```
#Datatypes

#Numeric datatype --- datatype that includes numeric values
    #Integer----Contains integer values(-infinity to infinity)(-1,3,5.....)
    #Float----Containe decimal point numbers(2.4,3.5......)
    #Complex-----Contains real and imaginary numbers(3+2j)(7-2j)........


#Character datatype
 #string-----character or sequance of characters

 #list------ordered,mutable collection of multiple items that can be heterogeneous
 #Ordered
 #mutable----
 #multiple items
 #heterogeneous----different data types
 #started and ended with 'square brackets'
list1=[1,2,9.8,True,'hello']
print(list1)
 #tuple------ordered, immutable collection of items that can be heterogeneous
 #started and ended with parentheses
 #ordered
 #Immutable
 #Multiple items
 #heterogeneous
tuple1=(1,2,9.8,True,'hello')
print(tuple1)
```

```
[1, 2, 9.8, True, 'hello']
(1, 2, 9.8, True, 'hello')
```

```
#list methods
#append(), extend(), insert(), remove(), pop(), clear(), index(), count(), sort(), reverse(), copy()

#append-adds a single element to the end of the list
li=[1,7,8,9]
li.append(10)
print(li)
```

```
[1, 7, 8, 9, 10]
```

```
#extend()-----adds multiple elements or a new list to the exisiting list
li.extend([1,2,3,4,5])
print(li)
```

```
[1, 7, 8, 9, 10, 1, 2, 3, 4, 5]
```

```
#insert---adds the element to a specific position
li.insert(2,100)
print(li)
```

```
[1, 7, 100, 8, 9, 10, 1, 2, 3, 4, 5]
```

```
#remove----removes a specific element
          # throws error if the element is not present
#When the element is present---it removes the element
li.remove(100)
print(li)
```

```
[1, 7, 8, 9, 10, 1, 2, 3, 4, 5]
```

```
#When the element is not present in the list---it throws error
#li.remove(100)
#print(li)


#pop----removes the element on the position given
li.pop(2)
print(li)
```

```
[1, 7, 9, 10, 1, 2, 3, 4, 5]
```

```
li.pop()
```

```
5
```

```
print(li)
```

```
[1, 7, 9, 10, 1, 2, 3, 4]
```

```
#li.remove()
#Will give error
```

```
#clear----will remove all the elements
li.clear()
print(li)
#Will remove the elements the structure will be there as it it
#[]
```

```
[]
```

```
li=[1,2,3,4,5,1,5,4,3,2,1,4,5,4,2,2,1,3]
li.index(2)
#will give the index of the first occurance of the element
```

```
1
```

```
#Count--gives the count of the specific element
li.count(2)
```

```
4
```

Start coding or generate with AI.