

Programming

with  pythonTM

By

Rami Tailakh

Senior Software Engineer and Data Science Practitioner
MSc in Applied Computing and Information Technology

Day-1 Agenda

- Course Overview
- Python Language Overview
- Python and Environment Installation
- Introduction to Coding

Course Outline

Introduction to Python

- Environment Installation
- “Hello World” Example
- Understanding Python variables
- Basic Operators

Data Types

- Declaration
- Use of List, Dictionary and Tuple data types

Flow Control

- Conditional blocks
- For loops; using ranges, lists and dictionaries
- While loops

Python Functions, Modules and Packages

- Code reuse and organization
- Creating and importing modules
- Understanding Packages
- Lambda function

Python Object Oriented Programming

- Concept of class, object and instances
- Constructor, class attributes
- Inheritance

Course Outline-Continue

Advanced Topics

- Comprehensions
- File Operations
- Regular Expression
- Pandas Basics
- GUI Basics
- Flask REST API

Sample Project

- Problem
- Individual/Group Work

Python Overview

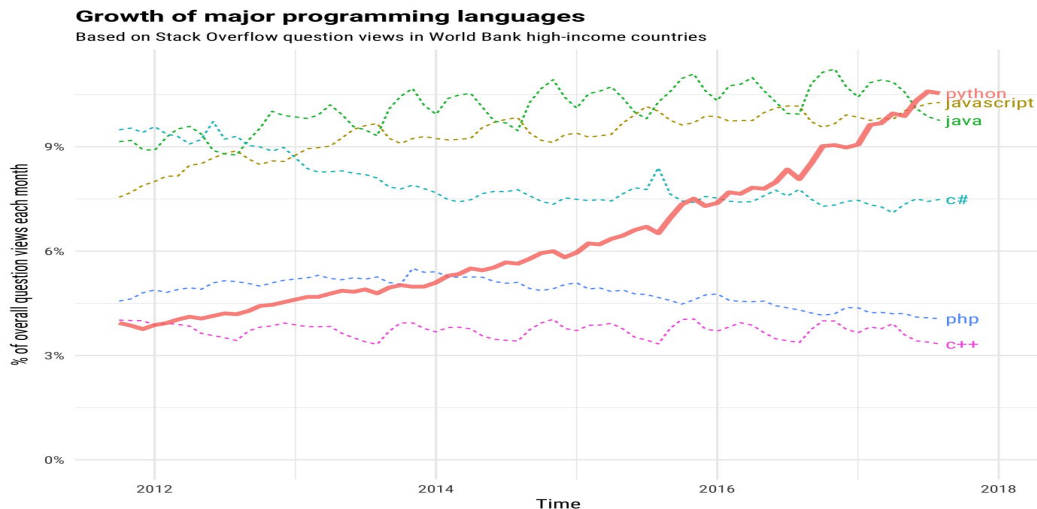
- Programming language; not a *snake* !
- First release in 1991
- Open Source
- Scripting Language
- Object oriented
- Interpreted
- Code readability
- Object oriented
- Small to large-scale projects
- Platform independent
- Extensive number of libraries
- Large and expanding community

Python Overview-Continue

What are the best Languages Tools? Based on "community".



However Developer Stack Overflow visits to Python have grown very quickly



Ref.: <https://stackshare.io/languages>

<https://stackoverflow.blog/2017/09/06/incredible-growth-python/>

Python Overview-Continue

What are the best Languages Tools? Based on analyzing how often language tutorials are searched on Google.

Rank	Change	Language	Share	Trend
1		Python	29.21 %	+4.6 %
2		Java	19.9 %	-2.2 %
3		Javascript	8.39 %	+0.0 %
4		C#	7.23 %	-0.6 %
5		PHP	6.69 %	-1.0 %
6		C/C++	5.8 %	-0.4 %
7		R	3.91 %	-0.2 %
8		Objective-C	2.63 %	-0.7 %
9		Swift	2.46 %	-0.3 %
10		Matlab	1.82 %	-0.2 %

Ref.: <http://pypl.github.io/PYPL.html>

Python Overview-Continue

What are the best Languages Tools? Based on "metrics" such as of coding

Rank	Language	Type	Score
1	Python	🌐 🖥️ ⚙️	100.0
2	Java	🌐 📱 🖥️	96.3
3	C	📱 🖥️ ⚙️	94.4
4	C++	📱 🖥️ ⚙️	87.5
5	R	🖥️	81.5
6	JavaScript	🌐	79.4
7	C#	🌐 📱 🖥️ ⚙️	74.5
8	Matlab	🖥️	70.6
9	Swift	📱 🖥️	69.1
10	Go	🌐 🖥️	68.0

Python Overview-Continue

Popular sites built
with Python

YouTube



Google

Quora

bitly

reddit



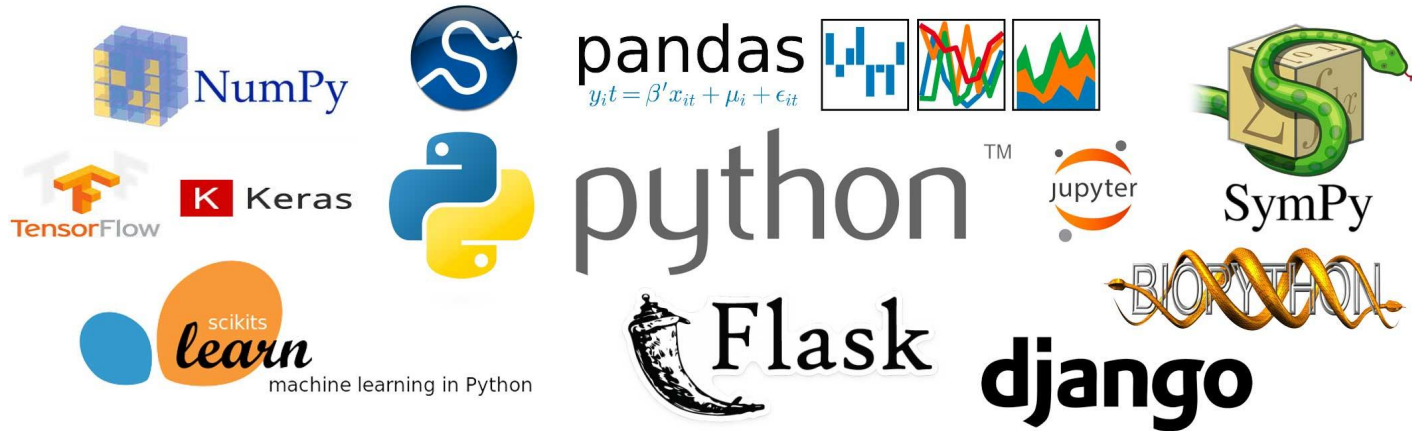
Pinterest

DISQUS

Python Overview-Continue

What are the most popular Python packages?

We offer the following Python training



Development Environment

- **Jupyter Notebook**; Interactive Environment for Development
- **VS-Code or PyCharm**; Finalization and Production and more Powerful for Processing

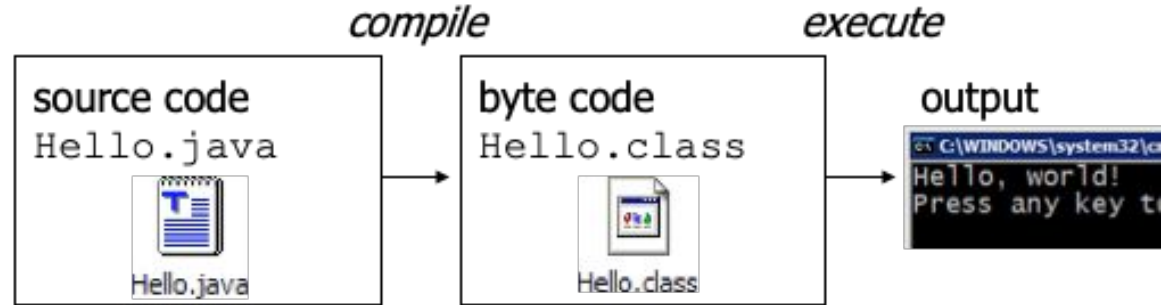
Current Versions of Python

- [Python 2.7](#), deprecated and will not be supported any more after 1 Jan 2020
- [Python 3.7.4](#), documentation released on 08 July 2019
- Python 3.7 will be used

Java vs Python

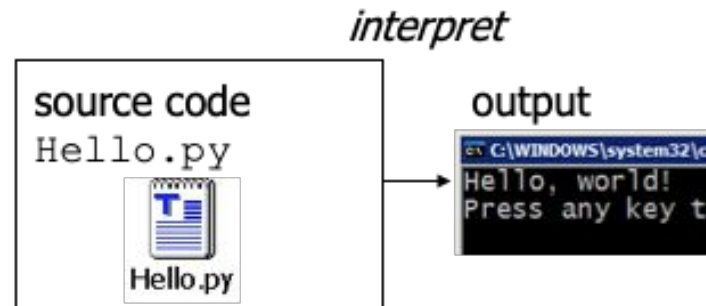
In Java:

```
public class Hello {
    public static void main(String[] args) {
        System.out.println("Hello, World");
    }
}
```



In Python:

```
print('Hello World')
```



Installing & Running Python

First, download the required software.

Download Python:

<https://www.python.org/downloads/>

Download Jupyter:

<https://www.anaconda.com/distribution/>

And

Download PyCharm:

<https://www.jetbrains.com/pycharm/download/#section=windows>

Or

Download PyCharm:

<https://code.visualstudio.com/download>

Coding

Hello World Example

```
print('Hello World')
```


Python Data Types

- **Dynamic Typing:** Python automatically determines the data types
- **Strong Typing:** Every change of a type of a value requires an explicit conversion, e.g.:

```
x = 'My age is '  
y = 30
```

```
print(x + y)
```

TypeError: can only concatenate str (not "int") to str

```
x = 'My age is '  
y = 30
```

```
print(x + str(y))
```

Output: My age is 30

Python Data Types-Basic Types

- **Integers (default for numbers)**

`x = 5 / 2` # Answer is 2, integer division.

- **Floats**

`x = 3.4`

- **Strings**

- "Abc" or 'abc' can be used
- """Abcd""" or '''abc''' triple double-quotes (or single-quotes) can be used for multi-line strings

- **Boolean True/False**

Python Data Types-Advanced Types

- **Lists**

[1, 'abc', 3.4, True]

[0, 1, 2, 3]

- **Tuples**

(1, 'abc', 3.4, True)

- **Dictionaries**

{**"key"**: value}

Naming Rules

- Names are case sensitive
- Names cannot start with a number
- Names can contain letters, numbers, and underscores.

```
var Var _var _2_var_ var_2 VaR
```

- There are some reserved words:

and, assert, break, class, continue, def, del, elif, else, except, exec, finally, for, from, global, if, import, in, is, lambda, not, or, pass, print, raise, return, try, while

Comments

- Line of comment starts with **#**
- It can include a “documentation string”; recommended for the first line of any **function** or **class** definitions, e.g:

```
def my_function(x):  
    """This is the docstring.  
  
    This function does blah blah blah."""  
  
    <implementation>
```

- A line of comment is ignored
- When document comments placed immediately after a function, a class definition, or on top of a module, they are called **docstrings** and made available via the special variable `my_function.__doc__`

String Methods: find

```
string = 'Hi there. This is Python.'
```

```
>>> string.find('.')
```

It finds the start of a substring

Out: **8**

```
>>> string.find('.',10)
```

position 10

It finds the start of a substring after

Out: **24**

Note: If no match, -1 is returned

String Methods: split

```
string = 'Hi there. This is Python.'
```

```
>>> string.split('.') # It splits the text into part with a . as a separator
```

```
Out: ['Hi there', ' This is Python', '']
```

String Methods: join

```
strings = ['This', 'is', 'Python']
```

```
>>> ' '.join(strings)
```

```
Out:      'This is Python'
```


String Methods: format

```
>>> name = 'Python'
```

```
>>> print('Hi there! My name is {}'.format(name))
```

Out: **Hi there! My name is Python.**

String Methods: more methods

```
string = 'Hi there. This is Python.'
```

```
>>> string.startswith('H')
```

Out: **True**

```
>>> string.lower()
```

Out: **'hi there. this is python.'**

```
>>> string.upper()
```

Out: **'HI THERE. THIS IS PYTHON.'**

See also: count, endswith, strip ...

String Operators: in

```
string = 'Hi there. This is Python.'
```

```
>>> 'Python' in string          # If a substring is a part of a text
```

```
Out:      True
```

```
>>> 'python' in string         # It is case sensitive
```

```
Out:      False
```

Conditional Statements

- **if/else**

if [not] *<variable1 == condition1>* [and/or] *<...>*:

 # implementation -1

else:

 # implementation -2

- **elif** # It allows for additional conditions

if *condition*:

elif *another condition*:

·

·

else: # none of the above

Error Handling

- **try/except**

try:

implementation -1

except:

implementation -2

Strings Challenge

Employing ONLY the string methods and operators find a solution of the following challenges:

1. Find if a given word is in lower/upper case
2. Find the length of a string
3. Find all occurrences of a word (from any text) in a given string.