Module Creation

```
# module : calc.py
def sum(a,b):
  return a+b
def sub(a,b):
  return a-b
def mul(a,b):
  return a*b
def div(a,b):
  return a/b
```

Import Statement

```
#Import Statement
import calc   #importing entire Module
a=int(input("Enter a :"))
b=int(input("Enter b :"))
print("Sum is :",calc.sum(a,b))
print("Sub is :",calc.sub(a,b))
print("Mul is :",calc.mul(a,b))
print("Div is :",calc.div(a,b))
```

```
Enter a :89
Enter b :65
Sum is : 154
Sub is : 24
Mul is : 5785
Div is : 1.3692307692307693
```

from - import statement

```
#from - import statement
from calc import sub,mul  #importing specific functionality from Module
a=int(input("Enter a :"))
b=int(input("Enter b :"))
print("Sub is :",sub(a,b))
print("Mul is :",mul(a,b))
```

```
Enter a :25
Enter b :6
Sub is : 19
Mul is : 150
```

Renaming Module

```
# Renaming Module
import calc as c
a=int(input("Enter a :"))
```

```
b=int(input("Enter b :"))
print("Sum is :",c.sum(a,b))
print("Sub is :",c.sub(a,b))
```

```
Enter a :12
Enter b :5
Sum is : 17
Sub is : 7
```

Scope of Variables

```
#Scope of Variables
name = "Madhu"  # Global Variable
def disp_name(name):
    print("Hi",name) #prints the name that is local to this function only.
name = input("Enter the name :")
disp_name(name)
```

```
Enter the name :Kiran
Hi Kiran
```

Namespaces

```
#Namespaces
print("Namespace Example")    #built-in namespace
a=10    #global namespace
def func1():
  b=20  #local namespace
  print(a+b)
func1()
```

```
Namespace Example
30
```

```
#Python supports “global” keyword to update global namespaces in local.
count = 5
def func1():
    global count  #To update global namespace
    count = count + 1
    print(count)
func1()
```

```
6
```

```
#non-local namespace
a=10 #global namespace
def func1():
  b=20  #non-local namespace
  def func2():
    nonlocal b
    c=30  #local namesapce
    global a
```

```
    a=a+c
    b=b+c
  func2()
  print(a,b)
func1()
```

```
 40 50
```

Module Built-in Functions

```
#dir()
import calc
ls=dir(calc)
print(ls)
```

```
['__builtins__', '__cached__', '__doc__', '__file__', '__loader__', '__name__', '__package__', '__spec__
```

```
#globals() & #locals()
xy="Madhu"
def sum(a,b):
  c=0
  c=a+b
  print(c)
  print(globals())
  print(locals())
sum(2,3)
```

```
5
{'__name__': '__main__', '__doc__': 'Automatically created module for IPython interactive environment',
{'c': 5, 'b': 3, 'a': 2}
```

```
#reload()
import calcmodule
import importlib
importlib.reload(calcmodule)
```

```
Welcome to Calc Module
<module 'calcmodule' from '/content/calcmodule.py'>
```

Packages

```
import mypkg.Mod1 as m1
import mypkg.Mod2 as m2
import mypkg.Mod3 as m3
m1.models()
m2.models()
m3.models()
```

```
These are the available models for SAMSUNG
['Galaxy J6', 'Galaxy M20', 'Galaxy A10']
These are the available models for REALME
['RealMe 1', 'RealMe 2', 'RealMe 3', 'RealMe 3']
These are the available models for IPHONE
['5', '5s', '6', '6s', 'X']
```

```
import mypkg as m
import mypkg.Mod1 as m1
import mypkg.Mod2 as m2
import mypkg.Mod3 as m3
print(m.ls)
m1.models()
m2.models()
m3.models()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-17-a4aa8b8c12d2> in <module>()
      3 import mypkg.Mod2 as m2
      4 import mypkg.Mod3 as m3
----> 5 print(m.ls)
      6 m1.models()
      7 m2.models()

AttributeError: module 'mypkg' has no attribute 'ls'
```

SEARCH STACK OVERFLOW