**INTRODUCTION**

...........................................................................................................................................................

**Overview:** Python is a high-level, interpreted, interactive, object-oriented, general purpose scripting language. Python is designed to be highly readable.

**Python is Interpreted:** Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

**Python is Interactive:** You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

**Python is Object-Oriented:** Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

**Python is a Beginner's Language:** Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

**History of Python:** Python was developed by Guido van Rossum (in December 1989)in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

**Who uses PYTHON ?**

|  |  |
| --- | --- |
| **Global Players** | **Local Players** |
| Google  YouTube  Dropbox  Intel/Cisco/HP  NASA | HCL Technologies  MaYo Technologies  Snapdeal |

**What can you do with PYTHON ?**

|  |  |
| --- | --- |
| System Programming  Database programming – SQL, “BIG DATA”  Numeric and scientific programming MATLAB  GUI Programming | Internet Scripting  Rapid Prototyping  Gaming |

**Technical Strengths**

|  |  |
| --- | --- |
| Object oriented.  It is free  It is Portable.  Powerful | Dynamic typing  Automatic memory management  Built in object type – list, dictionary,  Libraries, third party utilities. |

**Getting Python :** The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python: [**http://www.python.org/**](http://www.python.org/)

Python Documentation **Website :**[***www.python.org/doc/***](http://www.python.org/doc/)

**Install Python:** Follow this link *(*[*http://www.python.org/download/*](http://www.python.org/download/)*)* for installation on various operating system such as window linux, mac etc.

**Setting up PATH** The path variable is named as PATH in Unix or Path in Windows (Unix is case-sensitive; Windows is not).

**Setting path at Unix/Linux:** To add the Python directory to the path for a particular session in **Unix:**

**In the csh shell:** type ***setenv PATH "$PATH:/usr/local/bin/python"*** and press Enter.

**In the bash shell (Linux):** type ***export PATH="$PATH:/usr/local/bin/python"*** and press Enter.

**In the sh or ksh shell:** type ***PATH="$PATH:/usr/local/bin/python"***and press Enter.

**Note: /usr/local/bin/python is the path of the Python directory**

**Setting path at Windows** To add the Python directory to the path for a particular session in Windows: At the command prompt : type ***path %path%;C:\Python*** and press Enter. **Note: C:\Python is the path of the Python directory**

**Integrated Development Environment(IDE)**

1.IDLE

2.Spyder

3. PyCharm

**PYTHON BASIC SYNTAX**

**Python Identifiers**

A Python identifier is a name used to identify a variable, function, class, module or other object. Python is a case sensitive programming language. Here are naming conventions for Python identifiers −

Class names start with an uppercase letter. All other identifiers start with a lowercase letter. Starting an identifier with a single leading underscore indicates that the identifier is private. Starting an identifier with two leading underscores indicates a strongly private identifier. If the identifier also ends with two trailing underscores, the identifier is a language-defined special name.

**Reserved Words:**All the Python keywords contain lowercase letters only.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *And* | *Exec* | *Not* | *Except* |  | *Lambda* | *yield* | *Else* | *is* | *With* |
| *Assert* | *Finally* | *Or* | *Def* |  | *If* | *return* | *Continue* | *Global* | *Raise* |
| *Break* | *For* | *Pass* | *Del* |  | *Import* | *try* | *Elif* | *in* | *While* |
| *Class* | *From* | *Print* |  |  |  |  |  |  |  |

**Lines and Indentation**

Python provides no braces to indicate blocks of code for class and function definitions or flow control. Blocks of code are denoted by line indentation.

|  |  |
| --- | --- |
| begin ... end  do ... done  { ... }  if ... fi | C:\Users\Dev\Pictures\Screenshots\Python\IndentaionBlock.jpg |

if True:

print "True"

else:

print "False"

However, the following block generates an error −

if True:

print "Answer"

print "True"

else:

print "Answer"

print "False"

**Multi-Line Statements**

Statements in Python typically end with a new line. Python does, however, allow the use of the line continuation character (\) to denote that the line should continue.

**For example:**

**total = item\_one + \**

**item\_two + \**

**item\_three**

Statements contained within the [], {}, or () brackets do not need to use the line continuation character.

**For example:**

**days = ['Monday', 'Tuesday', 'Wednesday',**

**'Thursday', 'Friday']**

**Quotation in Python**

Python accepts single ('), double (") and triple (''' or """) quotes to denote string literals, as long as the same type of quote starts and ends the string.

**word = 'word'**

**sentence = "This is a sentence."**

**paragraph = """This is a paragraph. It is**

**made up of multiple lines and sentences."""**

**Comments in Python**

A hash sign (#) that is not inside a string literal begins a comment.

**Python Variable Types**

Variables are nothing but reserved memory locations to store values. This means that when you create a variable you reserve some space in memory.

Assigning Values to Variables

Python variables do not need explicit declaration to reserve memory space. The declaration happens automatically when you assign a value to a variable. The equal sign (=) is used to assign values to variables.

***counter = 100 # An integer assignment***

***miles = 1000.0 # A floating point***

***name = "John" # A string***

***print counter***

***print miles***

***print name***

Here, 100, 1000.0 and "John" are the values assigned to *counter*, *miles*, and *name* variables, respectively.

This produces the following result −

***100***

***1000.0***

***John***

**Multiple Assignment**

Python allows you to assign a single value to several variables simultaneously.

***Example1:***

***a = b = c = 1***

***Example2:***

***a, b, c = 1, 2, "john"***