

Язык программирования Python



Никита Андреевич Шварц



@nikitashvarts



shvarts.nikita.an@gmail.com

Telegram channel



GitHub Repository



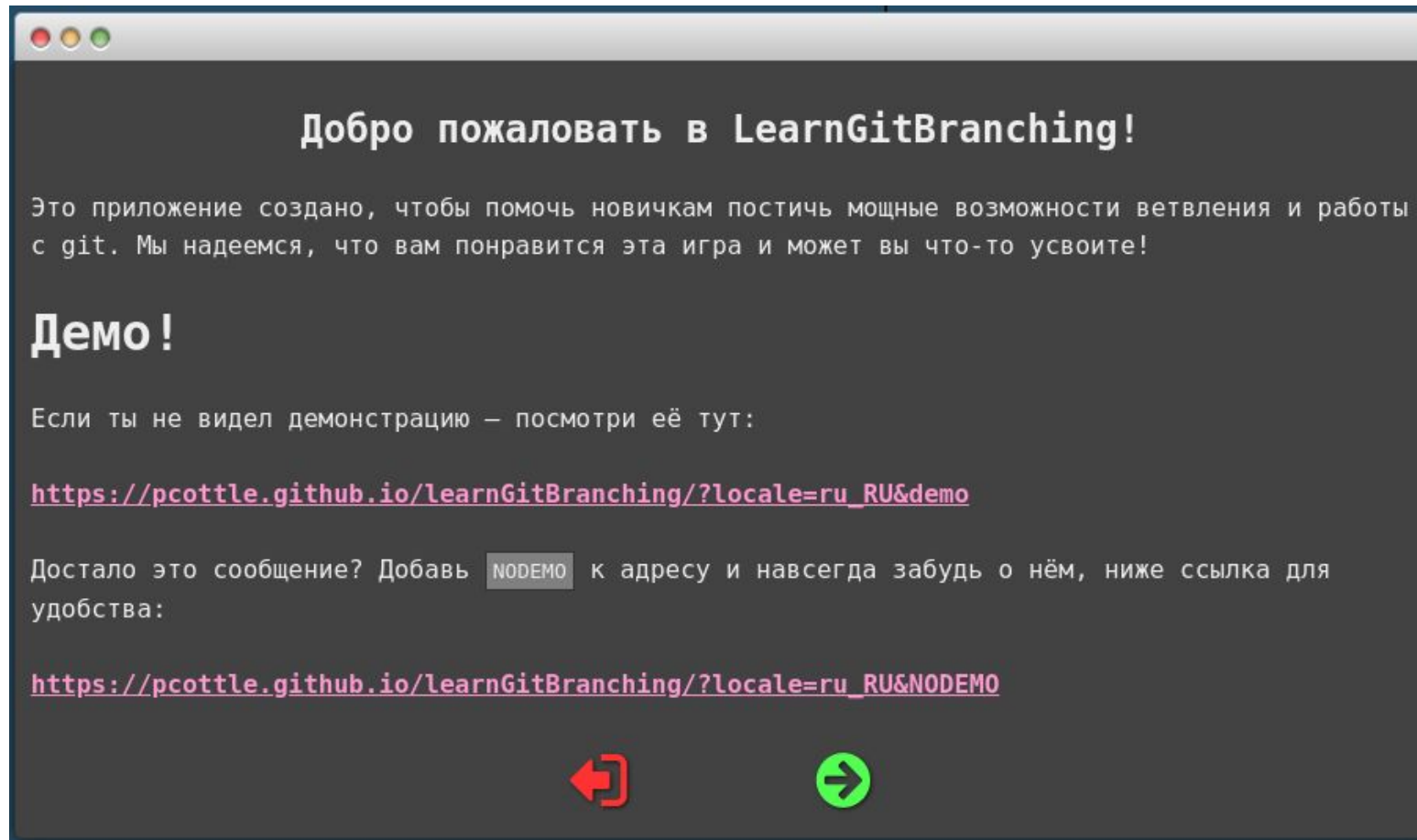
<https://github.com/nikitashvarts/python-basic-course>

Схема сдачи домашних заданий

- Создайте приватный репозиторий с названием `{surname}-{name}-python-basic-course`;
- Добавьте `nikitashvarts` в collaborators своего проекта;
- Склонируйте основной репозиторий себе локально на компьютер в свой проект;
- Создайте ветку с названием `{surname}-hw-*`, где * - номер домашнего задания;
- В этой ветке выложите решение в папку `hw-*`;
- Создайте merge request из созданной ветки в ветку `master`;
- Оставьте ссылку на ваш MR в [этой табличке](#).

Git Tutorial

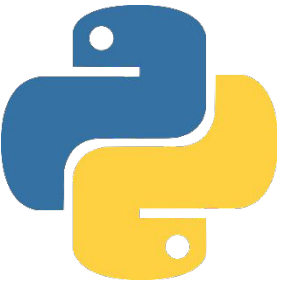
<https://learngitbranching.js.org/>



Пример сдачи ДЗ

- `git checkout -b hw-01`
- `git add homework_01.ipynb`
- `git commit -m 'homework-01'`
- `git push -u origin`

Создавать merge request лучше через интерфейс гитхаба.



Про что этот курс?

Что будет:

- Знакомство с языком программирования
- Изучение базового синтаксиса
- Обзор стандартных библиотек
- Обработка и анализ данных
- Можем потрогать ML

Чего не будет:

- Продвинутое программирование
- Упор на разработку



Section I: Введение



Гарри Поттер обнаружил, что знает
язык, бросил магическую школу
кодить на питоне.



Section I: Введение

Язык программирования Python

- Python - интерпретируемый язык программирования;
- Динамическая типизация;
- Хорошая поддержка модульности;
- Поддержка объектно-ориентированного программирования.

Section I: Введение

Язык программирования Python

- Автоматическая сборка мусора, отсутствие утечек памяти;
- Интеграция с C/C++, если возможностей Python недостаточно;
- Понятный и лаконичный синтаксис, способствующий ясному отображению кода;
- Огромное количество модулей, как входящих в стандартную поставку Python 3, так и сторонних;
- Кроссплатформенность.

Section I: Введение

Компилятор vs. Интерпретатор



Figure.1.1: Интерпретатор анализирует, транслирует и выполняет исходный код программы построчно

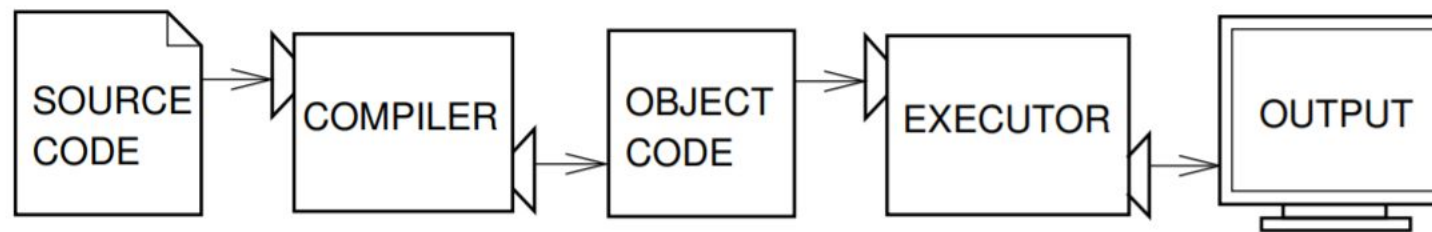


Figure.1.2: Компилятор анализирует и транслирует в машинный код весь текст программы без ее выполнения

Section I: Введение

Ну, здравствуй!

Python Online Interpreter

<https://repl.it/languages/python3>

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

```
print "Hello, World!"
```

```
my_string = "Hello, World!"  
print(my_string)
```

Section I: Введение

Python 3 vs. Python 2

```
print "Sharks are my favorite sea creatures"
```

```
print("Sharks are my favorite sea creatures")
```

```
a = 5 / 2  
print a  
2
```

```
a = 5 / 2  
print(a)  
2.5
```

Section I: Введение

Duck Typing

```
def read_ten(file_like_object):  
    for line_number in range(10):  
        x = file_like_object.readline()  
        print(f"{line_number} = {x.strip()}")
```

```
with open("types.py") as f:  
    read_ten(f)
```

```
class Duck():  
    def readline(self):  
        return "quack"
```

```
my_duck = Duck()  
read_ten(my_duck)
```

```
0 = quack  
1 = quack  
2 = quack  
3 = quack  
4 = quack  
5 = quack  
6 = quack  
7 = quack  
8 = quack  
9 = quack
```

Section I: Введение

Duck Typing

```
def read_ten(file_like_object):  
    for line_number in range(10):  
        x = file_like_object.readline()  
        print(f"{line_number} = {x.strip()}")
```

```
with open("types.py") as f:  
    read_ten(f)
```

```
class Duck():  
    def readline(self):  
        return "quack"
```

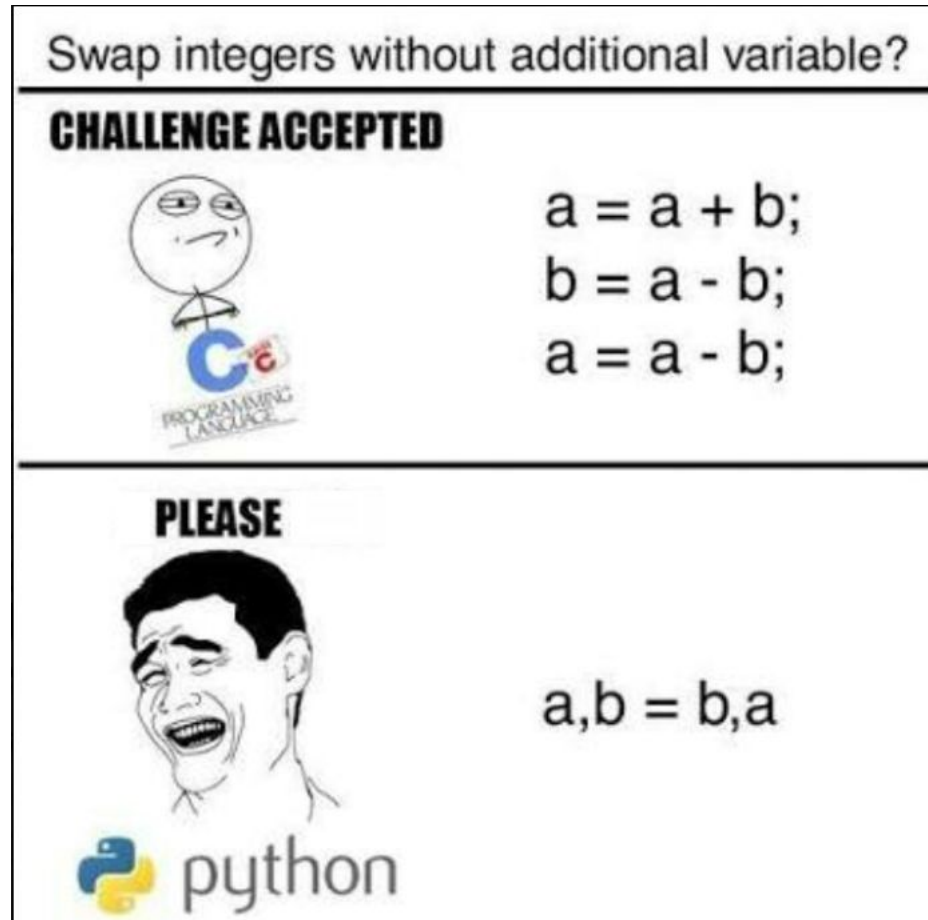
```
my_duck = Duck()  
read_ten(my_duck)
```

```
read_ten(3)
```

```
Traceback (most recent call last):  
  File "<stdin>", line 1, in <module>  
  File "duck_test.py", line 4, in read_ten  
    x = file_like_object.readline()  
AttributeError: 'int' object has no attribute 'readline'
```


Section I: Введение

Duck Typing



Section I: Введение

Запуск программ. Настройка окружения

Python + pip:

- <https://www.python.org/downloads/>
- <https://pypi.org/project/pip/>

Virtualenv vs. Anaconda

- <https://virtualenv.pypa.io/en/latest/>
- <https://www.anaconda.com/>

Google Colab

- <https://colab.research.google.com/>

Section I: Введение

Запуск программ. Настройка окружения

```
n = int(input('Type a number, then its factorial will be printed: '))

if n < 0:
    raise ValueError('You must enter a positive number')

fact = 1
i = 2
while i <= n:
    fact = fact * i
    i += 1

print(fact)
```

Section I: Введение

Запуск программ. Настройка окружения

One-Liner факториал:

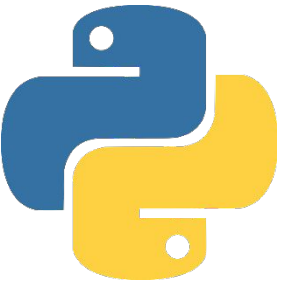
```
reduce(lambda x, y: x * y, range(1, n+1))
```

```
lambda L: [] if L==[] else qsort([x for x in L[1:] if x< L[0]]) +  
L[0:1] + qsort([x for x in L[1:] if x>=L[0]])
```

```
lambda x: x if x<=1 else fib(x-1) + fib(x-2)
```

```
[line.strip() for line in open(filename)]
```

```
phrase.find(phrase[::-1])
```



Section II: Переменные, выражения и операторы

Section II: Переменные, выражения и операторы

Переменные и типы

```
>>> type('Hello, World!')  
<class 'str'>
```

```
>>> type(17)  
<class 'int'>
```

```
>>> type(17.0)  
<class 'float'>
```

```
>>> type([17])  
<class 'list'>
```

```
>>> type({1: 7})  
<class 'dict'>
```

```
>>> type((17))  
<class 'int'>
```

```
>>> type((17,))  
<class 'tuple'>
```

Section II: Переменные, выражения и операторы

Переменные и типы

```
>>> 1000000
```

```
1000000
```

```
>>> 1.000.000
```

```
File "<stdin>", line 1
```

```
    1.000.000
```

```
        ^
```

```
SyntaxError: invalid syntax
```

```
>>> 1,000,000
```

```
(1, 0, 0)
```

```
>>> 1.000000
```

```
1.0
```

```
>>> .1000000
```

```
0.1
```

Section II: Переменные, выражения и операторы

Переменные и типы

```
>>> 2+3j
```

```
(2+3j)
```

```
>>> type(2+3j)
```

```
<class 'complex'>
```

```
>>> z = complex(2, 3)
```

```
>>> z.real
```

```
2.0
```

```
>>> z.imag
```

```
3.0
```

```
>>> a.conjugate()
```

```
(2-3j)
```

```
>>> abs(3+4j)
```

```
5.0
```

```
>>> pow(3+4j, 2)
```

```
(-7+24j)
```


Section II: Переменные, выражения и операторы

Конвертации типов

- `int(x, base)`
- `long(x, base)`
- `float(x)`
- `complex(real, imag)`
- `str(x)`
- `tuple(s)`
- `list(s)`
- `set(s)`
- `dict(d)`
- `frozenset(s)`
- `chr(x)`
- `unichr(x)`
- `ord(x)`
- `hex(x)`
- `oct(x)`

Section II: Переменные, выражения и операторы

Переменные и типы

```
>>> 76trombones = 'big parade'  
SyntaxError: invalid syntax
```

```
>>> more@ = 1000000  
SyntaxError: invalid syntax
```

```
>>> class = 'Advanced Theoretical Zymurgy'  
SyntaxError: invalid syntax
```

Section II: Переменные, выражения и операторы

Ключевые слова

False	def	if	raise
None	del	import	return
True	elif	in	try
and	else	is	while
as	except	lambda	with
assert	finally	nonlocal	yield
break	for	not	
class	from	or	
continue	global	pass	

Section II: Переменные, выражения и операторы

Операторы

`+` – сложение

`-` – вычитание

`*` – умножение

`**` – возведение в степень

`/` – деление

`//` – целочисленное деление

`%` – деление по модулю

`pow(x, y)` – возведение в степень

`divmod(x, y)` – пара $(x//y, x\%y)$

`>` – больше

`>=` – больше или равно

`<` – меньше

`<=` – меньше или равно

`==` – равно

`!=` – не равно

`and` – логическое И

`or` – логическое ИЛИ

`not` – логическое НЕ

Section II: Переменные, выражения и операторы

Строковые операции

```
>>> 1 + 2
```

```
3
```

```
>>> '1 + 2'
```

```
'1 + 2'
```

```
>>> '1' + '2'
```

```
'12'
```

```
>>> '1' - '2'
```

```
>>> '1' * 2
```

```
'11'
```

```
Traceback (most recent call
last):
  File "<stdin>", line 1, in
<module>
TypeError: unsupported operand
type(s) for -: 'str' and 'str'
```

Section II: Переменные, выражения и операторы

Комментарии

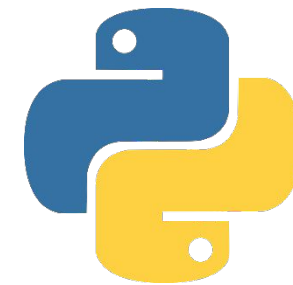
```
# compute the percentage of the hour that has elapsed  
percentage = (minute * 100) / 60
```

```
percentage = (minute * 100) / 60 # percentage of an hour
```

PEP8 – Python Enhancement Proposals

<https://www.python.org/dev/peps/pep-0008/>

<https://pythonworld.ru/osnovy/pep-8-rukovodstvo-po-napisaniyu-koda-na-python.html>



Section III: Функции

Section III: Функции

Математические функции

```
>>> import math
>>> print(math)
<module 'math' (built-in)>
```

```
>>> ratio = signal_power / noise_power
>>> decibels = 10 * math.log10(ratio)
```

```
>>> radians = 0.7
>>> height = math.sin(radians)
```

```
>>> x = math.sin(degrees / 360.0 * 2 * math.pi)
>>> x = math.exp(math.log(x+1))
```

```
>>> minutes = hours * 60 # right
>>> hours * 60 = minutes # wrong!
SyntaxError: can't assign to operator
```

```
>>> degrees = 45
>>> radians = degrees / 360.0 * 2 *
math.pi
>>> math.sin(radians)
0.707106781187
```

```
>>> math.sqrt(2) / 2.0
0.707106781187
```


Section III: Функции

Добавление новых функций

```
def print_lyrics():  
    print("I'm a lumberjack, and I'm okay.")  
    print("I sleep all night and I work all day.")
```

header

body

4

пробела

```
>>> def print_lyrics():  
...     print("I'm a lumberjack, and I'm okay.")  
...     print("I sleep all night and I work all day.")  
... 
```

Section III: Функции

Добавление новых функций

```
>>> print(print_lyrics)
<function print_lyrics at 0xb7e99e9c>
>>> type(print_lyrics)
<type 'function'>
```

```
>>> print_lyrics()
I'm a lumberjack, and I'm okay.
I sleep all night and I work all day.
```

```
def repeat_lyrics():
    print_lyrics()
    print_lyrics()
```

Section III: Функции

Определение и вызов

```
def print_lyrics():  
    print("I'm a lumberjack, and I'm okay.")  
    print("I sleep all night and I work all day.")  
  
def repeat_lyrics():  
    print_lyrics()  
    print_lyrics()  
  
repeat_lyrics()
```

Section III: Функции

Определение и вызов

```
repeat_lyrics()

def print_lyrics():
    print("I'm a lumberjack, and I'm okay.")
    print("I sleep all night and I work all day.")

def repeat_lyrics():
    print_lyrics()
    print_lyrics()
```

```
Traceback (most recent call last):
  File "main.py", line 1, in <module>
    repeat_lyrics()
NameError: name 'repeat_lyrics' is not defined
```

Section III: ФУНКЦИИ

Определение и вызов. Flow of execution

```
def repeat_lyrics():  
    print_lyrics()  
    print_lyrics()  
  
def print_lyrics():  
    print("I'm a lumberjack, and I'm okay.")  
    print("I sleep all night and I work all day.")  
  
repeat_lyrics()
```

```
I'm a lumberjack, and I'm okay.  
I sleep all night and I work all day.  
I'm a lumberjack, and I'm okay.  
I sleep all night and I work all day.
```

Section III: Функции

Параметры и аргументы

```
def print_twice(bruce):  
    print(bruce)  
    print(bruce)
```

```
>>> print_twice('Spam')  
Spam  
Spam  
>>> print_twice(17)  
17  
17  
>>> print_twice(math.pi)  
3.14159265359  
3.14159265359
```

Section III: Функции

Параметры и аргументы

```
>>> print_twice('Spam '*4)
Spam Spam Spam Spam
Spam Spam Spam Spam

>>> print_twice(math.cos(math.pi))
-1.0
-1.0
```

Section III: Функции

Параметры и аргументы. Локальность

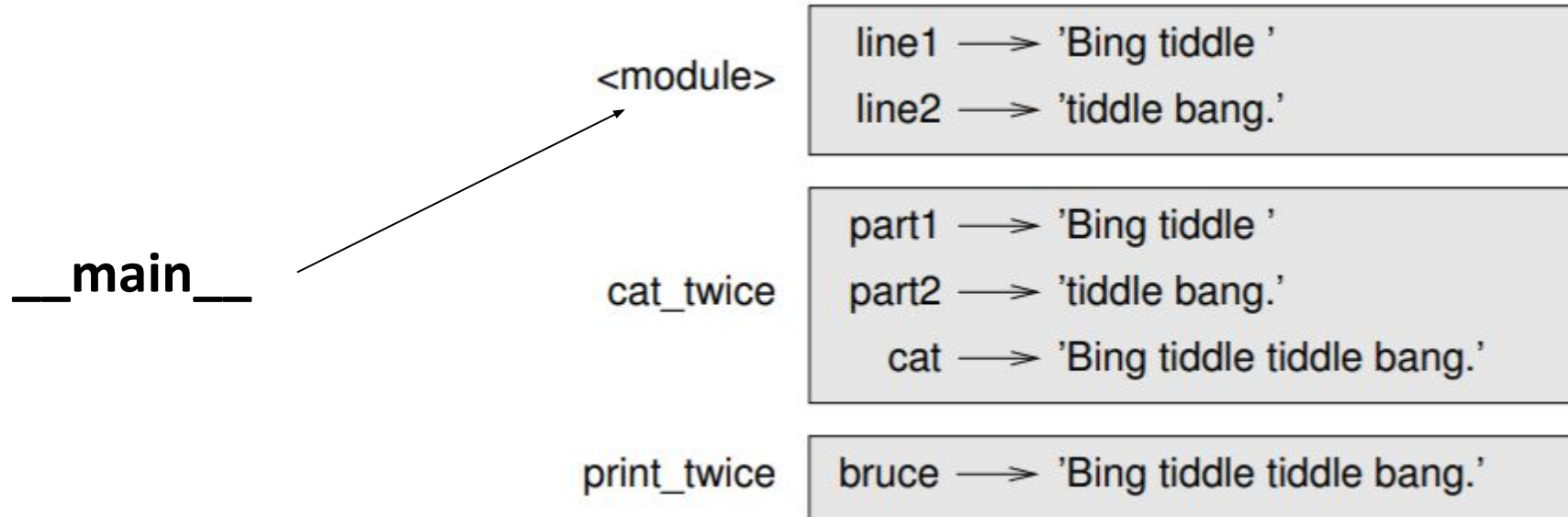
```
def cat_twice(part1, part2):  
    cat = part1 + part2  
    print_twice(cat)
```

```
>>> line1 = 'Bing tiddle '  
>>> line2 = 'tiddle bang.'  
>>> cat_twice(line1, line2)  
Bing tiddle tiddle bang.  
Bing tiddle tiddle bang.
```

```
>>> print cat  
NameError: name 'cat' is not defined
```


Section III: Функции

Stack diagrams



Section III: ФУНКЦИИ

Traceback

```
Traceback (innermost last):  
  File "test.py", line 13, in __main__  
    cat_twice(line1, line2)  
  File "test.py", line 5, in cat_twice  
    print_twice(cat)  
  File "test.py", line 9, in print_twice  
    print cat  
NameError: name 'cat' is not defined
```

Section III: Функции

Fruitful и void функции

```
x = math.cos(radians)
golden = (math.sqrt(5) + 1) / 2
```

```
>>> math.sqrt(5)
2.2360679774997898
```

```
math.sqrt(5)
```

```
>>> result = print_twice('Bing')
Bing
Bing
>>> print(result)
None
```

```
>>> print(type(None))
<type 'NoneType'>
```

Section III: Функции

Импорт с использованием `from`

```
>>> import math
>>> print(math)
<module 'math' (built-in)>
>>> print(math.pi)
3.14159265359
```

```
>>> print(pi)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'pi' is not defined
```

Section III: Функции

Импорт с использованием `from`

```
>>> from math import pi
>>> print(pi)
3.14159265359
```

```
>>> from math import pi as pi_number
>>> print(pi)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'pi' is not defined
>>> print(pi_number)
3.14159265359
```

```
>>> from math import *
>>> cos(pi)
-1.0
```