



Lists Part -1

In this lecture

- Lists
 - Creating a list
 - Indexing
 - Access components

Lists

Lists

- Generic data structure in Python consisting of an ordered collection of objects
- Objects in a list are also known as elements or components

- Elements of a list need not be of same data type
 - Lists can consist of a numeric array, a logical value, a matrix, a complex vector, a character array, a function etc.
- Enclosed between two square brackets - []

Creating a list

- Create the lists employee id and name
- Create a variable that contains number of employees

```
id = [1,2,3,4]
```

```
employee_name = ["Ram", "Preethi", "Sathish", "John"]
```

```
num_emp = 4
```

Creating a list

- Create an employee list using employee id, employee name and number of employees

```
employee_list = [id,employee_name,num_emp]
```

- To view a list

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

Indexing

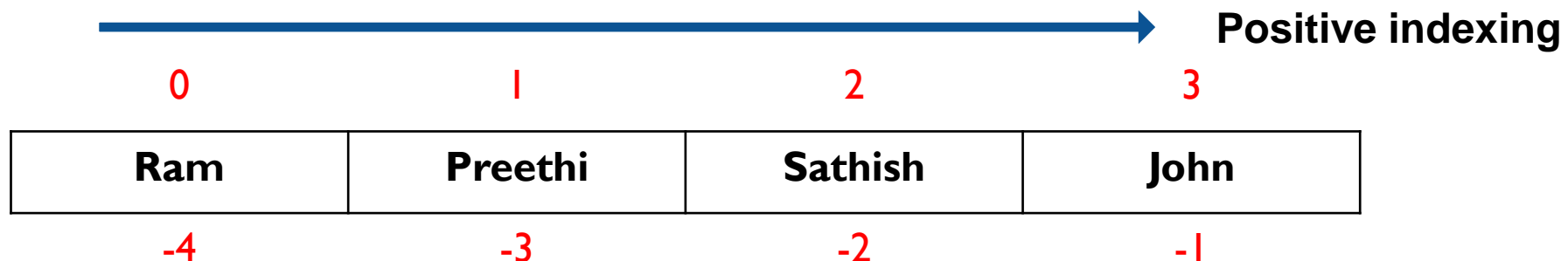
- There are two types of indexing - positive and negative
- Positive indexing
 - Starts from the left most element
 - 0 is the first index
- Consider the following list

```
In [2]: employee_name = ["Ram", "Preethi", "Sathish", "John"]
```


Indexing

- There are two types of indexing- positive and negative
- Positive indexing
 - Starts from the left most element
 - 0 is the first index
- Consider the following list

```
In [2]: employee_name = ["Ram", "Preethi", "Sathish", "John"]
```



Indexing

- Negative indexing
 - Starts from the right most element
 - -1 is the first index
- Consider the same list

```
In [2]: employee_name = ["Ram", "Preethi", "Sathish", "John"]
```

| | | | |
|-----|---------|---------|------|
| 0 | 1 | 2 | 3 |
| Ram | Preethi | Sathish | John |
| -4 | -3 | -2 | -1 |



Negative indexing

Accessing components of a list

- To access top level components, use single slicing operator “[]”
- For sub-level / inner level components use “[]” followed by another “[]”

Accessing components of a list

- To extract **id** from the **employee_list**

```
In [6]: print(employee_list[0])  
[1, 2, 3, 4]
```

- To extract 'Preethi' from the level **employee_name** that belongs to **employee_list**

```
In [7]: print(employee_list[1][1])  
Preethi
```

Accessing components of a list

- To extract the second id from the level **id** that belongs to **employee_list**

```
In [8]: print(employee_list[0][1])  
2
```

Summary

- Create lists
- Indexing
- Accessing top and sub level components from a list

```
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  
mirror_ob.select= 1  
modifier_ob.select=1  
context.scene.objects.active  
= ("Selected" + str(modifier_ob.name))  
mirror_ob.select = 0  
= bpy.context.selected_objects  
data.objects[one.name].select  
print("please select exactly one mirror")
```

WILLIAM C. CARR

```
def select_mirror(modifier):  
    #select mirror to the selected  
    #object -mirror_mirror  
    mirror_ob = bpy.context.selected_objects[0]  
    mirror_ob.select = 1
```

THANK YOU