

Anonymous (Lambda) Functions



Lambda Functions are not declared in the standard manner by using the `def` keyword.

We can use the `lambda` keyword to create small anonymous fn.

```
lambda [arg1 [,arg2,.....argn]]:expression
```

```
# Function definition is here
```

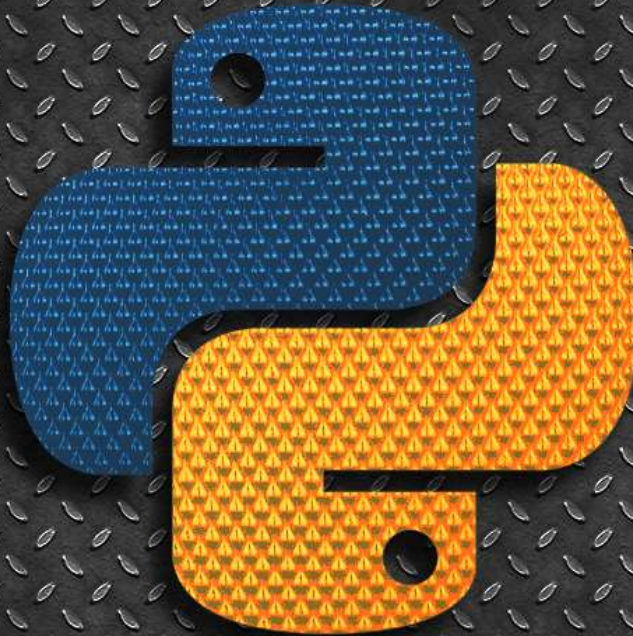
```
sum = lambda arg1, arg2: arg1 + arg2
```

```
# We can call sum as a function
```

```
print ("Value of total : ", sum( 10, 20 ))
```

```
print ("Value of total : ", sum( 20, 20 ))
```


Python Fundamentals



Modules

Modules



Python comes with a large number of built-in functions.

These functions are saved in files known as **modules**.

To use Python modules, we have to import them into our programs

We do that by using the **import** keyword.

```
#Eg: To use the randrange() function in the random module  
import random  
random.randrange(1, 10)
```

Modules



We can also write like:

```
#using alias name for the module
import random as r
r.randrange(1, 10)
```

```
#importing only specific function from the module
#from moduleName import name1, name2, name3
```

```
from random import randrange, randint
randrange(1, 10)
```

Creating Own Custom Module



We can also write like:

#Step 1: Define the module and save it as python file
prime.py (in the same folder)

```
def checkIfPrime (numberToCheck):  
    for x in range(2, numberToCheck):  
        if (numberToCheck%x == 0):  
            return False  
    return True
```


Creating Own Custom Module



Example:

#Step 1: Define the module and save it as a python file
prime.py (save in the same folder)

```
def checkIfPrime (numberToCheck):  
    for x in range(2, numberToCheck):  
        if (numberToCheck%x == 0):  
            return False  
    return True
```

Creating Own Custom Module



Example:

#Step 2: Import and use the module

```
import prime  
answer = prime.checkIfPrime(13)  
print (answer)
```