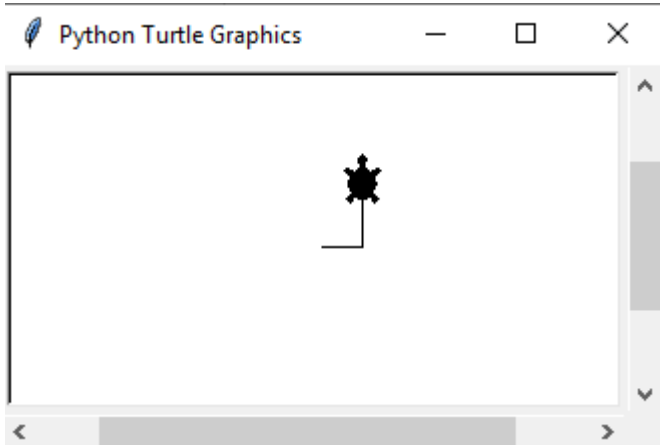
Простейшая программа на Python

```
print('Hello World!')
age = input('Введите возраст: ')
print('Вам ' + age + ' лет’)
input()
```

«Черепашья графика» на Python

```
>>> import turtle
>>> turtle.forward(20)
>>> turtle.shape("turtle")
>>> turtle.left(90)
>>> turtle.forward(30)
```



https://opentechschool.github.io/python-beginners/ru/simple_drawing.html

Простые типы данных в Python

В Python два простых типа данных:

1. Строки (string).
2. Числовой. Подтипы:
    - 2.1. Целое (integer).
    - 2.2. Вещественное (float).
    - 2.3. Комплексное (complex number).
    - 2.4. Логический (булев) тип (boolean).

Числовые типы в Python

Операции с целыми числами:

```
>>> x = 5 + 2 – 3 * 2
>>> x
1
>>> 5 / 2
2.5
>>> 5 // 2
2
>>> 5 % 2
1
>>> 2 ** 8
256
>>> 1000000001 ** 3
1000000003000000003000000001
```

Операции с вещественными числами:

```
>>> x = 4.3 ** 2.4
>>> x
33.13784737771648
>>> 3.5e30 * 2.77e45
9.695e+75
>>> 1000000001.0 ** 3
1.000000003e+27
```

Числовые типы в Python

Операции с комплексными числами:

```
>>> (3+2j) ** (2+3j)
(0.6817665190890336 2.120745776 6159625j)
>>> x = (3+2j) * (4+9j)
>>> x
(6+35j)
>>> x.real
6.0
>>> x.imag
35.0
```

Математические функции:

```
>>> round(3.49)
3
>>> import math
>>> math.ceil(3.49)
3
```

Логический тип:

```
>>> x = False
>>> x
False
>>> not x
True
>>> y = True * 2
>>> y
2
```

Строки в Python

- В Python строки могут задаваться в одинарных (‘), двойных (“), тройных (‘’’) кавычках и включать знаки перехода на новую строку (\n) и табуляции(\t).
- В три двойные кавычки (“””) можно заключать многострочные строки с любыми знаками препинания между ними. В строки можно включать различные переменные, задавать их формат представления.

```
>>> e = 2.718
>>> x = [1, "two", 3, 4.0, ["a", "b"], (5, 6)]
>>> print("The constant e is:", e, "and the list x is:", x)
The constant e is: 2.718 and the list x is: [1, 'two', 3, 4.0,
['a', 'b'], (5, 6)]
>>> print("the value of %s is: %.2f" % ("e", e))
the value of e is: 2.72
```

Преобразование типов в Python

Операторы преобразования типов:

int(x) – к целому

float(x) – к вещественному

complex(x) – к комплексному

bool(x) – к логическому

str(x) – к строке

```
>>> print('race ' + 828)
Traceback (most recent call last):
  File "<pyshell#8>", line 1, in <module>
    print('race ' + 828)
TypeError: must be str, not int
>>> print('race ' + str(828))
race 828
```

```
>>> print('Введите год рождения')
Введите год рождения
>>> year = input()
1985
>>> age = 2019 - year
Traceback (most recent call last):
  File "<pyshell#14>", line 1, in <module>
    age = 2019 - year
TypeError: unsupported operand type(s) for -: 'int' and 'str'
>>> age = 2019 - int(year)
>>> print(age)
34
```

Пример ветвления в Python

```
x = 5
if x < 5:
        y = 1
        z = 5
elif x > 5:
        y = 1
        z = 11
else:
        y = 0
        z = 10
print(x, y, z)
```

Блоки кода задаются с помощью отступов!

Операторы сравнения в Python:

== равно != не равно

больше < меньше

<= меньше или равно >= больше или равно

Логические операции в Python:

and логическое И

or логическое ИЛИ

not логическое НЕ

Цикл while в Python

```
u, v, x, y = 0, 0, 100, 30

while x > y:
        u = u + y
        x = x - y

        if x < y + 2:
                v = v + x
                x = 0

        else:
                v = v + y + 2
                x = x – y - 2
print(u, v)
```

Цикл for в Python

Используется в форме **for in *диапазон:***

```
item_list = [3, "string1", 23, 14.0, "string2", 49, 64,
70]
for x in item_list:
    if not isinstance(x, int):
        continue
    if not x % 7:
        print("found an integer divisible \
by seven: %d" % x)
        break
```

Функция range в Python

Функция range имеет следующие формы:

- **range(stop)**: возвращает все целые числа от 0 до stop
- **range(start, stop)**: возвращает все целые числа в промежутке от start (включая) до stop (не включая).
- **range(start, stop, step)**: возвращает целые числа в промежутке от start (включая) до stop (не включая), которые увеличиваются на значение step