```
# list basics

data1=[1,2,3,4,5]
data2=['a','b','c']
data3=[12.6,11.6]
data4=['siddhu','eswar']
data5=[]
data6=['siddhu',10,56.4,'a']

print(data1)
print(data2)
print(data3)
print(data4)
print(data5)
print(data6)
```

```
[1, 2, 3, 4, 5]
['a', 'b', 'c']
[12.6, 11.6]
['siddhu', 'eswar']
[]
['siddhu', 10, 56.4, 'a']
```

```
# accessing value from it

data1=[1,2,3,4,5]
data2=['a','b','c']

print(data1[0])
print(data1[0:2])
print(data2[-3:-1])
print(data1[0:])
print(data2[:2])
```

```
1
[1, 2]
['a', 'b']
[1, 2, 3, 4, 5]
['a', 'b']
```

```
# adding list

list1=[10,20,30]
list2=[40,50]
list3=list1+list2
print(list3)
```

```
[10, 20, 30, 40, 50]
```

```
# replicating lists

list1=[10,20,30]
print(list1*2) # replicating means repeating.it can be performed by using '*' operator by a specific number of time.
```

```
[10, 20, 30, 10, 20, 30]
```

```
# list slicing

list1=[1,2,3,4,5]
print(list1[0:2])
print(list1[4])
list1[4]=9 # updating an element
print(list1)
```

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

```
# updating an elements in a list

data1=[1,2,3,4,5]
print("values of list are:")
print(data1)
data1[2]="multiple of 5" # update an element in perticular position
print("values of list are:")
print(data1)
```

```
values of list are:
[1, 2, 3, 4, 5]
values of list are:
[1, 2, 'multiple of 5', 4, 5]
```

```
# appending a list

list1=[17,"vivek","data-analytics"]
print("elements list are:")
print(list1)
list1.append("course") # append a list and remember append only add the content/element at last position.
print("list after appending:")
print(list1)
```

```
elements list are:
[17, 'vivek', 'data-analytics']
list after appending:
[17, 'vivek', 'data-analytics', 'course']
```

```
# deleting elements in list

list1=[17,"vivek","data-analytics"]
print("elements of list are:")
print(list1)
del(list1[0])
print("after deletion:")
print(list1)
```

```
elements of list are:
[17, 'vivek', 'data-analytics']
after deletion:
['vivek', 'data-analytics']
```

```
# membership operator

x=[1,2,3,4,5]
print(5 in x)
print(3 not in x)
print(90 in x)
print(90 not in x)
```

```
True
False
False
True
```

```
# multiplying with 2

x=[1,2,3,4,5]
y=[]
for i in x:
  y.append(i*2)
print(y)
```

```
[2, 4, 6, 8, 10]
```

```
# list comprehensions

s=range(1,20,3)
for i in s:
  print(i)
  m=[x for x in s if(x%2==0)]
print(m)
```

```
1
4
7
10
13
16
19
[4, 10, 16]
```

## Functions of a list

```
a = [1,2,3,4,5]
print(min(a))
```

```
1
```

```
a = [1,2,3,4,5]
print(max(a))
```

```
5
```

```
a = [1,2,3,4,5]
print(len(a))
```

```
5
```

## Methods of a list

```
a = [1,2,3,4,5]
print(a.index(1))
```

```
0
```

```
a = [1,2,3,4,5,4,4]
print(a.count(4))
```

```
3
```

```
a = [1,2,3,4,5]
a.pop(2) #give the index number only not the list of element
```

```
3
```

```
a = [1,2,3,4,5]
a.insert(5,6)
print(a)
```

```
[1, 2, 3, 4, 5, 6]
```

```
a = [1,2,3,4,5]
b = [6,7,8,9,10]
a.extend(b) # extend method is used for the continuity for the list
print(a)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
a = [1,2,3,4,5]
a.remove(1) # is same like the delete element function
print(a)
```

```
[2, 3, 4, 5]
```

```
a = [1,2,3,4,5]
a.reverse()
print(a)
```

```
[5, 4, 3, 2, 1]
```

```
a = [10,2,30,1,3]
a.sort()
print(a)
```

```
[1, 2, 3, 10, 30]
```

Start coding or generate with AI.