Python for JAVA Developers: Basics V 0.1

By Akash Panchal

1 Basic Syntax

1.1 End of Statements

Unlike Java to end a statement in Python, don't have to type in a semicolon, you simply press *Enter*.

```
message1 = 'Hello World!'
message2 = "Python gives no missing semicolon error!"

# Instead of System.out.print, we use print
print message1 # print 'Hello World!' on the console output
print "Hello Python!"
```

2 Data Types

Python sets the variable type based on the value that is assigned to it. Unlike more riggers languages, Python will change the variable type if the variable value is set to another value

```
var = 123 # This will create a number integer assignment
var = 'john' # the 'var' variable is now a string type.
```

2.1 Numbers

Most of the time using the standard Python number type is fine. Python will automatically convert a number from one type to another if it needs.

Type	Java	Python	Description
int	int a = 11	a = 11	Signed Integer
long	long a = 1712L	a = 1712L	(L) Long integers
float	float a = 19.91	a = 19.91	(.) Floating point values
complex	- - -	a = 3.14J	(J) integer [0 to 255]

2.2 String

Create string variables by enclosing characters in quotes. Python uses single quotes ' double quotes " and triple quotes """ to denote literal strings. Only the triple quoted strings """ will automatically continue across the end of line statement.

```
firstName = 'Jane'
lastName = "Doe"
message = """This is a string that will span across multiple
     lines. Using newline characters and no spaces for the
     next lines."""
```

2.3 List

Lists are a very useful variable type in Python. A list can contain a series of values. List variables are declared by using brackets [ ] following the variable name.

```
A = [] # This is a blank list variable
B = [1, 23, 45, 67] # creates an initial list of 4 numbers.
C = [2, 4, 'john'] # can contain different variable types.
```

2.4 Tuple

Tuples are a group of values like a list and are manipulated in similar ways. But, tuples are fixed in size once they are assigned.

```
myGroup = ('Lion', 'Tiger', 'Elephant', 'Giraffe')
```

2.5 Dictionary

Dictionaries in Python are lists of *Key*:*Value* pairs. This is a very powerful datatype to hold a lot of related information that can be associated through *keys*.

```
room_num = {'john': 121, 'tom': 307}
room_num['john'] = 432 # set the value associated with the
     'john' key to 432
print (room_num['tom']) # print the value of the 'tom' key.
room_num['isaac'] = 345 # Add a new key 'isaac' with the
     associated value
print (room_num.keys()) # print out a list of keys in the
     dictionary
print ('isaac' in room_num) # test to see if 'issac' is in the
     dictionary. This returns true.
```

3 Variables

3.1 Declaration

variables are created the first time a value is assigned to them.

```
number = 11
string = "This is a string"
```

You declare multiple variables by separating each variable name with a comma.

```
a, b = True, False
```

3.2 Assigning Values

```
a=300
```

The same value can be assigned to multiple variables at the same time:

```
a = b = c = 1
```

And multiple variables can be assigned different values on a single line:

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

This is the same as:

```
a = 1
b = 2
c = "john"
```

4 Conditionals

4.1 if

```
var1 = 250
if var1 == 250 : print "The value of the variable is 250"
```

4.2 if..else

```
var1 = 250
if var1 == 0 :
   MyLayerColor = 'vbRed'
   MyObjectColor = 'vbBlue'
else :
   MyLayerColor = 'vbGreen'
   MyObjectColor = 'vbBlack'
print MyLayerColor
```

4.3 if..elif..elif..else

```
var1 = 0
if var1 == 0 :
   print "This is the first " + str(var1)
elif var1 == 1 :
   print "This is the second " + str(var1)
elif var1 == 2 :
   print "This is the third " + str(var1)
else :
   print "Value out of range!"
```

5 Looping

5.1 For Loop

Python will automatically increments the counter (x) variable by 1 after coming to end of the execution block.

```
for x in range(0, 5):
   print "It's a loop: " + str(x)
```

Increase or decrease the counter variable by the value you specify.

```
# the counter variable j is incremented by 2 each time the
     loop repeats
for j in range(0, 10, 2):
   print "We're on loop " + str(j)

# the counter variable j is decreased by 2 each time the loop
     repeats
for j in range(10, 0, -2):
   print "We're on loop " + str(j)
```

You can exit any for statement before the counter reaches its end value by using the $\boxed{break}$ statement.

5.2 While Loop

Simple example of while loop.

```
var1 = 3
while var1 < 37:
   var1 = var1 * 2
   print var1
print "Exited while loop."
```

Another example of while loop with break statement.

```
while True:
   n = raw_input("Please enter 'hello':")
   if n.strip() == 'hello':
      break
```

6 Operators

6.0.1 Operator Precedence

Operator Precedence is same as that of JAVA, let's revise it in python.Arithmetic operators are evaluated first, comparison operators are evaluated next, and logical operators are evaluated last.

Arithmetic operators are evaluated in the following order of precedence.

Description	Symbol
Exponentiation	**
Unary negation	-
Multiplication	*
Division	/
Modulus arithmetic	mod
Addition	+
Subtraction	-
String concatenation	&

Logical operators are evaluated in the following order of precedence.

Description	Symbol
Logical negation	not
Logical conjunction	and
Logical disjunction	or
Logical exclusionto	xor
Logical equivalence	eqv
Logical implication	imp

Comparison operators all have equal precedence; that is, they are evaluated in the left-to-right order in which they appear.

Description	Symbol
Less than	<
Greater than	>
Less than or equal to	<=
Greater than or equal to	>=
Equality	==
Inequality	!=
Object equivalence	is

7 The End

That’s all Folks.s