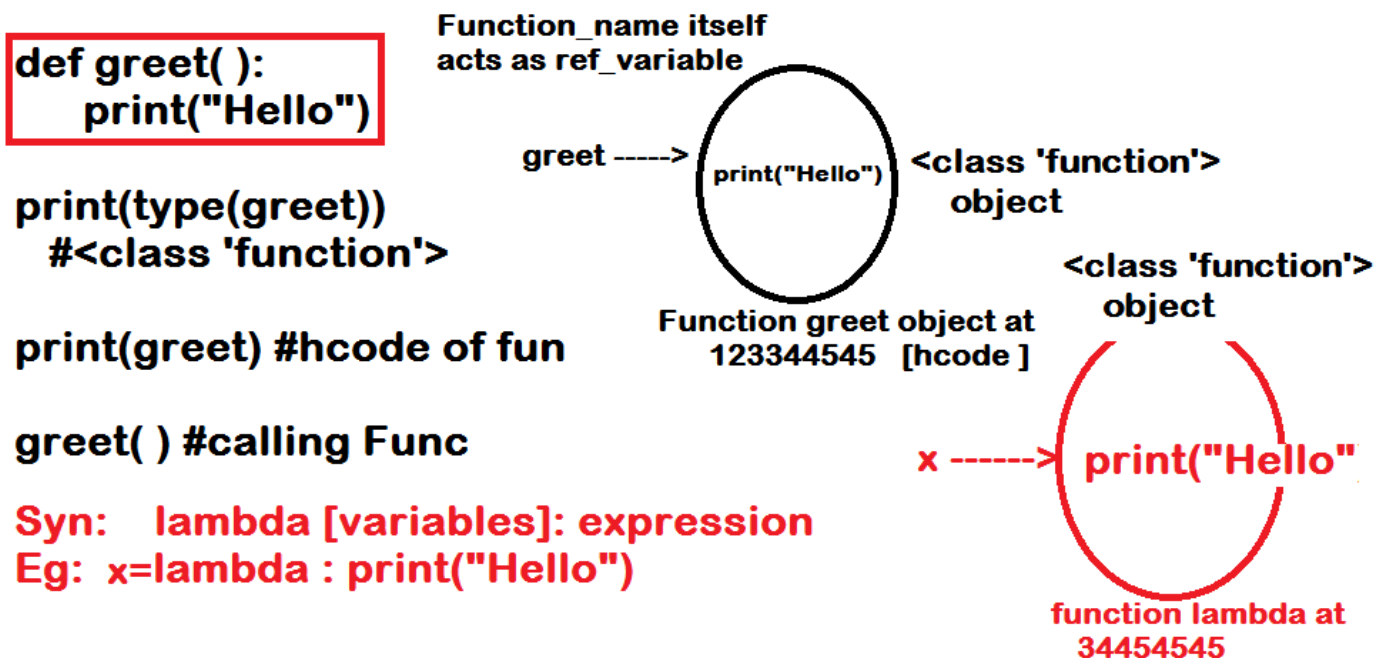# Lambda Function:

> **Lambda is keyword**
> **Using lambda keywords we can define a function without name in simple words lambda function are nothing but nameless functions**
> **Lambda function useful to avoid defining simplified functions**
> **Simplified function is a function which having very less logic**
> **Lambda function implicitly returns the value**

```
def greet( ):
    print("Hello")

print(type(greet))
  #<class 'function'>

print(greet) #hcode of fun

greet( ) #calling Func
```

Function_name itself acts as ref_variable

greet -----> print("Hello")   **<class 'function'> object**

Function greet object at 123344545   [hcode]

<class 'function'> object

x ------> print("Hello"

function lambda at 34454545

Syn: lambda [variables]: expression
Eg: x=lambda : print("Hello")

Example 1:
x=lambda : print("Hello")
print("Type is : ",type(x)) #<class 'function'>
print("Hcode is : ",x) #<function lambda at 23343434>
x() #calling lambda function

Example 2:
add=lambda x,y: x+y

```
r=add(10,20)
print("Result is : ",r)
```

Example 3:
biggest=lambda x,y : x if x>y else y

```
#calling
big=biggest(110,20)
print("Result is : ",big)
```

Exampe 4:
myUpper=lambda name: name.upper()

```
#calling
r=myUpper("Hello my dear")
print("Result is UCASE : ",r)
```