Python 3

Введение

Установка и среды разработки

Установщики https://www.python.org/downloads/
Среды разработки http://wiki.python.org/moin/PythonEditors

Windows
просто использовать msi пакет

Linux

```
sudo apt-get install python3
sudo yum install python3
```

Mac
либо установщик .dmg
mac ports
brew

Структура файла\ов

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

# наша крутая программа из 1 строки
print("Hello Python!")  # та самая строка
```

Комментарии

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

# наша крутая программа из 1 строки
print("Hello Python!")  # та самая строка
```

Блоки кода и двумерный синтаксис

```
if a > 5:
    print("We are inside the block")
    print(a)
print("We are outside the block")
```

Следование

```
a = 1
b = 2
print(a + b)
print("hello")
```

Переменные

```
a = 5
b = "Hello"
c = [1, 2, 3]
d = {
   "key1": "string_value",
   "key2": 1,
   "key3": [1, 2, 3]
}
```

Ветвление

```
if a > b:
   c = a
   print(c)
else:
   c = b
   print(c)
```

Оператор выбора

```
if a > b:
   c = a
   print(c)
elif c > b:
   b = a
   print(b)
else:
   c = b
   print(c)
```

```
if a > b:
   c = a
   print(c)
else:
   if c > b:
       b = a
       print(b)
   else:
       c = b
       print(c)
```

Операторы сравнения

Operator	Name	Example
==	Equal	x == y
!=	Not equal	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

Логические связки (операторы)

Operator	Description	Example
and	Returns True if both statements are true	x < 5 and x < 10
or	Returns True if one of the statements is true	x < 5 or x < 4
not	Reverse the result, returns False if the result is true	not(x < 5 and x < 10)

Цикл while

```
i = 10
while i != 0:
   print(i)
   i = i - 1
```

Цикл for

```
for i in range(0, 10):
   for j in range(0, 10):
       print(i, " ", j)
```

Оператор прерывания цикла break

```
i = 10
while i != 0:
   i = i - 1
   if i > 5:
       break
   print(i)
```

```
9
8
7
6
5
```

Оператор перехода на следующую итерацию цикла continue

```
i = 10
while i != 0:
   i = i - 1
   if i % 2 == 1:
       continue
   print(i)
```

```
9
7
5
3
1
```

Функции

```
def quadratic(a, b, c):
  x1 = -b / (2 * a)
  x2 = math.sqrt(b ** 2 - 4 * a * c) / (2 * a)
  return (x1 + x2), (x1 - x2)


print(quadratic(2, 5, 2))
```

```
./pycharm_project/main.py"
(-0.5, -2.0)

Process finished with exit code 0
```

Исключения

```
try:
   a = 5 / 0
except ZeroDivisionError:
   print("Error: division by zero")
else:
   print("Division result=", a)
finally:
   print("We will get here anyway")

print("Continue code execution")
```

```
Error: division by zero
We will get here anyway
Continue code execution
```

Итераторы

```
class                                   ReverseStringIterator:
                        def          __init__(self,          s):
                                        self.__s      =      s

                                def             __iter__(self):
                                        self.__i      =      0
                                             return       self

                                def             __next__(self):
                    if    self.__i    >    len(self.__s)    -    1:
                                            raise    StopIteration
                                                                 else:
                                    a    =    self.__s[-self.__i    -    1]
                                       self.__i    =    self.__i    +    1
                                                          return     a

r_string             =             ReverseStringIterator("Hello")
for                symbol                in                r_string:
   print(symbol)
```

```
o
l
l
e
H

Process finished with exit code 0
```

Генераторы

```
def f_gen():
    for i in range(1, 5):
        yield i

for n in f_gen():
    print(n)
```

```
1
2
3
4

Process finished with exit code 0
```

Ввод с клавиатуры

```
a = input()
print(a)
```

```
1
1
```

Файлы чтение

```
f_in = open("in.txt", 'r')
for line in f_in:
   print(line)
f_in.close()
```

```
1

2

3

4
```

```
with open("in.txt", 'r') as f_in:
   for line in f_in:
       print(line)
```

```
1

2

3

4
```

Файлы запись

```
f_in = open("in.txt", 'r')
f_out = open("out.txt", 'w')
for line in f_in:
   f_out.write(line)
f_in.close()
f_out.close()
```

Файлы режим открытия

Режим	Обозначение
‘r’	открытие на чтение (является значением по умолчанию).
‘w’	открытие на запись, содержимое файла удаляется, если файла не существует, создается новый.
‘x’	открытие на запись, если файла не существует, иначе исключение.
‘a’	открытие на дозапись, информация добавляется в конец файла.
‘b’	открытие в двоичном режиме.
‘t’	открытие в текстовом режиме (является значением по умолчанию).
‘+’	открытие на чтение и запись

Еще есть классы и модули, о них дальше