

Лекция 2

Язык программирования Python.

Хайрулин Сергей Сергеевич

email: s.khairulin@g.nsu.ru, 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.

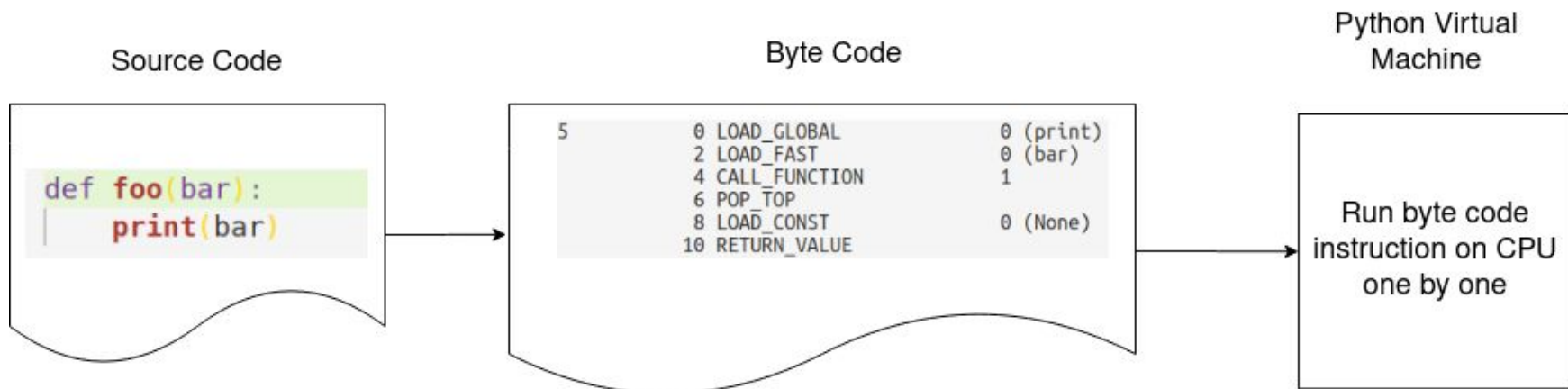
Версии Python

- 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

- Еще немного об интерпретации кода
- Базовые типы данных
 - неизменяемые
 - изменяемые
- Арифметические операции над числами
- Динамическая типизация
- Переменные
- Работа с вводом/выводом
- Практика

Еще немного об интерпретации кода



Базовые типы данных

Числовые типы данных - int, float complex

- **int** (integer) - целое число, например 10, 1, 0,
- **float** - числа с плавающей точкой например 1.2, 3.14,
- **complex** - комплексные числа определяются двумя числами вещественной частью и мнимой:

$$\text{num} = a + i*b \rightarrow \text{где } i = \sqrt{-1}$$

Итераторы

- Специальные объекты позволяющие пробежаться по последовательности

Базовые типы данных

Последовательности

- list -> [1, 3, 4, 5]
- tuple -> (1, 2, 3, 4)
- range -> range(start, end, step) e.g. range(0,10,1)

Текстовые последовательности

- str -> "Hello Python!"

Словари/множества

- set -> set(1,2,3,4)
- frozenset -> frozenset(1,2,3,4)
- dict -> {"John Doe": "+7903222334", "Albert Einstein": "+142345553", ...}

Базовые типы данных

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

Арифметические операции над числами

Operation	Result	Notes	Full documentation
<code>x + y</code>	sum of <code>x</code> and <code>y</code>		
<code>x - y</code>	difference of <code>x</code> and <code>y</code>		
<code>x * y</code>	product of <code>x</code> and <code>y</code>		
<code>x / y</code>	quotient of <code>x</code> and <code>y</code>		
<code>x // y</code>	floored quotient of <code>x</code> and <code>y</code>	(1)	
<code>x % y</code>	remainder of <code>x / y</code>	(2)	
<code>-x</code>	<code>x</code> negated		
<code>+x</code>	<code>x</code> unchanged		
<code>abs(x)</code>	absolute value or magnitude of <code>x</code>		abs()
<code>int(x)</code>	<code>x</code> converted to integer	(3)(6)	int()
<code>float(x)</code>	<code>x</code> converted to floating point	(4)(6)	float()
<code>complex(re, im)</code>	a complex number with real part <code>re</code> , imaginary part <code>im</code> . <code>im</code> defaults to zero.	(6)	complex()
<code>c.conjugate()</code>	conjugate of the complex number <code>c</code>		
<code>divmod(x, y)</code>	the pair <code>(x // y, x % y)</code>	(2)	divmod()
<code>pow(x, y)</code>	<code>x</code> to the power <code>y</code>	(5)	pow()
<code>x ** y</code>	<code>x</code> to the power <code>y</code>	(5)	

Побитовые операции с целыми типами

Operation	Result	Notes
<code>x y</code>	bitwise <i>or</i> of <i>x</i> and <i>y</i>	(4)
<code>x ^ y</code>	bitwise <i>exclusive or</i> of <i>x</i> and <i>y</i>	(4)
<code>x & y</code>	bitwise <i>and</i> of <i>x</i> and <i>y</i>	(4)
<code>x << n</code>	<i>x</i> shifted left by <i>n</i> bits	(1)(2)
<code>x >> n</code>	<i>x</i> shifted right by <i>n</i> bits	(1)(3)
<code>~x</code>	the bits of <i>x</i> inverted	

<https://docs.python.org/3/library/stdtypes.html#bitwise-operations-on-integer-types>

Операции приведения типа

- `int` - приведение к целому числу
- `float` - приведение к числу с плавающей точкой
- `str` - приведение к строке

Операции приведения типа

```
In [10]: x = 3.14
```

```
In [11]: type(x)
```

```
Out[11]: float
```

```
In [12]: y = int(x)
```

```
In [13]: type(y)
```

```
Out[13]: int
```

Операции приведения типа

```
In [14]: z = str(y)
```

```
In [15]: type(y)
```

```
Out[15]: int
```

```
In [16]: z
```

```
Out[16]: '3'
```

```
In [17]: int('123')
```

```
Out[17]: 123
```

```
In [18]: int('one')
```

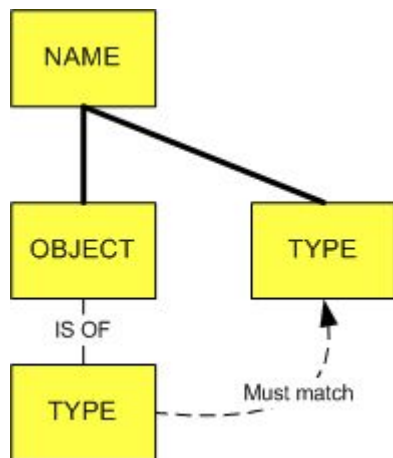
```
-----  
ValueError                                Traceback (most recent call last)  
<ipython-input-18-4f3ea1d042dc> in <module>  
----> 1 int('one')
```

```
ValueError: invalid literal for int() with base 10: 'one'
```

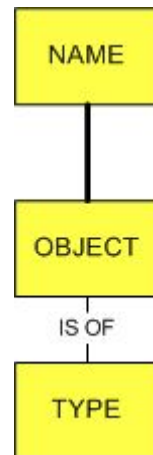
```
In [19]: █
```

Динамическая типизация

Статическая типизация



Динамическая типизация



Динамическая типизация

python

```
x = 10
```

```
x = 'Hello World'
```

```
x = 3.14
```

C++

```
int x = 10;
```

```
string x = "Hello world"
```

```
float x = 3.14
```

Переменные

```
# Создание переменной
```

```
x = 10
```

```
# Переопределение значения переменной,
```

```
# то есть после этой операции переменная
```

```
# будет указывать на другой объект в памяти.
```

```
x = 'Hello World'
```

```
x = 3.14
```

```
# Изменение переменной. После этой операции переменная
```

```
# x указывает на новое значение 4.14
```

```
x = x + 1
```

Переменные

Для сокращения записи, если выражение

подразумевает изменение той же переменной,

то разумно использовать следующие варианты записи

`x += 1` *# тоже самое что и $x = x + 1$*

`x -= 1` *# тоже самое что и $x = x - 1$*

`x *= 1` *# тоже самой что и $x = x * 1$*

`x /= 2` *# тоже самое что и $x = x / 2$*

`x **= 2` *# тоже самое что и $x = x ** 2$*

Присвоение

`x = 1`

`y = 2`

`x = y` *# теперь переменная x равна 2*

`z = x + y` *# теперь переменная z равна 4*

Переменные

Узнать тип переменной можно с помощью функции `type(...)`

```
x = 10
```

```
# результат выполнения этой функции будет
```

```
# тип объекта переданного в аргументе
```

```
# в нашем случае int
```

```
type(x)
```

Узнать месторасположение объекта в памяти можно с помощью `id(...)`

```
# результат зависит от платформы
```

```
# но точно будет являться целым числом
```

```
id(x)
```

Работа с вводом/выводом

Взаимодействие с пользователем может настраиваться через стандартные потоки ввода и вывода `stdin/stdout`.

Для получения входной информации от пользователя можно воспользоваться вызовом функции [input\(...\)](#)

Для вывода информации из программы можно воспользоваться функцией [print\(...\)](#)

Важно: функция `input` блокирует выполнение скрипта и ждет ввода пользователя, для того чтобы продолжить работу. Кроме того сама функция возвращает введенную строку, **НЕ ЗАБЫВАЙТЕ КОНВЕРТИРОВАТЬ ТИПЫ**

Работа с вводом/выводом

```
In [21]: my_name = input("Put your name here: ")  
Put your name here: Sergey
```

```
In [22]: print(my_name)  
Sergey
```

```
In [23]: █
```

Практическая Часть

1. [Возведение числа в степень](#)
2. [Различные операции](#)
3. [Переменные](#)
4. [Ввод/Вывод](#)

ДЗ: Реализовать программу, которая спрашивает у пользователя: имя, фамилию, год рождения. После ввода всех данных программа должна выводить строку следующего вида:

```
"Hello {Name} {Surname} your age is {year} year"
```