lec6

August 12, 2016

```
In [ ]: #Today we are going to review functions.
        #We are going to go over how one function
        #calls another function and we are going to review
        #some functions from lab 2.
        #In lab, you are going to have new
        #exercises of writing simple functions

        #input--->function--->output
```

```
In [ ]: #Write a function called multiplyByTwo
        #that takes in an int and
        #returns the input * 2.

        def multiplyByTwo(z):
            return z*2

        #what is the name of the function? multiplyByTwo
        #what is the input? z
        #what is the output z*2
        #the function returns the output which is z*2
```

```
In [7]: #Now we want to add 3 to a number and multiply it by 2
        #i.e. We want to create a function called
        #multiplyByTwoAddThree that takes in a number
        #and returns the (number +3)*2.
        def addThree(z):
            return z+3

        #One function can call another function.
        #Here, multiplyByTwoAddThree calls another function
        #called addThree. When we call multiplyByTwoAddThree
        #with the input 5, multiplyByTwoAddThree calls
        #another function addThree. This function adds
        #3 to its input and returns the output.

        def multiplyByTwoAddThree(z):
            return addThree(z)*2
        y=multiplyByTwoAddThree(5)
        print y

16


In [52]: #printing something is different from returning
         #something. In this example, when we print y
         #what does it print? why?
         #what happens if we call
         #y=multiplyByTwo(5)
         #print y
         #why?
         def multiplyByTwo(z):
             if type(z)==int:
                 return  z*2
             else:
                 print 'I only want integer output'
                 return
         y=multiplyByTwo(5)
         print y

10


In [ ]: #Having more than one function
        def passing_grade(h):
            if h>50:
                print 'good'#True  <--Some people are confused by print Vs. return
                return True
            else:
                print 'bad'#False
                return False
        y=passing_grade(100)
        print y

In [59]: #One function can call another function. What does this function do?
         #what is printed?
         def candy_for_grade(g):
             if passing_grade(g):
                 return 'candy'
             else:
                 return 'no_candy'
         y=candy_for_grade(51)
         print y

good
candy
```

```
In [60]: #Having more than one function
         def passing_grade(h):
             if h>50:
                 print 'good'#True  <--Some people are confused by print Vs. retur
                 #return True
             else:
                 print 'bad'#False
                 return False
         y=passing_grade(100)
         print y

         #Now what is printed? why?

good
None
```

```
In [ ]: #Can you give me a function called reverse, that takes in an input string
        #And returns the reversed version of the string.
        #For example, if the input is abcde it returns edcba.
        #This will give you an exercise using len and range
```

```
In [30]: '''
         Lets say input_string='abcde'
         Before you answer the question, you should try
         to figure out how the function would work on paper.
         what do we want to do?
         -->go through all the characters in input_string
         one by one. (for loop)
         -->Start from e (last character) and go to
         the first character.
            --start here:
                 input_string[len(input_string)-1]
                 and also go to
                 input_string[len(input_string)-1-1]
                 input_string[len(input_string)-1-2]
                 .....
                 --do this until you get to
                 input_string[0]

            --end here:
                 input_string[0]
         For each of these characters,
         add them to an output string
         When you are done, return the output string.
         '''

Out[30]: [0, 1, 2, 3, 4]
```

```
In [41]: input_string='abcde'
         output_string=''
         for x in range(len(input_string)):
           output_string += input_string[len(input_string)-x-1]
         print output_string

edcba
```

```
In [51]: def reverse(x):
             y=''
             for n in range(len(x)):
               y += x[len(x)-n-1]
             return y
         my_string=reverse('12345')
         print my_string

54321
```

```
In [53]: #what kind of error does this give me?
         #how do I fix it?
         abc='123456'
         for x  range(len(abc)):
           print x
           print abc[x]
         #Reading the error message that you get helps you figure out where you mig
         #have gone wrong in your code. So always read the error message you get.
         #many times it points you to the line and exact word that is problematic.


          File "<ipython-input-53-0bc4d3491bad>", line 2
        for x  range(len(abc)):
                   ^
    SyntaxError: invalid syntax
```

```
In [42]: #when you print the string it does not print it with quotation marks.
         x='timnit'
         print x

timnit
```

```
In [ ]:
```