Learn Python: Basic Syntax

Abrar Hussain

March 1 2015

Python has many similarities to other languages such as C and Java. There are some differences between the languages, so this section is meant to get you up to speed on the syntax that is expected in Python.

Interactive and Script Mode Programming

When we open up IDLE we're presented with the interactive mode with the following prompt. We can keep entering statements into the prompt similar to our "hello world" program.

```
Python 3.4.2 (v3.4.2:ab2c023a9432, Oct  6 2014, 22:15:05) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
```

We can create a new file and run it in order to program in script mode. When we decide to save a new .py file and "Run Module", the interpreter begins execution of the script until the script is finished. When the script is finished, the interpreter is no longer active.

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. So, Human and human are two different identifiers in Python. Here are some identifier naming convention for Python:

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

There are some other conventions that come into play with more advanced uses of Python.

Lines and Indentation

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 can vary, 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 "Truth"
    print "True"
else:
```

```
    print "Truth"
  print "False"
```

So, all of the lines in sequences with a similar number of spaces would form a block.

Blank Lines and Comments

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. If we run a script with the following:

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

We will get:

```
Hello, Python!
```

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

```
name = "Morgan" # this is also a comment
```

You can comment multiple lines as follows:

```
# This is a comment.
# This is a comment as well
# This is another comment.
# They're all comments!
```

A line containing only whitespace, possibly with a comment, is known as a blank line and Python totally ignores it. In an interactive interpreter session, you must enter an empty physical line to terminate a multiline statement (if you use the interpreter you can notice this).

Questions?

If you have any questions, feel free to make a post in the discussions.