lec2

August 12, 2016

In [206]:
```
#if Statements: The if statement allows you to only
#conditionally execute some code block,
#conditioned on some expression evaluating to True.

#if BOOLEAN EXPRESSION:
#    CODE BLOCK
#elif BOOLEAN EXPRESSION:
#    CODE BLOCK
#elif BOOLEAN EXPRESSION:
#    CODE BLOCK
#else:
#    CODE BLOCK
```

In [ ]:
```
#In the above code, exactly one of the code blocks is executed,
#corresponding to the first BOOLEAN EXPRESSION which evaluates to True
#(or the final code block corresponding to the else
#in the case that none of the BOOLEAN EXPRESSIONS evaluates to True).
#The elif and else statements are optional.
```

In [3]:
```
#Example
def printSign(n):
    if n < 0:
        print 'Negative'
    elif n > 0:
        print 'Positive'
printSign(10)
#what does this print
```

In [126]:
```
#What about printSign(0) what happens there?
def printSign(n):
    if n>0:
        print 'Positive'
    elif n<0:
        print 'Negative'
    elif n==0:
        print 'Zero'
    print 'my name is timnit'
```

```
In [130]: #What does this print? Do you understand why?
          printSign(0)
```

```
Zero
my name is timnit
```

```
In [ ]: #for Statements: The for statement allows you to iterate over data
        #in Python (for example,iterating over items in a list,
        #or characters in str).
        #for var in v:
        #    CODE BLOCK
        #In the for loop below, x is like var and fruits is like v
```

```
In [204]: fruits = ['orange', 'pineapple', 'banana', 'mango']
          my_fruits=[]
          for x in fruits:
              my_fruits += ['my_'+x]

          fruits[0]='my_' + fruits[0]
          fruits[1]='my_'+ fruits[1]
          print my_fruits
```

```
['my_orange', 'my_pineapple', 'my_banana', 'my_mango']
```

```
In [11]: #For loop example
         fruits = ['orange', 'pineapple', 'banana', 'mango']
         pluralFruits = []
         for x in fruits:
             plural = x + 's'
             print x,plural
             pluralFruits += [plural]
         print pluralFruits
```

```
orange oranges
pineapple pineapples
banana bananas
mango mangos
['oranges', 'pineapples', 'bananas', 'mangos']
```

```
In [ ]: #While statements
        #The while statement allows you to repeatedly execute a code block
        #as long as some bool expression evaluates to True.
        #while BOOLEAN EXPRESSION:
        #   CODE BLOCK
```

```
In [183]: #While loop example
          x = []
```

```
y=0
while True:
    x += [y]
    y += 1
    print x
    print y
    if y < 3:
        break

#print x
#print y
```

```
[0]
1
```

In [ ]:
```
#break and continue: Sometimes you might want to stop iterating
#in a for or while early, or just skip some particular iteration.
#The break and continue statements are useful for this.
#break exits the loop early, and continue moves back to
#the beginning of the loop.
```

In [200]:
```
myList = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
for x in myList:
    if x < 5:
        continue
    print x
```

```
5
6
7
8
9
10
```

In [202]:
```
#Example: Both of these code examples print only the odd numbers between
myList = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
for x in myList:
    if x > 5:
        break
    elif x%2 == 0:
        continue
    else:
        print x
```

```
1
3
5
```

In [201]:
```
# Example with while loop x=0
x=0
while True:
    if x > 5:
        break
    elif x%2 == 0:
        x += 1 #<----What happens if I leave out this line?
        continue
    else:
        print x
    x += 1
```

```
1
3
5
```

In [209]:
```
#Other useful functions:
#It will be helpful for today's lab to know the following functions.
#len(x) returns the length of an iterable data
#type (such as a str or list) as an int. For example,
print len('abc')
print len(['a', 'b', 'c'])
print len(['a', ['b', 'c', 'd']])

print type('abc')
print type(['a', 'b', 'c'])
print type(5)
```

```
3
3
2
<type 'str'>
<type 'list'>
<type 'int'>
```

In [207]:
```
#range(x) returns a list of ints from 0 to x - 1.
#example,
print range(5)
print range(2, 5) #Start at 2
print range(0, 10, 2) #give every 2 values back
```

```
[0, 1, 2, 3, 4]
[2, 3, 4]
[0, 2, 4, 6, 8]
```

In [48]:
```
# Example 1
for x in range(1000):
```

```
            if x == 6:
                break
        print x
        #What does this print?
```

```
6
115.312085 seconds
```

```
In [210]: x_str='bbcdefg'
          print x_str
          print len(x_str)
          x_str[0]

          x=['a','b','c','d','e','f','g']
          print x
          print len(x)
          print x[0]
          x[0]=x_str[0]
```

```
bbcdefg
7
['a', 'b', 'c', 'd', 'e', 'f', 'g']
7
a
```

```
In [107]: x=5
          x=['a','b','c','d','e','f','g']
          x[0]=1
          print x
```

```
[1, 'b', 'c', 'd', 'e', 'f', 'g']
```

```
In [110]: x_str='bbcdefg'
          print x_str[0]
          print x[0]
```

```
b
1
```

```
In [111]: x[0]=x_str[0]
```