# PYTHON QUICK GUIDE

http://www.tutorialspoint.com/python/python_quick_guide.htm

## PYTHON OVERVIEW:

Python is a high-level, interpreted, interactive and object oriented-scripting language.

- **Python is Interpreted**
- **Python is Interactive**
- **Python is Object-Oriented**
- **Python is Beginner's Language**

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python's feature highlights include:

- **Easy-to-learn**
- **Easy-to-read**
- **Easy-to-maintain**
- **A broad standard library**
- **Interactive Mode**
- **Portable**
- **Extendable**
- **Databases**
- **GUI Programming**
- **Scalable**

## GETTING PYTHON:

The most up-to-date and current source code, binaries, documentation, news, etc. is available at the official website of Python:

**Python Official Website :** http://www.python.org/

You can download the Python documentation from the following site. The documentation is available in HTML, PDF, and PostScript formats.

**Python Documentation Website :** www.python.org/doc/

## FIRST PYTHON PROGRAM:

### Interactive Mode Programming:

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

```
root# python
Python 2.5 (r25:51908, Nov  6 2007, 16:54:01)
[GCC 4.1.2 20070925 (Red Hat 4.1.2-27)] on linux2
Type "help", "copyright", "credits" or "license" for more info.
>>>
```

Type the following text to the right of the Python prompt and press the Enter key:

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

This will produce following result:

```
Hello, Python!
```

## PYTHON IDENTIFIERS:

A Python identifier is a name used to identify a variable, function, class, module, or other object. 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).

Python does not allow punctuation characters such as @, $, and % within identifiers. Python is a case sensitive programming language. Thus **Manpower** and **manpower** are two different identifiers in Python.

Here are following identifier naming convention for Python:

- Class names start with an uppercase letter and all other identifiers with a lowercase letter.
- Starting an identifier with a single leading underscore indicates by convention that the identifier is meant to be 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:

The following list shows the reserved words in Python. These reserved words may not be used as constant or variable or any other identifier names.

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
with
yield

## LINES AND INDENTATION:

One of the first caveats programmers encounter when learning Python is the fact that there are no braces to indicate blocks of code for class and function definitions or flow control. Blocks of code are denoted by line indentation, which is rigidly enforced.

The number of spaces in the indentation is variable, but all statements within the block must be indented the same amount. Both blocks in this example are fine:

```
if True:
    print "True"
else:
  print "False"
```

However, the second block in this example will generate 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.

The triple quotes can be used to span the string across multiple lines. For example, all the following are legal:

```
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. All characters after the # and up to the physical line end are part of the comment, and the Python interpreter ignores them.

```
#!/usr/bin/python

# First comment
print "Hello, Python!";  # second comment
```

This will produce following result:

```
Hello, Python!
```

A comment may be on the same line after a statement or expression: