Getting Started with Python

Vikram Sangat

Python Bootcamp
Nagpur, 2017.

Prerequisites

1. Basic understanding of computer programming terminologies.

2. Writing code in modern programing language is a plus.

   - Ex, Perl, PHP, JAVA, JavaScript etc.

3. Text Editor

Prerequisites

4. System with Python Installed
   - Recommended: Python 3.4 or later

Mode of Programming

- Interactive Mode Programming
- Script Mode Programming

Basic Syntax

1. Identifiers
   - An identifier starts with a letter A to Z or a to z or an underscore (_) followed by zero or more letters, underscores and digits (0 to 9).
2. Reserved Words
   - And, exec, not, as, finally, or etc..

Basic Syntax

1. Identifiers
   - An identifier starts with a letter A to Z or a to z or an underscore (_) followed by zero or more letters, underscores and digits (0 to 9).
2. Reserved Words
   - And, exec, not, as, finally, or etc..

Basic Syntax

3. Lines and Indentation

- Python does not use braces({})
- Multiples of four white spaces

4. Multi-Line Statements

```
total = item_one + \
        item_two + \
        item_three
```

Variables

1. Every thing is a object in python.
    - -Variables, function, code
2. Every object has ID, Type and Value.
    1. ID uniq identifier
    2. Type identifies class of object
    3. Value is contents of object

Basic Operators

1. Arithmetic Operators
   + , - , *, /, %, **, //
2. Comparison (Relational) Operators
   ==,   !=, >, <, >=, <=
3. Assignment Operators
   =, +=, -=, *=, /=, %=, //=, **=
4. Logical Operators
   and, or, not
5. Membership Operators
   in, not in
6. Identity Operators
   is, is not

Conditionals

1. any non-zero and non-null values as TRUE.
2. any zero or null values as FALSE.

- if:
- If: else:
- If: elif: else:

Loops

1. for
2. While

Loop Control Statements

- Break
- Continue

Data Structures

Python has five standard data types −

- Numbers
- String
- List
- Tuple
- Dictionary

Functions

- Function blocks begin with the keyword **def** followed by the **functionname** and parentheses (**( )**).
- You can also define parameters inside these parentheses.
- The code block within every function starts with a colon (:) and is indented.

```
def functionname( parameters ):
    "function_docstring"
    function_suite
    return [expression]
```

Functions Arguments

You can call a function by using the following types of formal arguments −

- Required arguments
- Keyword arguments
- Default arguments
- Variable-length arguments

The Anonymous Functions

You can use the **lambda** keyword to create small anonymous functions

- any number of arguments
- return just one value in the form of an expression
- own local namespace
- a one-line version of a function

Syntax

lambda [arg1 [,arg2,.....argn]]:**expression**

Return

- statement **return** [expression] exits a function.
- A return statement with no arguments is the same as **return None**.

Modules

- a module is a file consisting of Python code
- allows you to logically organize your Python code
- Grouping related code into a module makes the code easier to understand and use.
  - **import** Statement

    import *module1[, module2[,... moduleN]*
  - **from**...**import** Statement

    from <u>modname</u> **import** *name1[, name2[, ... nameN]]*
  - **from**...**import *** Statement

Packages in Python

- a module is a file consisting of Python code
- allows you to logically organize your Python code
- Grouping related code into a module makes the code easier to understand and use.
  - **import** Statement

    import *module1[, module2[,... moduleN]*
  - **from**...**import** Statement

    from modname **import** *name1[, name2[, ... nameN]]*
  - **from**...**import *** Statement