Introduction to Lambda

- At the end of this class, students will be able to understand that-

  Instead of the def syntax for function declaration, we can use a lambda expression to write Python functions.

  The lambda syntax closely follows the def syntax, but it's not a 1-to-1 mapping.

The Lambda Expression: an anonymous function.

- lambda *operator or lambda function* is used for creating small, one-time and anonymous function objects in Python.
- **Basic syntax**

  lambda arguments : expression
- *lambda operator can have any number of arguments, but it can have only one expression.*
- *It cannot contain any statements and it returns a function object which can be assigned to any variable.*

With arguments

```
g = lambda x: x**3
print(g(7))
```

Output:

343

Without arguments and name

```
a, b=1,2
c=lambda :a+b
print(c())
```

Output:

3

Example:

```
# Python code to illustrate cube of a number
# showing difference between def() and lambda().
def cube (y):
   return y

g = lambda x: x**3
print(g(7))

print(cube(3))
```

Output:

```
343
3
```

Use of lambda in map

```
# Python code to illustrate  map() with lambda()
# to get double of a list.
li = [5, 7, 22, 97, 54, 62, 77, 23, 73, 61]
final_list = list(map(lambda x: x*2 , li))
print(final_list)
```

Output:

[10, 14, 44, 194, 108, 124, 154, 46, 146, 122]

Double all numbers using map and lambda

```
numbers = (1, 2, 3, 4)
result = map(lambda x: x + x, numbers)
print(list(result))
```

Output:

[2, 4, 6, 8]

Add two lists using map and lambda

```
numbers1 = [1, 2, 3]
numbers2 = [4, 5, 6]

result = map(lambda x, y: x + y, numbers1, numbers2)
print(list(result))
```

Output:

[5, 7, 9]

Use of lambda in filter

```
# a list contains both even and odd numbers.
seq = [0, 1, 2, 3, 5, 8, 13]

# result contains odd numbers of the list
result = filter(lambda x: x % 2, seq)
print(list(result))

# result contains even numbers of the list
result = filter(lambda x: x % 2 == 0, seq)
print(list(result))
```

Output:

[1, 3, 5, 13]
[0, 2, 8]

Summary

- *lambda operator or lambda function* is used for creating small, one-time and anonymous function objects in Python.