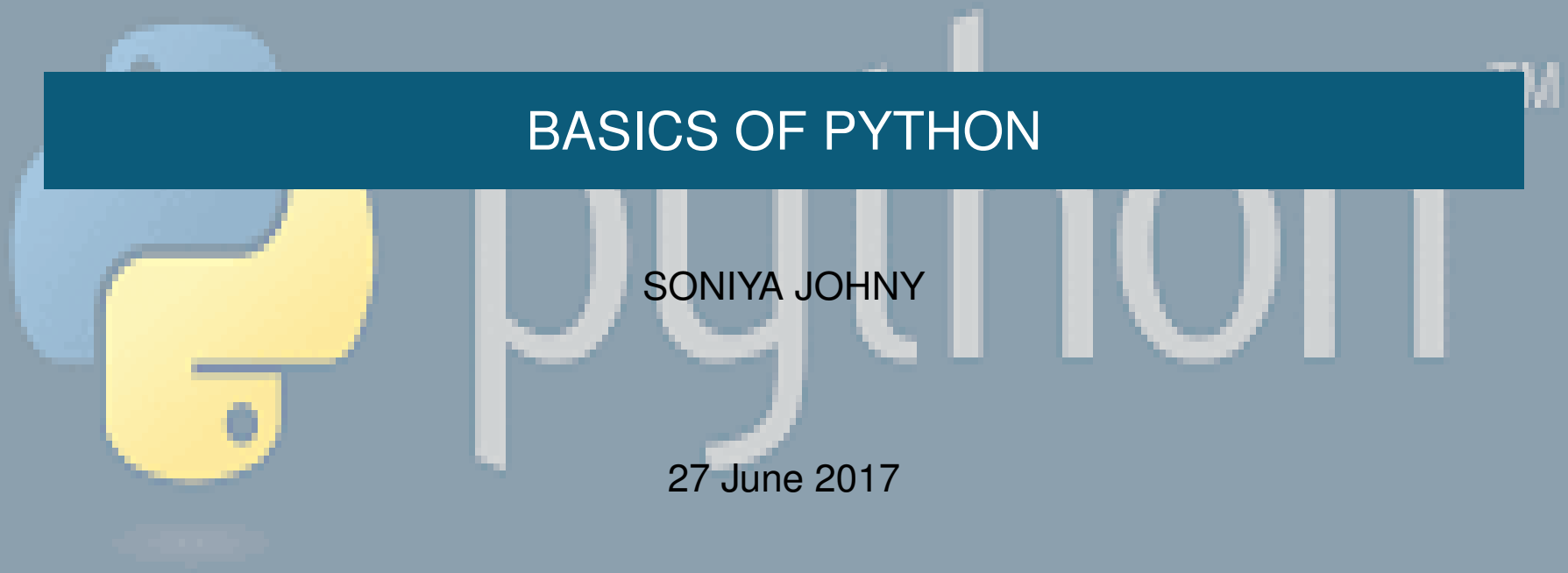


BASICS OF PYTHON

SONIYA JOHNY

27 June 2017

Table Of Contents

Features

- **Simple language which is easier to learn**
  Python has a very simple and elegant syntax. Python makes programming fun and allows us to focus on the solution rather than syntax.
- **Free and open-source**
  We can freely use and distribute Python, even for commercial use. Not only can use and distribute softwares written in it, we can even make changes to the Python's source code.
- **Portability**
  We can move Python programs from one platform to another, and run it without any changes.
- **Object-oriented**
  Everything in Python is an object. Object oriented programming (OOP) helps us to solve a complex problem intuitively.

Environment Setup

- **Simple steps to install Python on Unix/Linux machine.**
  - Open a Web browser and go to https ://www.python.org/downloads/.
  - Follow the link to download zipped source code available for Unix/Linux.
  - Download and extract files.
  - Editing the Modules/Setup file if you want to customize some options.
  - run ./configure script
  - make
  - make install
- This installs Python at standard location /usr/local/bin and its libraries at /usr/local/lib/pythonXX where XX is the version of Python.

Interactive Mode Programming

- Invoking the interpreter without passing a script file as a parameter brings up the following prompt

```
$ python
Python 2.4.3 (#1, Nov 11 2010, 13:34:43)
[GCC 4.1.2 20080704 (Red Hat 4.1.2-48)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

- Type the following text at the Python prompt and press the Enter

```
>>> print "Hello, Python!"
```

- It produces the following result

```
Hello, Python!
```

- Invoking the interpreter with a script parameter begins execution of the script and continues until the script is finished. When the script is finished, the interpreter is no longer active.
  Let us write a simple Python program in a script. Python files have extension .py. Type the following source code in a test.py file

```
print "Hello, Python!"
```

- We assume that you have Python interpreter set in PATH variable. Now, try to run this program as follows

```
$ python test.py
```

- This produces the following result

```
Hello, Python!
```

Variables

- Variables are nothing but reserved memory locations to store values. This means that when you create a variable you reserve some space in memory.
  Based on the data type of a variable, the interpreter allocates memory and decides what can be stored in the reserved memory. Therefore, by assigning different data types to variables, you can store integers, decimals or characters in these variables.

```
#!/usr/bin/python

counter = 100           # An integer assignment
miles   = 1000.0        # A floating point
name    = "John"        # A string

print counter
print miles
print name
```

- This produces the following result

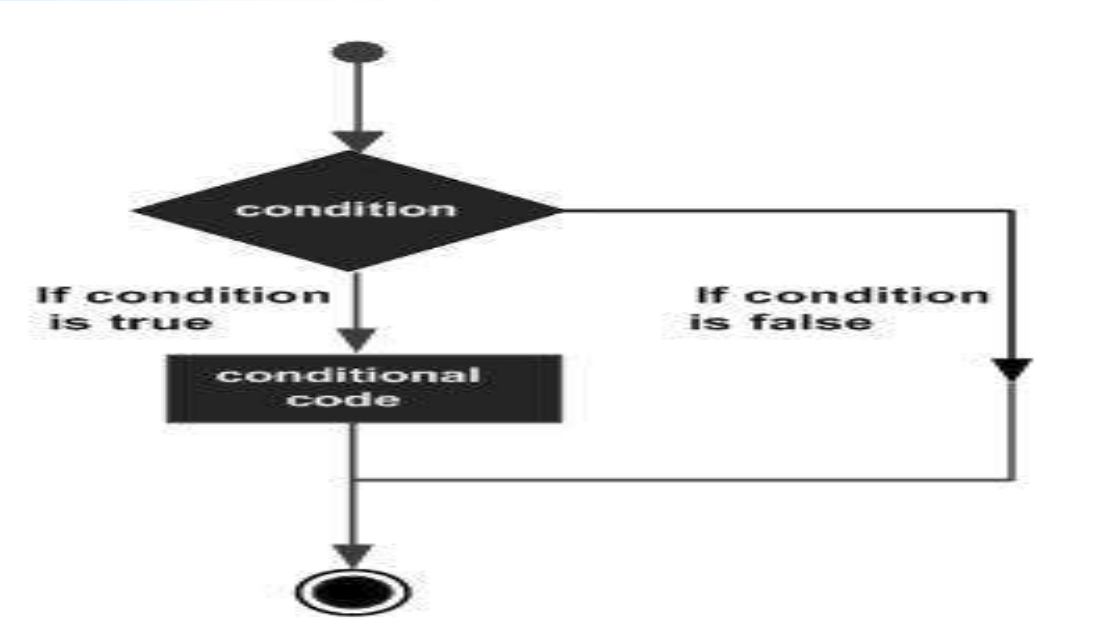
```
100
1000.0
John
```

DataTypes

- Python has five standard data types
  - Numbers
  - String
  - List
  - Tuple
  - Dictionary

Decision Making

- Decision structures evaluate multiple expressions which produce TRUE or FALSE as outcome. You need to determine which action to take and which statements to execute if outcome is TRUE or FALSE otherwise.
- Python programming language assumes any non-zero and non-null values as TRUE, and if it is either zero or null, then it is assumed as FALSE value.



Loops

- A loop statement allows us to execute a statement or group of statements multiple times.
- **while loop** Repeats a statement or group of statements while a given condition is TRUE. It tests the condition before executing the loop body.
- **for loop** Executes a sequence of statements multiple times and abbreviates the code that manages the loop variable.
- **nested loops** You can use one or more loop inside any another while, for or do..while loop.


Thank you!