# Lambda

# Session Objectives

- Concept
- Why we need Lambda & Benefits
- With or Without Lambda
- Map, Reduce & Filter
- Summary

# Session Objectives

- Concept
  - A lambda function is a small anonymous function that is defined without a name.
  - A lambda function can take any number of arguments, but can only have one expression.
  - Syntax
    - lambda *arguments* : *expression*
  - Example:
    - double= lambda x : x*2

      print(double(10))

      output:20

# Why we need Lambda & Benefits

- A lambda expression can also enable us to write parallel processing because every processor is a multi-core processor nowadays.
- Lambda functions **reduce the number of lines of code when compared to normal python function defined using def keyword**
- They are generally used when a function is needed temporarily for a short period of time, often to be used inside another function such as filter , map and reduce .

# Without Lambda

```
def find_max(a,b)
      if  a>b:
            return a
      else
            return b

z=find_max(10,30)
print("max value",z)
```

Output:max value 30

# With Lambda

```
max= lambda a,b:a if a>b else b

print("max value:",max(10,30))
```

Output:max value 30

# Filter,map,reduce

- filter ():
  - The filter() function in Python takes in a function and a list as arguments.
  - The function is called with all the items in the list and a new list is returned which contains items for which the function evaluates to True.

  - # Program to filter out only the even items from a list

```
my_list = [1, 5, 4, 6, 8, 11, 3, 12]

new_list = list(filter(lambda x: (x%2 == 0) , my_list))

print(new_list)
```

Output

```
[4, 6, 8, 12]
```

- map()
  - The map() function in Python takes in a function and a list.
  - The function is called with all the items in the list and a new list is returned which contains items returned by that function for each item.
  - Example:

    Program to double each item in a list using map()

```
my_list = [1, 5, 4, 6, 8, 11, 3, 12]
new_list = list(map(lambda x: x * 2 , my_list))
print(new_list)
```

Output

```
[2, 10, 8, 12, 16, 22, 6, 24]
```

- reduce()
  - The reduce(fun,seq) function is used to apply a particular function passed in its argument to all of the list elements mentioned in the sequence passed along.This function is defined in “functools” module.
  - Working :
    - At first step, **first two elements of sequence are picked and the result is obtained**.
    - Next step is to **apply the same function to the previously attained result and the number just succeeding the second element and the result is again stored**.
    - This process continues till no more elements are left in the container.
    - The final returned result is returned and printed on console.

- Example:

```
import functools
my_list = [1, 3, 5, 6, 2, ]
# using reduce to compute sum of list
print("The sum of the list elements is : ", end="")
print(functools.reduce(lambda a, b: a+b,my_ list))
```

Output:

The sum of the list elements is : 17

# Summary

- Concept
- Why we need Lambda & Benefits
- With or Without Lambda
- Filter,Map, Reduce

Demo

???

The Important thing is not to stop

**Questioning**

RECORDED

CORPORATE

INTERACTIVE