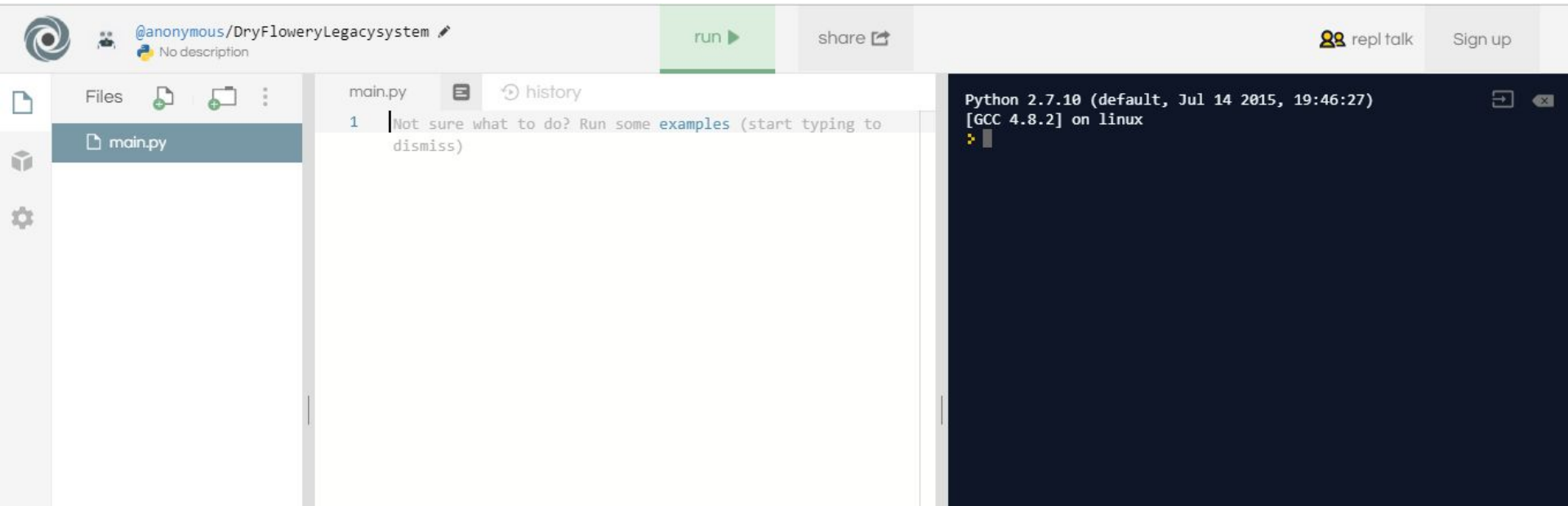
УСТАНОВКА





PYTHON ONLINE

repl.it





PYTHON и command line



```
C:\> Command Prompt - python

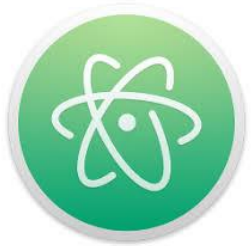
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Eric>python
Python 2.7.12 (v2.7.12:d33e0cf91556, Jun 27 2016, 15:19:22) [MSC v.1500 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> pytest.py
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'pytest' is not defined
>>> _
```




РЕДАКТОРЫ & IDE

Integrated
Development
Environment



ATOM



SUBLIME TEXT 3



PYCHARM



VS CODE



ПАРАДИГМЫ ПРОГРАММИРОВАНИЯ



Императивное программирование

Структурное программирование

Функциональное программирование

Объектно-ориентированное

программирование



ТИПЫ ДАННЫХ



int

ЦЕЛОЕ - INTEGER

1

2

3



ПРОЦЕДУРНОЕ ПРОГРАММИРОВАНИЕ



Выполнение программы сводится к **последовательному выполнению** операторов с целью преобразования **исходного состояния** памяти, то есть значений исходных данных, в заключительное, то есть **в результаты**.

Таким образом, с точки зрения программиста имеются программа и память, причем первая последовательно обновляет содержимое последней.



ДЖОН ФОН НЕЙМАН



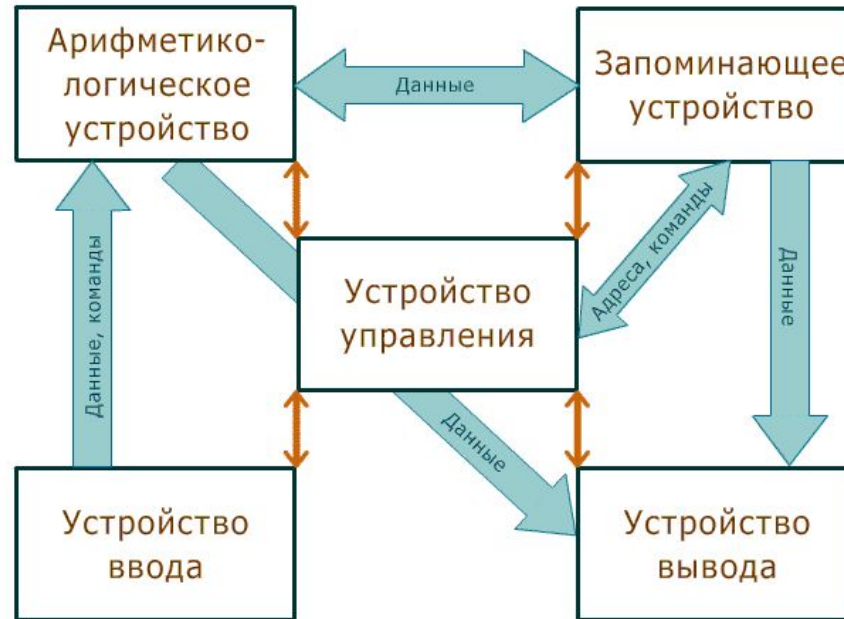


АРХИТЕКТУРА ФОН НЕЙМАНА



<http://inf1.info>

Схема вычислительной машины фон Неймана





IDLE

Integrated Development and Learning Environment
интегрированная среда разработки и обучения



```
Python 3.6.4 Shell
File Edit Shell Debug Options Window Help
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> import this
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
>>> |
```



.py



hello.py

```
print('Hello World!')
```

```
c:\python.exe hello.py
```

```
print('Hello', 'World!')
```



ФУНКЦИЯ



`print()`
`len()`



ТИПЫ ДАННЫХ



str

СТРОКА - STRING

"Hello"

'Hello'

'''hello'''

"""hello"""

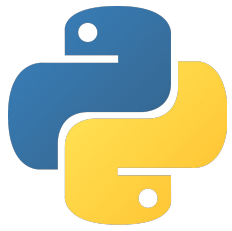
r"raw_string"



ТИПЫ ДАННЫХ: строка



“Королёв” [2] >>> ...



ТИПЫ ДАННЫХ: строка



`"Королёв"[2] >>> 'р'`



ТИПЫ ДАННЫХ неизменяемость



int
str

a = 5

b = 'MAN'

b[1]



ТИПЫ ДАННЫХ неизменяемость



int

str

a = 5

b = 'MAN'

b[1] = 'E' ???



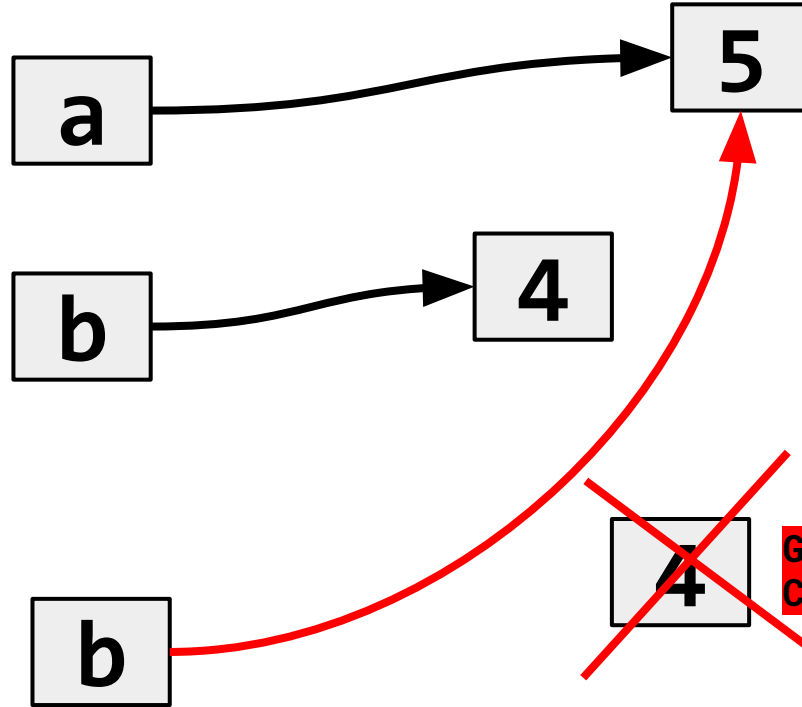
ТИПЫ ДАННЫХ

присваивание значений

a = 5

b = 4

a = b



id()



ТИПЫ ДАННЫХ ТИПИЗАЦИЯ

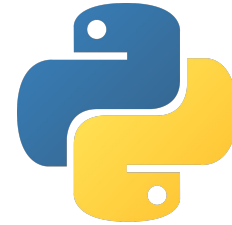


```
a=5  
a='hello'
```

СТАТИЧЕСКАЯ

ДИНАМИЧЕСКАЯ

type()



**приведение
ТИПОВ**



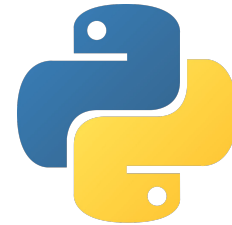
ТИПЫ ДАННЫХ ТИПИЗАЦИЯ



СЛАБАЯ

'слово' + 6

**СИЛЬНАЯ
(строгая)**



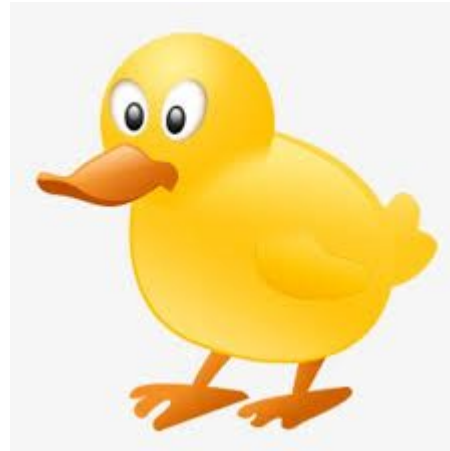


ТИПЫ ДАННЫХ ТИПИЗАЦИЯ



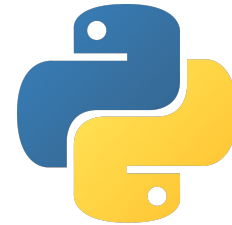
ЯВНАЯ

```
int a = ?
```



НЕЯВНАЯ

```
a = ?
```





СТРУКТУРЫ СПИСОК LIST



```
x = [1, 2, "a", a, [1, 3]]  
y = list(1, 2, "g")  
len(x)  
x[3]  
x[1] = 4
```



СТРУКТУРЫ КОРТЕЖ TUPLE

```
z = 1,  
x = (1, 2, "a", a, [1, 3])  
y = tuple(1, 2, "g")  
len(x)  
x[3]  
x[1] = 4
```



СТРУКТУРЫ LIST МЕТОДЫ

```
a = [1, 2, 3]  
a.append(4) # [1, 2, 3, 4]  
list.append(a, 4)  
a.insert(1, 'yeah')  
a.remove(2)
```



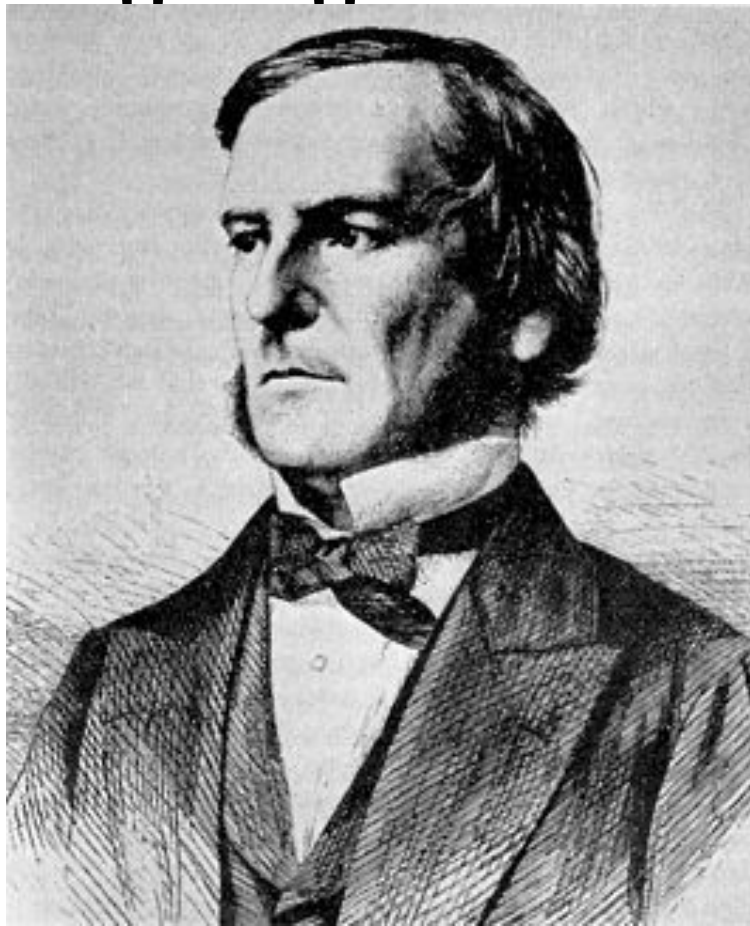
ЛОГИЧЕСКИЕ ОПЕРАЦИИ

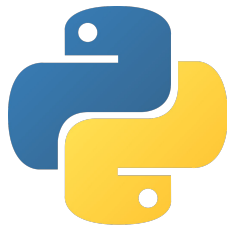


True - ПРАВДА - 1
False - ЛОЖЬ - 0



ДЖОРДЖ БУЛЬ





ЛОГИЧЕСКИЕ ОПЕРАЦИИ

a is b - идентичность
(сравнение адресов в памяти)

is / is not
None

a == b
(сравнение значений)



ЛОГИЧЕСКИЕ ОПЕРАЦИИ



a = 4

b = 10

a == b # False

a > b # False

a >= b # False

a < b # True

a <= b # True

a != b # True

a <= 5 <= b



ЛОГИЧЕСКИЕ ОПЕРАЦИИ

принадлежность группе

`in` / `not in`



```
a = [1, 6, 7, 'j', 'alpha']  
7 in a # True
```

```
b = 'Я хочу спать!'  
'ь' not in b # False
```



ЛОГИЧЕСКИЕ ОПЕРАЦИИ

операторы

and, or, not



`1 < 2 and 2 < 4`
`3 > 4 or 2 < 5`
`not 3 < 2`



УПРАВЛЕНИЕ ПОТОКОМ ВЕТВЛЕНИЕ



if





УПРАВЛЕНИЕ ПОТОКОМ ВЕТВЛЕНИЕ



if условие: # True / False
выражение



отступ

```
if (условие) {  
    выражение;  
}
```



УПРАВЛЕНИЕ ПОТОКОМ ВЕТВЛЕНИЕ



```
if условие:  
    выражение  
else:  
    выражение
```



УПРАВЛЕНИЕ ПОТОКОМ ВЕТВЛЕНИЕ



```
if условие:  
    выражение  
elif условие:  
    выражение  
else:  
    выражение # pass
```



УПРАВЛЕНИЕ ПОТОКОМ ВЕТВЛЕНИЕ



```
if условие:  
    выражение  
elif условие:  
    выражение  
elif условие:  
    выражение  
else:  
    выражение
```



УПРАВЛЕНИЕ ПОТОКОМ ЦИКЛ WHILE



while условие: # True / False
выражение

while True:

break
continue



УПРАВЛЕНИЕ ПОТОКОМ ЦИКЛ FOR .. IN



**for переменная in итератор:
выражение**

**break
continue**



УПРАВЛЕНИЕ ПОТОКОМ ОБРАБОТКА ИСКЛЮЧЕНИЙ



try:

выражение

except:

выражение

except TypeError:

except ValueError as err:



УПРАВЛЕНИЕ ПОТОКОМ АРИФМЕТИКА



5 + 6

11

a = 1

3 - 7

-4

a += 2 # 3

4 * 8

32

a = a + 2

3 / 2

1.5

5 // 2

2

5 % 2

1

3 ** 2

9



“АРИФМЕТИКА”

“Hello” + “World” = “HelloWorld”

“Rx” * 3 = “RxRxRx”

[1, 2, 3] + [2] = [1, 2, 3, 2]

[1, 2] * 2 = [1, 2, 1, 2]



ВВОД ДАННЫХ

```
x = input( 'Введите число: ' )  
int(x)
```



ФУНКЦИИ



```
def имя(аргументы):  
    операции  
    return результат
```



import



```
import random  
print(random.randint(1, 6))  
print(random.choice("любит", "не любит"))
```

```
from random import randint  
print(randint(0, 41))
```



`range()`



`range(10)`

`range(1, 10)`

`range(1, 10, 3)`

`range(100, 10, -2)`