Python_tutorials

July 26, 2016

In [1]:
```
print "Welcome to Python!"
print "Hi i am happy to start off with python "
```

```
Welcome to Python!
Hi i am happy to start off with python
```

In [4]:
```
#Declaring variables
my_int = 7
my_float = 1.23
my_bool = True

print my_int
print my_float
print my_bool
```

```
7
1.23
True
```

In [5]:
```
def spam():
eggs = 12
return eggs

print spam()
```

```
  File "<ipython-input-5-597b13143f1d>", line 2
eggs = 12
   ^
IndentationError: expected an indented block
```

In [6]:
```
#indentation
def spam():
    eggs = 12
    return eggs

print spam()
```

```
12
```

In [9]:
```
"""
Mathematical operations
"""
```

```
        count_to = 1234560 + 9876540
        print count_to

        eggs = 10 ** 2
        print eggs

        spam = 5 % 4
        print spam
```

```
11111100
100
1
```

```
In [13]: fifth_letter = "PYTHON"[4]
         print fifth_letter

         parrot = "Norwegian Blue"
         len(parrot)
         print len(parrot)
         print parrot.lower()
         print parrot.upper()
```

```
O
14
norwegian blue
NORWEGIAN BLUE
```

```
In [15]: pi = 3.14
         str(pi)
         print pi
         print "The value of pi is around " + str(pi)
```

```
3.14
The value of pi is around 3.14
```

```
In [17]: string_1 = "Camelot"
         string_2 = "place"

         print "Let's not go to %s. its a silly %s." % (string_1, string_2)
```

```
Let's not go to Camelot. its a silly place.
```

```
In [18]: name = raw_input("What is your name?")
         quest = raw_input("What is your quest?")
         color = raw_input("What is your favorite color?")

         print "Ah, so your name is %s, your quest is %s, " \
         "and your favorite color is %s." % (name, quest, color)
```

```
What is your name?VIKI
What is your quest?TO TEACH PYTHON
What is your favorite color?ORANGE
Ah, so your name is VIKI, your quest is TO TEACH PYTHON, and your favorite color is ORANGE.
```

```
In [20]: from datetime import datetime

         now = datetime.now()
         print now
         print now.year
         print now.month
         print now.day
```

```
2016-07-17 13:16:08.663654
2016
7
17
```

```
In [22]: print '%s/%s/%s' % (now.year, now.month, now.day)

         print '%s/%s/%s %s:%s:%s' % (now.month, now.day, now.year, now.hour, now.minute, now.second)
```

```
2016/7/17
7/17/2016 13:16:8
```

```
In [27]: response = 'Y'

         answer = "Left"
         if answer == "Left":
             print "T"
```

```
T
```

```
In [31]: def greater_less_equal_5(answer):
             if answer>5:
                 return 1
             elif answer< 5:
                 return -1
             else:
                 return 0

         print greater_less_equal_5(4)
         print greater_less_equal_5(5)
         print greater_less_equal_5(6)
```

```
-1
0
1
```

```
In [34]: import math
         print math.sqrt(25)
```

```
5.0
```

```
In [36]: #Import sqrt from math module
         from math import sqrt
         sqrt(25)
```

```
Out[36]: 5.0
```

```
In [35]: # Import *everything* from the math module
         from math import *
         sqrt(25)
```

```
Out[35]: 5.0

In [37]: def biggest_number(*args):
             print max(args)
             return max(args)

         def smallest_number(*args):
             print min(args)
             return min(args)

         def distance_from_zero(arg):
             print abs(arg)
             return abs(arg)


         biggest_number(-10, -5, 5, 10)
         smallest_number(-10, -5, 5, 10)
         distance_from_zero(-10)

10
-10
10

Out[37]: 10

In [38]: print type(42)
         print type(4.2)
         print type('spam')

<type 'int'>
<type 'float'>
<type 'str'>

In [42]: #list and dictionary

         #Introduction to list
         zoo_animals = ["pangolin", "cassowary", "sloth", "monkey"];
         # One animal is missing!

         if len(zoo_animals) > 3:
                 print "The first animal at the zoo is the " + zoo_animals[0]
                 print "The second animal at the zoo is the " + zoo_animals[1]
                 print "The third animal at the zoo is the " + zoo_animals[2]
                 print "The fourth animal at the zoo is the " + zoo_animals[3]

         suitcase = ["sunglasses", "hat", "passport", "laptop", "suit", "shoes"]

         first  = suitcase[0:2]  # The first and second items (index zero and one)
         middle = suitcase[2:4]  # Third and fourth items (index two and three)
         last   = suitcase[4:6]   # The last two items (index four and five)

         print first
         print middle
         print last
```

```
        animals = "catdogfrog"
        cat  =  animals[:3]   # The first three characters of animals
        dog  =  animals[3:6]             # The fourth through sixth characters
        frog =  animals[6:11] # From the seventh character to the end

        print cat
        print dog
        print frog
```

```
The first animal at the zoo is the pangolin
The second animal at the zoo is the cassowary
The third animal at the zoo is the sloth
The fourth animal at the zoo is the monkey
['sunglasses', 'hat']
['passport', 'laptop']
['suit', 'shoes']
cat
dog
frog
```

```
In [44]: animals = ["aardvark", "badger", "duck", "emu", "fennec fox"]
         duck_index = animals.index("duck")
         animals.insert(duck_index ,"cobra")
         print animals
```

```
['aardvark', 'badger', 'cobra', 'duck', 'emu', 'fennec fox']
```

```
In [45]: start_list = [5, 3, 1, 2, 4]
         square_list = []

         for number in start_list:
             square_list.append(number**2)
             square_list.sort()

         print square_list
```

```
[1, 4, 9, 16, 25]
```

```
In [46]: # Assigning a dictionary with three key-value pairs to residents:
         residents = {'Puffin' : 104, 'Sloth' : 105, 'Burmese Python' : 106}

         print residents['Puffin']
         print residents['Sloth']
         print residents['Burmese Python']
```

```
104
105
106
```

```
In [48]: backpack = ['xylophone', 'dagger', 'tent', 'bread loaf']
         backpack.remove("dagger")
         print backpack
```

```
['xylophone', 'tent', 'bread loaf']
```

```
In [50]: inventory = {
             'gold' : 500,
             'pouch' : ['flint', 'twine', 'gemstone'], # Assigned a new list to 'pouch' key
             'backpack' : ['xylophone','dagger', 'bedroll','bread loaf']
         }

         # Adding a key 'burlap bag' and assigning a list to it
         inventory['burlap bag'] = ['apple', 'small ruby', 'three-toed sloth']

         # Sorting the list found under the key 'pouch'
         inventory['pouch'].sort()

         print inventory
```

```
{'backpack': ['xylophone', 'dagger', 'bedroll', 'bread loaf'], 'pouch': ['flint', 'gemstone', 'twine'],
```

```
In [51]: n = [1, 3, 5]
         # Append the number 4 here
         n.append(4)
         print n
```

```
[1, 3, 5, 4]
```

```
In [52]: n = [1, 3, 5]
         # Remove the first item in the list here
         n.remove(1)
         print n
```

```
[3, 5]
```

```
In [53]: n = [[1, 2, 3], [4, 5, 6, 7, 8, 9]]
         # Add your function here

         def flatten(lists):
             results=[]
             for numbers in lists:
                 for i in numbers:
                     results.append(i)
             return results

         print flatten(n)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
In [54]: count = 0

         print type(count)

         if count < 5:
             print "Hello, I am an if statement and count is", count

         while count < 10:
             print "Hello, I am a while and count is", count
             count += 1
```

```
<type 'int'>
Hello, I am an if statement and count is 0
```

```
Hello, I am a while and count is 0
Hello, I am a while and count is 1
Hello, I am a while and count is 2
Hello, I am a while and count is 3
Hello, I am a while and count is 4
Hello, I am a while and count is 5
Hello, I am a while and count is 6
Hello, I am a while and count is 7
Hello, I am a while and count is 8
Hello, I am a while and count is 9
```

```
In [55]: count = 0

         while True:
             print count
             count += 1
             if count >= 10:
                 break
```

```
0
1
2
3
4
5
6
7
8
9
```

```
In [57]: numbers  = [7, 9, 12, 54, 99]

         print "This list contains: "

         for num in numbers:
             print num

         print "**************"

         for num in numbers:
             print num**2
```

```
This list contains:
7
9
12
54
99
**************
49
81
144
2916
9801
```

```
In [62]: grades = [100, 100, 90, 40, 80, 100, 85, 70, 90, 65, 90, 85, 50.5]

         def grades_sum(scores):
             total= 0
             for item in scores:
                 total += item
             return total

         print grades_sum(grades)

         def grades_average(grades):
             s = grades_sum(grades)
             a = s/float(len(grades))
             return a

         print grades_average(grades)


         def grades_variance(scores):
             average = grades_average(scores)
             variance = 0
             for score in scores:
                 variance =  (average - score)**2 + variance
             result = variance/float(len(scores))
             return result

         def grades_std_deviation(variance):
             return variance ** 0.5

         variance = grades_variance(grades)
         print variance
         print grades_std_deviation(variance)
```

```
1045.5
80.4230769231
334.071005917
18.2776094147
```

```
In [69]: my_dict = {
             "Name" : "Viki",
             "Age" : 21
             }
         print my_dict.items()
         print my_dict.keys()
         print my_dict.values()

         for i in my_dict.keys():
             print my_dict[i]

         for i in my_dict:
             print i, my_dict[i]
```

```
[('Age', 21), ('Name', 'Viki')]
['Age', 'Name']
[21, 'Viki']
```

```
21
Viki
Age 21
Name Viki
```

```
In [70]: cubes_by_four = [ x**3 for x in range(1,11) if (x**3)%4 ==0]
         print cubes_by_four
```

```
[8, 64, 216, 512, 1000]
```

```
In [72]: to_21 = range(1,22)
         odds = to_21[::2]
         middle_third= to_21[7:14]

         print to_21
         print odds
         print middle_third
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21]
[1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21]
[8, 9, 10, 11, 12, 13, 14]
```

```
In [73]: squares = [x**2 for x in range(1 ,11)]
         print squares
         print filter(lambda x:x >=30 and x <=70 ,squares  )
```

```
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
[36, 49, 64]
```

```
In [74]: garbled = "IXXX aXXmX aXXXnXoXXXXXtXhXeXXXXrX sXXXXeXcXXXrXeXt mXXeXsXXXsXaXXXXXXgXeX!XX"
         message = filter( lambda x:x != 'X' , garbled)
         print message
```

```
I am another secret message!
```

```
In [77]: print 5 >> 4  # Right Shift
         print 5 << 1  # Left Shift
         print 8 & 5   # Bitwise AND
         print 9 | 4   # Bitwise OR
         print 12 ^ 42 # Bitwise XOR
         print ~88     # Bitwise NOT
```

```
0
10
0
13
38
-89
```

```
In [78]: print 0b1,    #1
         print 0b10,   #2
         print 0b11,   #3
         print 0b100,  #4
         print 0b101,  #5
         print 0b110,  #6
         print 0b111   #7
         print "******"
         print 0b1 + 0b11
         print 0b11 * 0b11
```

```
1 2 3 4 5 6 7
******
4
9
```

```
In [79]: print bin(2)
         print bin(3)
         print bin(4)
         print bin(5)
```

```
0b10
0b11
0b100
0b101
```

```
In [81]: print int("1",2)
         print int("10",2)
         print int("111",2)
         print int("0b100",2)
         print int(bin(5),2)
```

```
1
2
7
4
5
```

```
In [82]: shift_right = 0b1100
         shift_left = 0b1

         # Your code here!
         shift_right = shift_right >> 2
         shift_left = shift_left << 2

         print bin(shift_right)
         print bin(shift_left)
```

```
0b11
0b100
```

```
In [76]: class Fruit(object):
             """A class that makes various tasty fruits."""
             def __init__(self, name, color, flavor, poisonous):
                 self.name = name
                 self.color = color
                 self.flavor = flavor
                 self.poisonous = poisonous

             def description(self):
                 print "I'm a %s %s and I taste %s." % (self.color, self.name, self.flavor)

             def is_edible(self):
                 if not self.poisonous:
                     print "Yep! I'm edible."
                 else:
                     print "Don't eat me! I am super poisonous."
```

```
lemon = Fruit("lemon", "yellow", "sour", False)
lemon.description()
lemon.is_edible()
```

```
I'm a yellow lemon and I taste sour.
Yep! I'm edible.
```

```
In [85]: class ShoppingCart(object):
             """Creates shopping cart objects
             for users of our fine website."""

             items_in_cart = {}

             def __init__(self, customer_name):
                 self.customer_name = customer_name

             def add_item(self, product, price):
                 """Add product to the cart."""
                 if not product in self.items_in_cart:
                     self.items_in_cart[product] = price
                     print product + " added."
                 else:
                     print product + " is already in the cart."

             def remove_item(self, product):
                 """Remove product from the cart."""
                 if product in self.items_in_cart:
                     del self.items_in_cart[product]
                     print product + " removed."
                 else:
                     print product + " is not in the cart."

         my_cart = ShoppingCart("viki")
         my_cart.add_item("qwe",12)
         my_cart.add_item("qwe1",123)
         my_cart.add_item("qwe2",212)
         my_cart.add_item("qwe3",122)
         my_cart.add_item("qwe4",121)
         my_cart.remove_item("qwe5")
         my_cart.remove_item("qwe3")
```

```
qwe added.
qwe1 added.
qwe2 added.
qwe3 added.
qwe4 added.
qwe5 is not in the cart.
qwe3 removed.
```

```
In [86]: #Inheritance
         class Customer(object):
             """Produces objects that represent customers."""
             def __init__(self, customer_id):
                 self.customer_id = customer_id
```

```
        def display_cart(self):
            print "I'm a string that stands in for the contents of your shopping cart!"

    class ReturningCustomer(Customer):
        """For customers of the repeat variety."""
        def display_order_history(self):
            print "I'm a string that stands in for your order history!"

    monty_python = ReturningCustomer("ID: 12345")
    monty_python.display_cart()
    monty_python.display_order_history()
```

```
I'm a string that stands in for the contents of your shopping cart!
I'm a string that stands in for your order history!
```

```
In [88]: #Override
    class Employee(object):
        def __init__(self, name):
            self.name = name
        def greet(self, other):
            print "Hello, %s" % other.name

    class CEO(Employee):
        def greet(self, other):
            print "Get back to work, %s!" % other.name

    ceo = CEO("Emily")
    emp = Employee("Steve")
    emp.greet(ceo)
    ceo.greet(emp)
```

```
Hello, Emily
Get back to work, Steve!
```

```
In [90]: #superclass
    class Employee(object):
        """Models real-life employees!"""
        def __init__(self, employee_name):
            self.employee_name = employee_name

        def calculate_wage(self, hours):
            self.hours = hours
            return hours * 20.00

    # Add your code below!
    class PartTimeEmployee(Employee):
        def calculate_wage(self,hours):
            self.hours = hours
            return self.hours *12

        def full_time_wage(self,hours):
            return super(PartTimeEmployee,self).calculate_wage(hours)

    milton = PartTimeEmployee("mskfj")
    print milton.full_time_wage(10)
```

```
200.0
```