```
# 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)
```

```
L1 = ['RPI', 'WPI', 'MIT']
L2 = L1
L3 = ['RPI', 'WPI', 'MIT']
L2.append('RIT')
L2[1] = 'CalTech'
L1
L2
L3
L1 = [1, 2, 3]
L2 = L1.copy()
L1.pop()
L2.append(4)
print(L1)
print(L2)
```