# Assign a list to an variable named new\_list

new\_list = [1,2,3]

##Assign different objects

new\_list = ['New string',53,'z',200.324]

##Length of the list

len(new\_list)

##List with same elements repeated

new\_list=[1]\*6

new\_list

##Indexing and Slicing

new\_list = ['one','two','three',4,5,6]

# Grab element at index 0

new\_list[0]

# Grab index 1 and everything past it

new\_list[1:]

# Grab everything UP TO index 3

new\_list[:3]

##Indexing beyond len-1 will give error

new\_list[7]

##Concatenate Lists

new\_list + ['new object']

##Original List stays the same

new\_list

##Can re-assign

new\_list=new\_list+['new element']

##Can multiply the list

new\_list\*3

##Methods

# Create a new list

ls1 = [5,1,2,4,3]

# Append

ls1.append('add me to list')

ls1

##Insert -- insert (at position, 'insert this')

ls1.insert(5,'insert this')

# Show

ls1

# Assign the popped element,

#remember default popped index is -1

popped\_item = ls1.pop()

popped\_item

##Removing element-- what element you want to remove

ls1.remove(3)

ls1

ls1.extend([6,7])

##Reversing

mynew\_list = ['a','m','x','b','c','f']

# Use reverse to reverse order (this is permanent!)

mynew\_list.reverse()

mynew\_list

##Sorting

# Use sort to sort the list (in this case alphabetical order,

# but for numbers it will go ascending)

mynew\_list.sort()

mynew\_list

ls1.sort()

ls1

List\_of\_Integers = [1,5,0,2,6,8,10]

List\_of\_Integers.append(12)

List\_of\_Integers

List\_of\_Integers.sort()

List\_of\_Integers.remove(0)

##Nesting

# Let's make three lists

list1=[1,2,3]

list2=[4,5,6]

list3=[7,8,9]

# Make a list of lists to form a matrix

my\_matrix = [list1,list2,list3]

my\_matrix

# Grab first item in matrix object

my\_matrix[0]

# Grab first item of the first item in the matrix object

my\_matrix[2][0]

##Built in functions

test\_list=[1.6,2.5,3.8,4.1]

sum(test\_list)

##Compare

lst1=[1,2,3,4]

lst2=[2,3]

max(lst1)

max(lst2)

min(lst1)

len(lst2)