

# Лекция 3

Язык программирования Python.

Хайрулин Сергей Сергеевич

email: [s.khairulin@g.nsu.ru](mailto:s.khairulin@g.nsu.ru), [s.khayrulin@gmail.com](mailto:s.khayrulin@gmail.com)

Ссылка на [материалы](#)

# План

- Лекции/практические занятия
  - Тест
- Дифференцированный зачет в конце семестра
  - Защита задания

# Литература

## **Начальный уровень**

- Mark Pilgrim. Dive into Python - <http://www.diveintopython.net/>
- Марк Лутц. Изучаем Python, 4-е издание // Символ-Плюс 2011.
- ...

## **Стандарт/Документация**

- PEP-8 - <https://www.python.org/dev/peps/pep-0008/>
- <https://www.python.org/>
- <https://github.com/python/cpython>

## **Экспертный уровень**

- Лучано Рамальо: Python. К вершинам мастерства
- Mitchell L. Model. Bioinformatics Programming Using Python // O'Reilly 2010.

# Версии Python

- Python 2 вышел 2010 году последняя версия 2.7.16 - исправлялись только баги(ошибки) с января 2020 года поддержка прекращена.
- Python 3 появился в 2008, является актуальной версией языка. Текущая стабильная версия 3.8.5 -> в предрелиз 3.9, в разработке 3.10
  - Python 3 не гарантирует совместимости кода с Python 2

# План занятия

- None
- Функции
- Определение
- Аргументы функции
  - Обязательные
  - Необязательные
  - Передача аргументов
    - по значению
    - по ссылке
- Области видимости переменных
- Рекурсия
- Практика

# None

[None](#)

```
In [1]: a = None  
  
In [2]: a is None  
Out[2]: True  
  
In [3]:
```

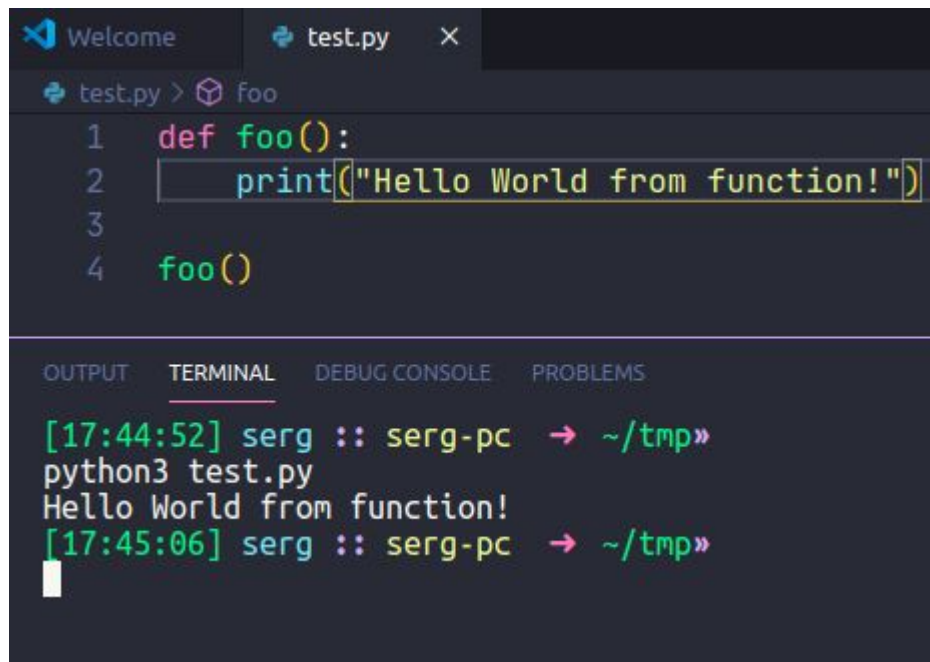
# Функции

**Функция в программировании, или подпрограмма** — фрагмент программного кода, к которому можно обратиться из другого места программы.



# Определение

```
def function_name(arg1, arg2,...,argN):  
    do_some_work  
    return result
```



The screenshot shows a code editor with a dark theme. At the top, there are two tabs: 'Welcome' and 'test.py'. The 'test.py' tab is active, showing a Python function definition. The function is named 'foo' and takes no arguments. It contains a single line of code: 'print("Hello World from function!")'. Below the code editor, there is a panel with four tabs: 'OUTPUT', 'TERMINAL', 'DEBUG CONSOLE', and 'PROBLEMS'. The 'TERMINAL' tab is active, showing the output of running the code. The output consists of three lines: a prompt '[17:44:52] serg :: serg-pc' followed by a right arrow and '~ /tmp', the command 'python3 test.py', and the output 'Hello World from function!'. Below this, there is another prompt '[17:45:06] serg :: serg-pc' followed by a right arrow and '~ /tmp', and a cursor is visible on the line below.

```
Welcome test.py X  
test.py > foo  
1 def foo():  
2     print("Hello World from function!")  
3  
4     foo()  
  
OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS  
[17:44:52] serg :: serg-pc → ~ /tmp»  
python3 test.py  
Hello World from function!  
[17:45:06] serg :: serg-pc → ~ /tmp»  
|
```

# Определение

остановка выполнение  
функции  
ключевое слово **return**

```
1 def triple_sumator(a, b, c):  
2     return a + b + c  
3  
4 print(triple_sumator(1, 2, 3))
```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS

```
[18:24:49] serg :: serg-pc → ~/tmp»  
python3 test.py
```

```
6
```

```
[18:24:50] serg :: serg-pc → ~/tmp»
```

```
□
```

# Аргументы функции

**Аргументы функции** - входные данные над которыми оперирует функция

# Аргументы функции(Обязательные)

```
6 def require_args(arg1, arg2):  
7     print(arg1, arg2)  
8  
9     require_args(1, 2)  
10    require_args(1)  
11
```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS

[18:31:47] serg :: serg-pc → ~/tmp»

python3 test.py

1 2

Traceback (most recent call last):

File "test.py", line 10, in <module>

require\_args(1)

TypeError: require\_args() missing 1 required positional argument: 'arg2'

# Аргументы функции(Необязательные)

```
5
6 def not_require_args(arg1, arg2=2):
7     print(arg1, arg2)
8
9 not_require_args(1, 3)
10 not_require_args(1)
11
```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS

```
[18:32:55] serg :: serg-pc → ~/tmp»
python3 test.py
1 3
1 2
```

# Аргументы функции(Передача аргументов)

Class	Description	Immutable?
<b>bool</b>	Boolean value	✓
<b>int</b>	integer (arbitrary magnitude)	✓
<b>float</b>	floating-point number	✓
<b>list</b>	mutable sequence of objects	
<b>tuple</b>	immutable sequence of objects	✓
<b>str</b>	character string	✓
<b>set</b>	unordered set of distinct objects	
<b>frozenset</b>	immutable form of set class	✓
<b>dict</b>	associative mapping (aka dictionary)	

# Области видимости переменных

```
1 x = 10
2 def bar():
3     print(x)
4 bar()
5
6 x = 10
7 def foobar():
8     global x
9     print(x)
10    x += 1
11 foobar()
12 print(x)
```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS

```
[18:55:07] serg :: serg-pc → ~/tmp»
python3 test.py
10
10
11
```

# Рекурсия

```
11
12 def fib(n):
13     if n == 1 or n == 0:
14         return n
15
16     return fib(n - 1) + fib(n - 2)
17
18 N = int(input("Input num: "))
19 print(f"{N} fib number {fib(N)}")
```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS

```
[18:52:20] serg :: serg-pc → ~/tmp»
python3 test.py
Input num: 10
10 fib number 55
```



# Практическая Часть

## 1. Функции