Lists Part -2

In this lecture

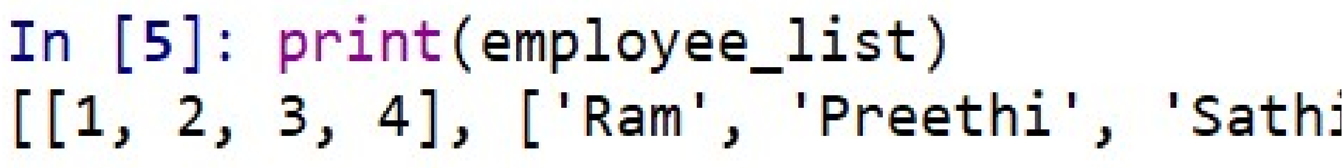
- Modify lists
  - Add elements
  - Remove elements

Modifying components of a list

- Elements inside a list can be modified using two methods
- Assigning the new element directly to the index position that has to be updated
- Using in built functions where the element that is to be updated with is given as an input to the function along with the index position

Modifying components of a list us

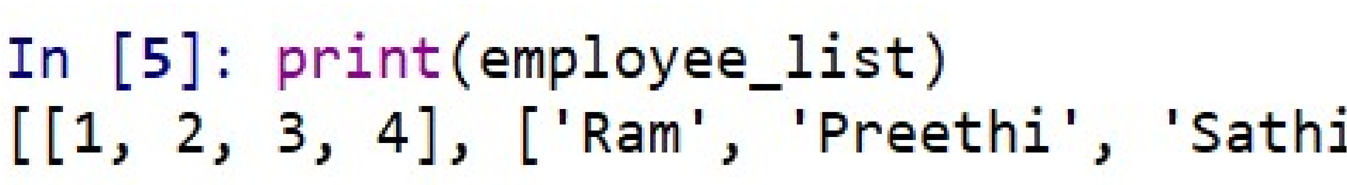
- Assign the values to be changed to c
  the list
- Eg- Change the value in top level comp
- Existing list



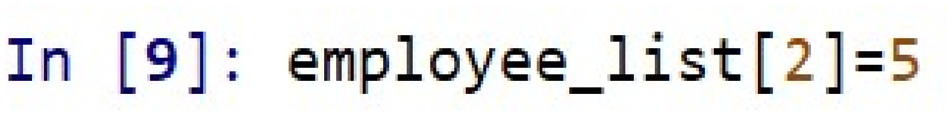
```
In [5]: print(employee_list)
[[1, 2, 3, 4], ['Ram', 'Preethi', 'Sathi
```

Modifying components of a list us

- Here the value of 4 should be updated

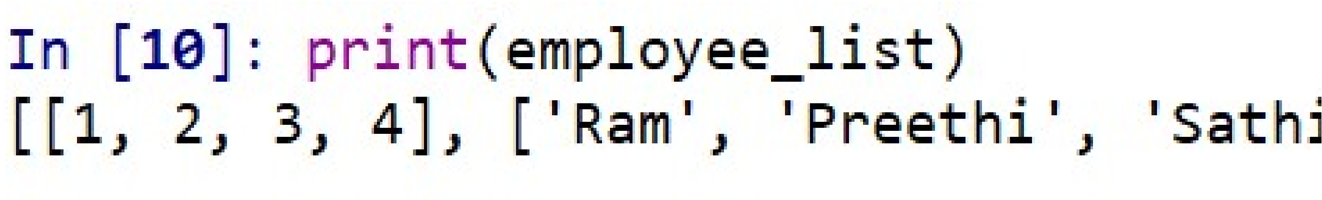


```
In [5]: print(employee_list)
[[1, 2, 3, 4], ['Ram', 'Preethi', 'Sathi
```



```
In [9]: employee_list[2]=5
```

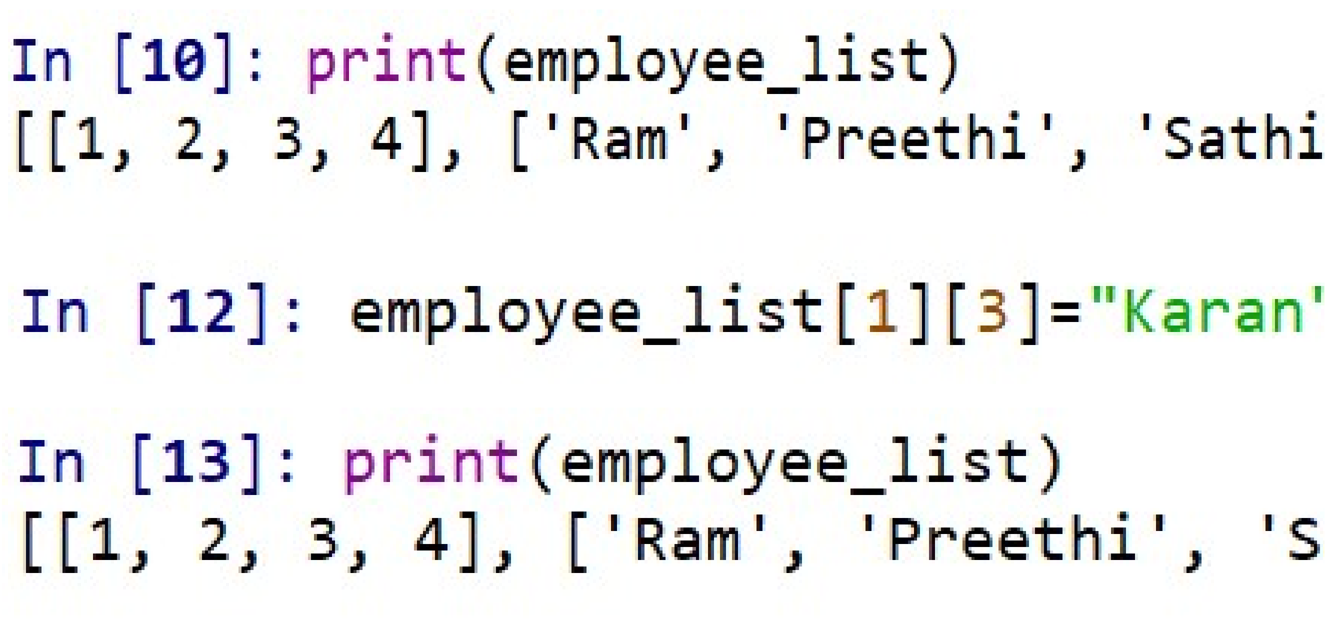
- Print the updated list



```
In [10]: print(employee_list)
[[1, 2, 3, 4], ['Ram', 'Preethi', 'Sathi
```

Modifying components of a list us

- Eg- Change value in sub level compone



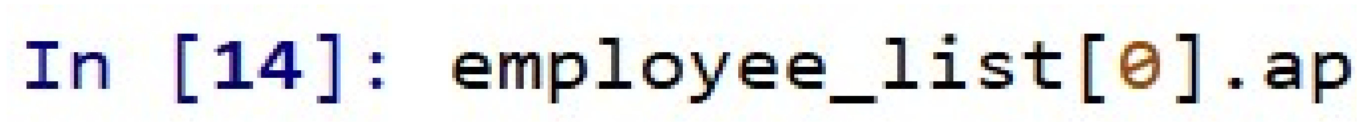
```
In [10]: print(employee_list)
[[1, 2, 3, 4], ['Ram', 'Preethi', 'Sathi

In [12]: employee_list[1][3]="Karan'

In [13]: print(employee_list)
[[1, 2, 3, 4], ['Ram', 'Preethi', 'S
```

Modifying components using app

- append()- adds an object at the end o
- Syntax: list_name[index].append
- In the above syntax if the 'index' is object gets added as a new level in the
- There are two ways to add an object to
  - Adding an element to a list
  - Adding a list to a list

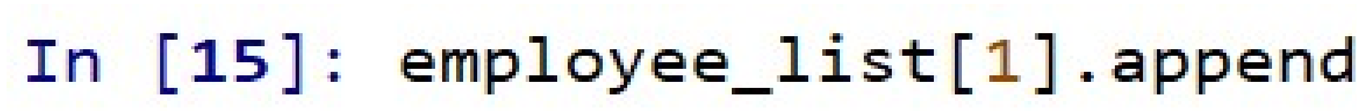
Modifying components using app

- Adding an element to a list
- Adding number ***'5'*** to the level ***id*** in ***en***



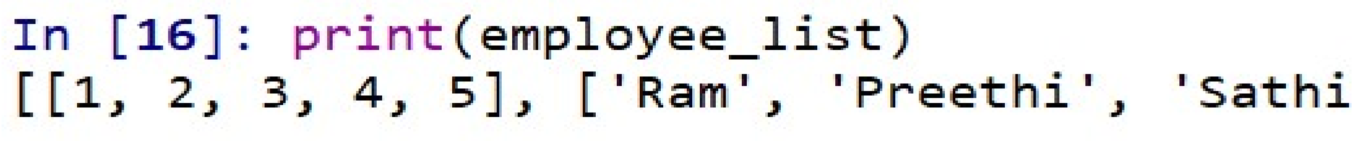
```
In [14]: employee_list[0].ap
```

- Adding name ***'nirmal'*** to the level ***emp*** ***employee_list***



```
In [15]: employee_list[1].append
```

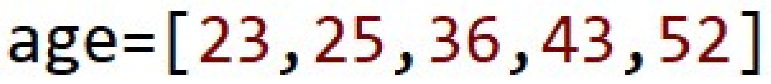
- Print the updated list



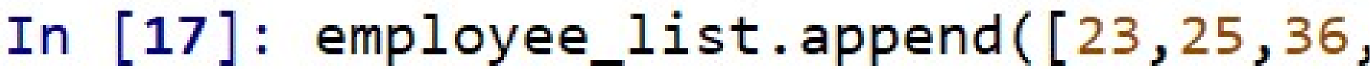
```
In [16]: print(employee_list)
[[1, 2, 3, 4, 5], ['Ram', 'Preethi', 'Sathi
```

Modifying components using app

- Adding a list to a list (also termed as co
- Adding a new list ***age*** to the existing ***e***

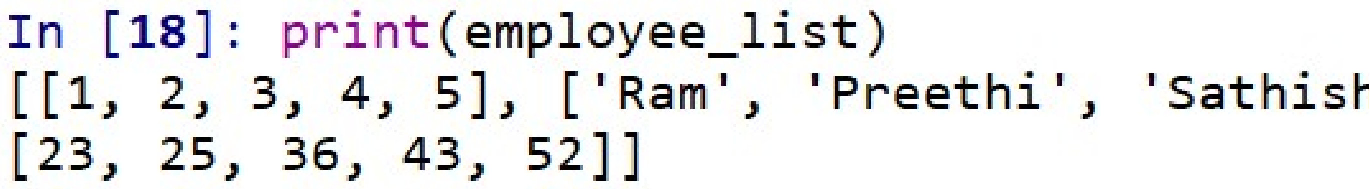


```
age=[23,25,36,43,52]
```



```
In [17]: employee_list.append([23,25,36,
```

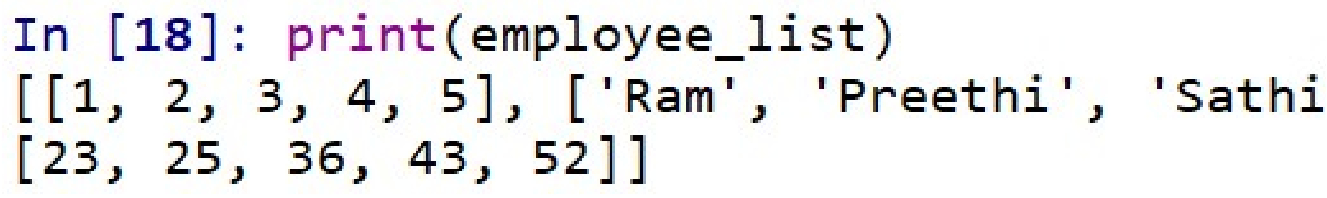
- The new list gets added as a new level
- Print the updated list



```
In [18]: print(employee_list)
[[1, 2, 3, 4, 5], ['Ram', 'Preethi', 'Sathish
[23, 25, 36, 43, 52]]
```

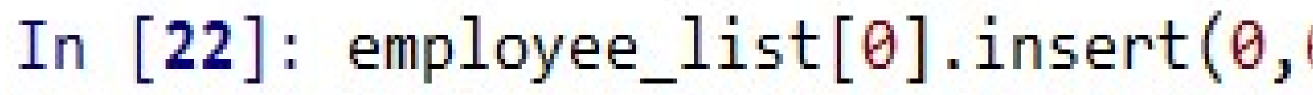
Modifying components using ins

- `insert()`- adds an object at the given
- Syntax: `list_name[index].insert`
- Existing list



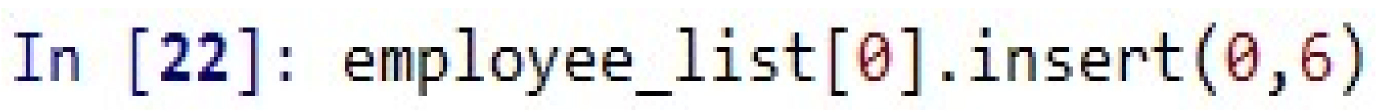
```
In [18]: print(employee_list)
[[1, 2, 3, 4, 5], ['Ram', 'Preethi', 'Sathi
[23, 25, 36, 43, 52]]
```

- Adding number ***'6'*** at the ***1st positio*** ***employee_list***



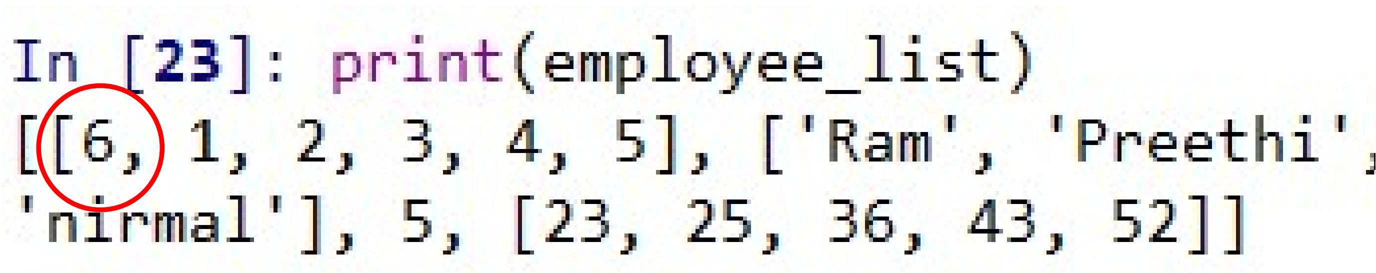
```
In [22]: employee_list[0].insert(0,
```

Modifying components using ins



```
In [22]: employee_list[0].insert(0,6)
```

- Print the updated list



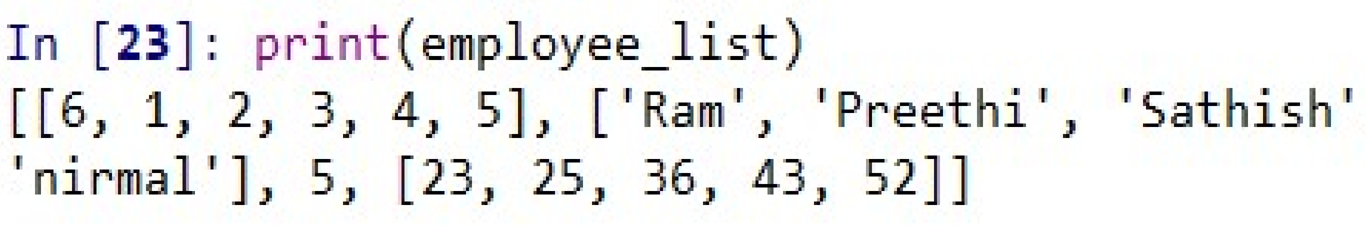
```
In [23]: print(employee_list)
[[6, 1, 2, 3, 4, 5], ['Ram', 'Preethi',
'nirmal'], 5, [23, 25, 36, 43, 52]]
```

Modifying components using del

- `del`- removes the object at the specifie
- Syntax: `del list_name[index1][i`
- In the above syntax,
  - `index1`- index number of the top le dropped
  - `index2` corresponds to the sub level of c

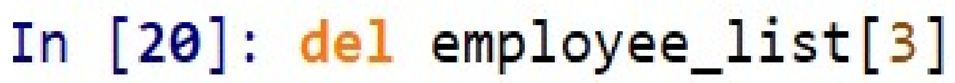
Modifying components using del

- Existing list



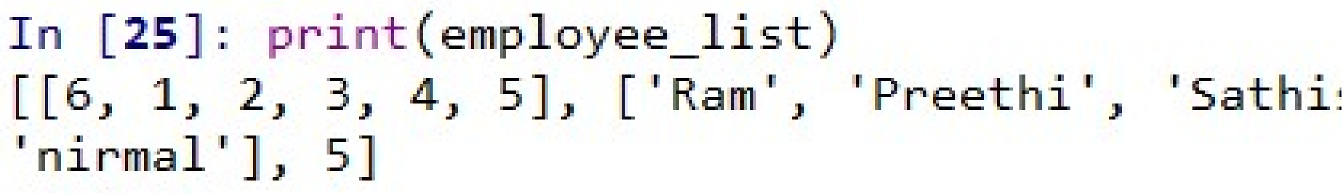
```
In [23]: print(employee_list)
[[6, 1, 2, 3, 4, 5], ['Ram', 'Preethi', 'Sathish'
'nirmal'], 5, [23, 25, 36, 43, 52]]
```

- Drop the last level i.e. ***age*** from ***emplo***



```
In [20]: del employee_list[3]
```

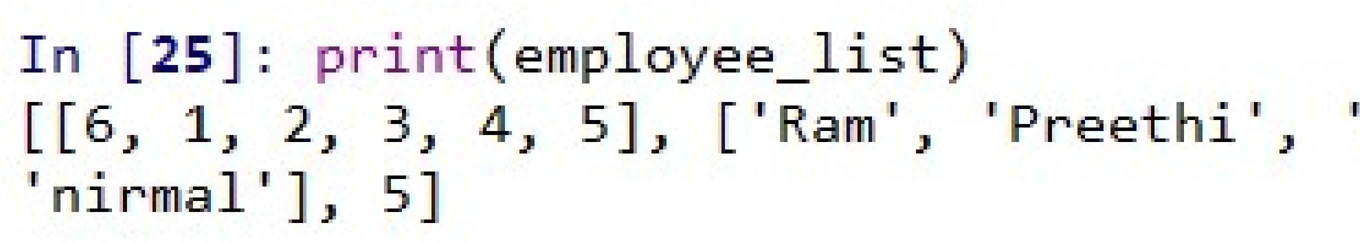
- Print the updated list



```
In [25]: print(employee_list)
[[6, 1, 2, 3, 4, 5], ['Ram', 'Preethi', 'Sathi
'nirmal'], 5]
```

Modifying components using rer

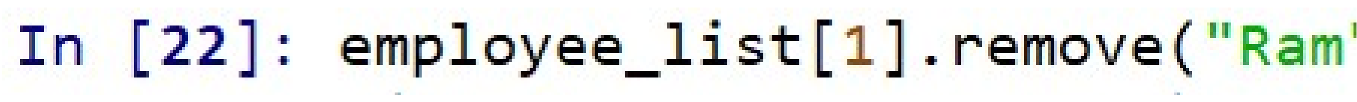
- remove()- removes the first matching
- Syntax: list_name[index].remove
- Existing list



```
In [25]: print(employee_list)
[[6, 1, 2, 3, 4, 5], ['Ram', 'Preethi', '
'nirmal'], 5]
```

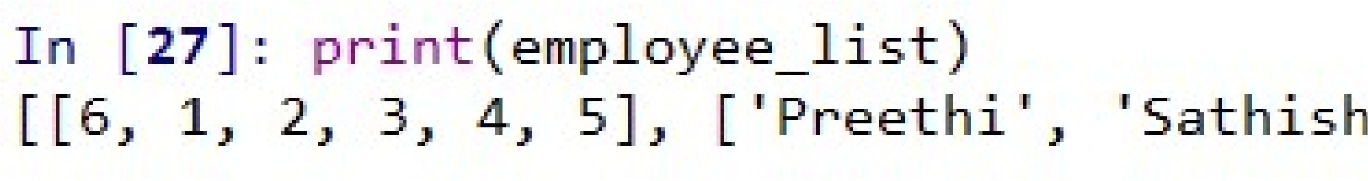
Modifying components using rer

- Remove ***‘Ram’*** from the level ***er***
  employee_list



```
In [22]: employee_list[1].remove("Ram'
```

- Print updated list



```
In [27]: print(employee_list)
[[6, 1, 2, 3, 4, 5], ['Preethi', 'Sathish
```

- Here ***‘Ram’*** occurs only once

Modifying components using re

- Consider another list

```
salary=['High','Low','Medium','Low']
```

- Removing the first occurrence of ***'Low***

```
In [22]: salary.remove('Low')
```

- Print the updated list

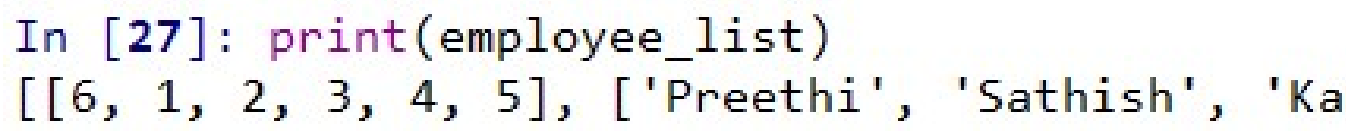
```
In [23]: print(salary)
['High', 'Medium', 'Low']
```

Modifying components using po

- pop()- displays the object that is being
  list at the specified index number
- Syntax: list_name[index1].pop(i
- In the above syntax,
  - index1- index number of the top lev
    dropped
  - index2 corresponds to the sub leve
    dropped

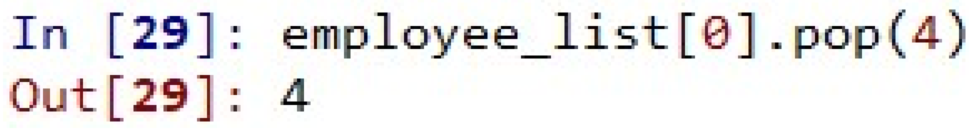
Modifying components using po

- Existing list



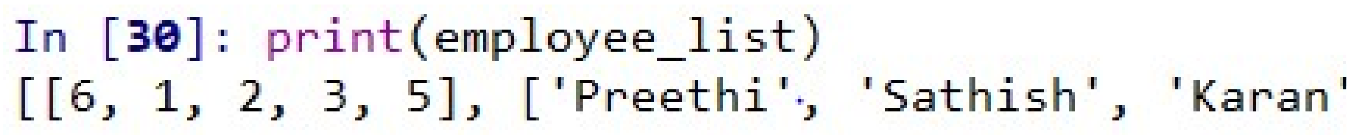
```
In [27]: print(employee_list)
[[6, 1, 2, 3, 4, 5], ['Preethi', 'Sathish', 'Ka
```

- Removing number ***'4'*** from the ***5th po***
  employee_list



```
In [29]: employee_list[0].pop(4)
Out[29]: 4
```

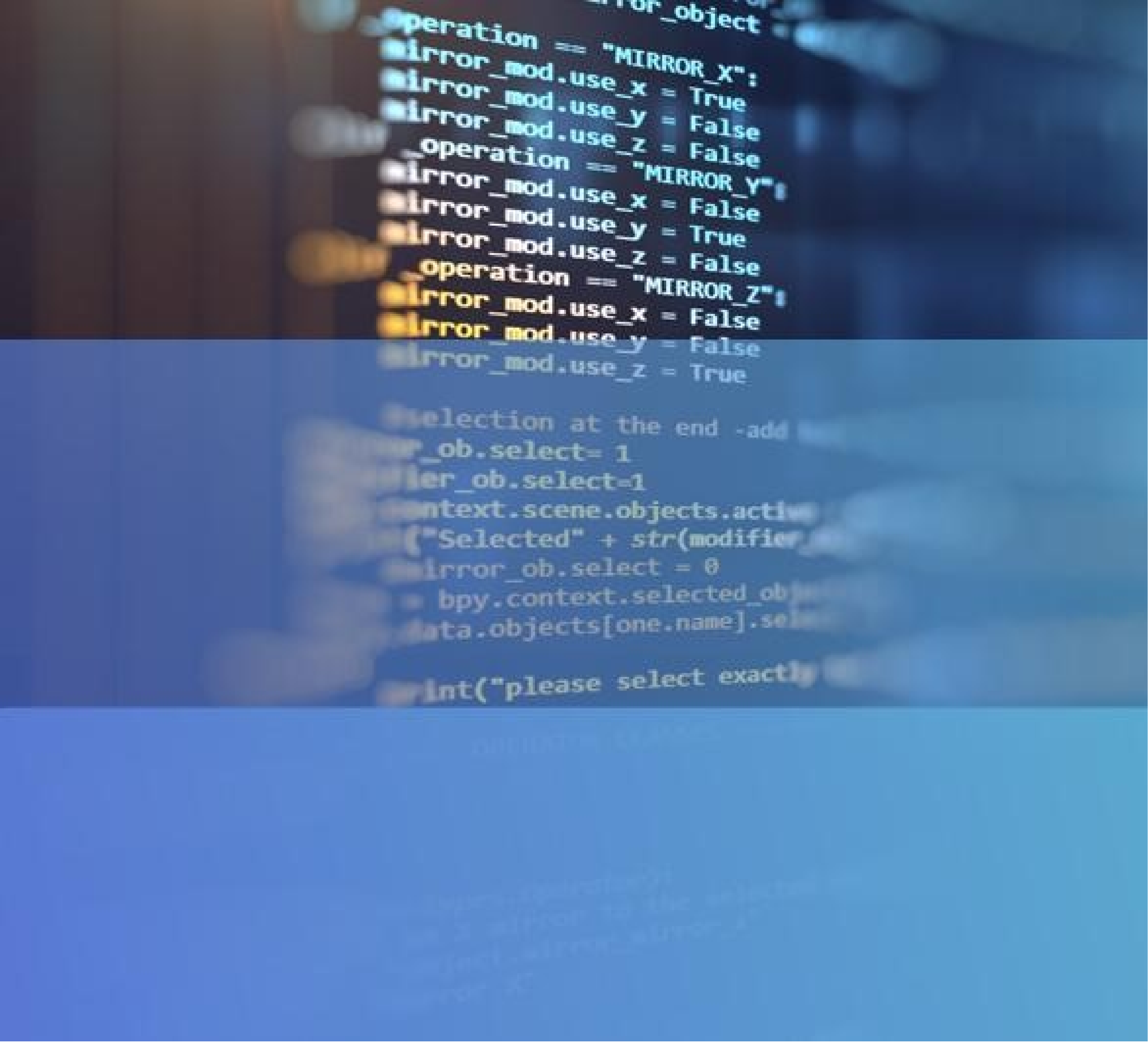
- Print the updated list



```
In [30]: print(employee_list)
[[6, 1, 2, 3, 5], ['Preethi', 'Sathish', 'Karan'
```

Summary

- Manipulate lists directly using the index
- Manipulate lists using functions:
  - append - adds an element at the end of t
  - insert - adds an element at the specified
  - del - removes the element at the spe
  - remove - removes the first matching elem
  - pop - displays and removes the eleme



```
rror_object
operation == "MIRROR_X":
mirror_mod.use_x = True
mirror_mod.use_y = False
mirror_mod.use_z = False
_operation == "MIRROR_Y":
mirror_mod.use_x = False
mirror_mod.use_y = True
mirror_mod.use_z = False
_operation == "MIRROR_Z":
mirror_mod.use_x = False
mirror_mod.use_y = False
mirror_mod.use_z = True

selection at the end -add
r_ob.select= 1
ier_ob.select=1
ntext.scene.objects.acti
("Selected" + str(modifie
mirror_ob.select = 0
= bpy.context.selected_ob
data.objects[one.name].se

int("please select exact
```