```
#Functions:

# re use the same code
# 20 having birtah day
# omkar hapy bday
# suresh hapy bday
```

In [ ]: In [1]:

```
num1=eval(input("enter
the number1:"))
num2=eval(input("enter
the number2:"))
print(num1+num2)
```

```
enter the number1:100
enter the number2:200
300
```

In [ ]: In [3]:

```
def <function_name>():
 # write your code
```

```
def summ():
 num1=eval(input("enter
the number1:"))
num2=eval(input("enter
the number2:"))
print(num1+num2)

summ()
```

In [12]:

```
enter the number1:20
enter the number2:30
50
```

variables

Data types

type casting

print

Basic codes

eval

input

if-else conditions

try-except

In [9]:

```
def addition1():
 num1=eval(input("enter
the number1:"))
num2=eval(input("enter
the number2:"))
print(num11+num2)
```

```
In [13]: In [19]:
```

```
addition1()

enter the number1:20
enter the number2:30

-----------------------------------------
--------------------------------- -
NameError Traceback (most recent call las
t)
Cell In[9], line 1
In [20]:
```

```
----> 1 addition1()

Cell In[8], line 4, in addition1()
 2 num1=eval(input("enter the number1:"))
 3 num2=eval(input("enter the number2:"))
----> 4 print(num11+num2)

NameError: name 'num11' is not defined
```

defining function will not give the error

you will identify the error when you call the function only

syntax error only will get , when define the function call

```
# wap ask the user enter three numbers
# find the average
# implement the function call
n1=eval(input("enter number1:"))
n2=eval(input("enter number2:"))
n3=eval(input("enter number3:"))
avg=(n1+n2+n3)/3
print(f"the average of {n1},{n2} and {n3}
is {avg}")
```

```
enter number1:20
enter number2:30
enter number3:40
the average of 20,30 and 40 is 30.0
```

```
def average():
 n1=eval(input("enter number1:"))
 n2=eval(input("enter number2:"))
 n3=eval(input("enter number3:"))
 avg=(n1+n2+n3)/3 # round((n1+n2+n3)/3,2)
 out=round(avg,2)
 print(f"the average of {n1},{n2} and
{n3} is {out}")
average()
```

```
enter number1:3
enter number2:4
enter number3:4
the average of 3,4 and 4 is 3.67
```

```
bill=eval(input("enter the
bill amount:"))
tip_per=eval(input("enter the
tip in %:"))
tip_amount=bill*tip_per/100 #
1000*10/100=100
total_amount=bill+tip_amount
print(f"the total bill is:
{total_amount}")
```

```
enter the bill amount:1000
enter the tip in %:10
the total bill is: 1100.0
```

In [21]: In [24]:

```
def bill_pay():
 bill=eval(input("enter the
bill amount:"))
tip_per=eval(input("enter the
tip in %:"))
tip_amount=bill*tip_per/100 #
1000*10/100=100
total_amount=bill+tip_amount
 print(f"the total bill is:
{total_amount}")
bill_pay()
```

```
# WAP ask the user enter bill
amount
# ask the user enter tip
percentage # calculate total
bill
#implement the function
```

```
enter the bill amount:1000
enter the tip in %:10
the total bill is: 1100.0
```

```
bill_pay()
```

```
Out[24]: <function __main__.bill_pay()>
```

In [26]:

```
import random
random.randi
nt()
```

```
Out[26]: <bound method Random.randint of <random.Random object at 0x000001E4B912985
0>>
```

Note: **Functions-methods both are same**

In [27]:

```
def addition1():
 try:
 num1=eval(input("enter the
number1:"))
num2=eval(input("enter the
number2:"))
print(num1+num2)
 except Exception as e:
 print(e)

addition1()
```

```
enter the number1:p
name 'p' is not defined
```

```
# BASIC CODES ASSIGNMENT :
3days
```

In [28]:

```
# WAP ask the user enter a
number
# find it is an even or odd
# create a function on this

num=eval(input("enter the
number:")) if num%2==0:
 print("even")
else:
 print("odd")
```

```
enter the number:20
even
```

In [2]:

In [ ]: In [1]:

```
print("hello")
print(1)
def addition1():
 num1=eval(input("enter the
number1:"))
num2=eval(input("enter the
number2:"))
print(num1+num2)
 print(2)

print("good")
addition1()
print('bye')

# hello 1 good 20 30 50 2
bye
```

In [3]: In [ ]:

In [5]:

```
hello
1
good
enter the number1:20
enter the number2:30
50
2
bye
```

```
In [ ]: In [8]:

def even_odd():
 num=eval(input("enter the
number:"))
 if num%2==0:
 print("even")
 else:
 print("odd")


even_odd()

enter the number:30
even


offline:79039 43156 Piyush
DS JAN 2024 Offline online:
DS JAN 2024 Online
discussions healthy
discusiions
information post it
material post it
qn post it
```

```
qn post it

class information
doubt


def even_odd():
 try:
 num=eval(input("enter the
number:"))  if num%2==0:
 print("even")
 else:
 print("odd")
 except Exception as e:
 print(e)

even_odd()

enter the number:p
name 'p' is not defined


# Implement the above code
by providing a random value


import random

def even_odd():
 try:
 num=random.randint(10,100)
 if num%2==0:
 print(f"{num} is an even")
 else:
 print(f"{num} is an odd")
 except Exception as e:
 print(e)

even_odd()
```

```
         12 is an even
In [13]:
# wap ask the user get a random
number : num1 # ask the user
enter a number from keyboard :
num2 # if num1==num2:
print("in")
# other wise print("out")
# create the function on this

def game():
 n1=random.randint(1,10)
```

```
 n2=eval(input("enter the
number"))
 if n1==n2:
 print("in")
 else:
 print("out")

game()

enter the number9
out
```

```
 In [16]: def summ(): # inside bracket any values are there : NO
                print("summ function called")
          num1=eval(input("enter the number1:"))
          num2=eval(input("enter the number2:"))
          print(num1+num2)


         ################################################################
         def average(): # NO
          print("avergae function called")
          n1=eval(input("enter number1:"))
          n2=eval(input("enter number2:"))
          n3=eval(input("enter number3:"))
          avg=(n1+n2+n3)/3 # round((n1+n2+n3)/3,2)
          out=round(avg,2)
           print(f"the average of {n1},{n2} and {n3} is {out}")


         #######################################################################

         def bill_pay(): # NO
          print("billpay function called")
          bill=eval(input("enter the bill amount:"))
          tip_per=eval(input("enter the tip in %:"))
          tip_amount=bill*tip_per/100 # 1000*10/100=100
          total_amount=bill+tip_amount
          print(f"the total bill is: {total_amount}")



         #########################################################################

         def even_odd(): NO
          print("even_odd function called")
          num=eval(input("enter the number:"))
          if num%2==0:
          print("even")
          else:
          print("odd")

         ############################################################################
         def game():# NO
          print("game function called")
          n1=random.randint(1,10)
          n2=eval(input("enter the number"))
          if n1==n2:
          print("in")
          else:
          print("out")
```

```
In [18]:
piyush.summ()
print("===============
==========")
piusdh.average()
print("===============
==========")
bill_pay()
print("===============
==========")
even_odd()
print("===============
============") game()
```

```
summ function called
enter the number1:20
enter the number2:30
50
======================
===
avergae function
called
enter number1:20
enter number2:30
enter number3:40
the average of 20,30
and 40 is 30.0
======================
===
billpay function
called
enter the bill
amount:1000
enter the tip in %:10
the total bill is:
1100.0
======================
===
even_odd function
called
enter the number:55
odd
======================
=====
game function called
enter the number7
```

```
out
In [19]: dir(random)
Out[19]: ['BPF',
          'LOG4',
          'NV_MAGICCONST',
          'RECIP_BPF',
          'Random',
          'SG_MAGICCONST',
          'SystemRandom',
          'TWOPI',
          '_ONE',
          '_Sequence',
          '_Set',
          '__all__',
          '__builtins__',
          '__cached__',
          '__doc__',
          '__file__',
          '__loader__',
          '__name__',
          '__package__',
          '__spec__',
          '_accumulate',
          '_acos',
          '_bisect',
          '_ceil',
          '_cos',
          '_e',
          '_exp',
          '_floor',
          '_index',
          '_inst',
          '_isfinite',
          '_log',
          '_os',
          '_pi',
          '_random',
          '_repeat',
          '_sha512',
          '_sin',
          '_sqrt',
          '_test',
          '_test_generator',
          '_urandom',
          '_warn',
          'betavariate',
          'choice',
          'choices',
          'expovariate',
          'gammavariate',
          'gauss',
          'getrandbits',
          'getstate',
          'lognormvariate',
          'normalvariate',
          'paretovariate',
          'randbytes',
          'randint',
          'random',
          'randrange',
          'sample',
          'seed',
          'setstate',
```

till now we developed some functions

inside bracket there is no values

what ever you provide inside bracket is called as arguments or parameters

Function with arguments

```
def summ():
 num1=eval(input("enter the
number1:"))
 num2=eval(input("enter the
number2:"))
 add=num1+num2
 print(add)
```

In [ ]:

```
# Q1) in above function how many
variables are there # num1 num2 add
#Q2) how many input variable are
there: 2 num1 num2 # Q3) how many
output variables are there: 1 add
```

```
def summ(num1,num2):
 add=num1+num2
 print(add)
```

In [20]:

```
# In above inside function how many

lines are :2 summ(20,30)
```

```
50
```

```
def summ(num1,num2):
 add=num1+num2
 print(add)
```

In [25]: In [ ]:

```
summ(50,50)
```

```
100
```

```
def average():
 n1=eval(input("enter number1:"))
 n2=eval(input("enter number2:"))
 n3=eval(input("enter number3:"))
 avg=(n1+n2+n3)/3 #
round((n1+n2+n3)/3,2)
 out=round(avg,2)
 print(f"the average of {n1},{n2}
and {n3} is {out}")
```

```
'shuffle',
'triangular',
'uniform',
'vonmisesvariate',
'weibullvariate']
```

function with out arguments

In [26]:

```
30.0
```

```
def summ(num1,num2):
 print("num1:",num1)
 print("num2:",num2)
 add=num1+num2
 print(add)
```

```
In [27]:
summ(150,50)

num1: 150
num2: 50
200
```

```
def summ(num1):
 num2=200
 add=num1+num2
 print(add)

summ(200)
```

```
In [32]: In [35]:

400
```

```
num1=500
##################################
def summ(num1,num2):
 num1=2000
 add=num1+num2
 print(add)
##################################
###
num1=1000
summ(150,50)


#have you intialized any values
before function call #what are
the new values when you are
calling the function #what are
the new values when function is
executing
```

```
In [34]:
def average(n1,n2,n3):
 avg=(n1+n2+n3)/3
 out=round(avg,2)
 print(f"the average of {n1},{n2}
and {n3} is {out}")
average(20,30,40)

the average of 20,30 and 40 is

Out[34]: 1000
In [36]:
num1=500
num2=700
def summ(num1,num2):
 num1=2000
 add=num1+num2
 print(add)

num1=num2 # 700
num2=num1 # 700 is
summ(150,50) # it is not return any
```

```
200

num1=100
num1=1000
num1

   values
   add=num1+num2
   add

   # Function will not return any
   values
   # untill unless you mentiond return

   keyword inside the function 2050
```

```
Out[36]: 1400
```

In [ ]: In [37]: In [38]: In

[ ]:

```
def bill_pay():
 bill=eval(input("enter the
bill amount:"))
tip_per=eval(input("enter the
tip in %:"))
tip_amount=bill*tip_per/100 #
1000*10/100=100
total_amount=bill+tip_amount
 print(f"the total bill is:
{total_amount}")
```

In [39]: In [ ]:

In [ ]: In [ ]:

```
# Implement the above
function with arguments
```

```
def bill_pay(bill,tip_per):
 tip_amount=bill*tip_per/100
 total_amount=bill+tip_amount
 print(f"the total bill is:
{total_amount}")
bill_pay(1000,10)
```

```
the total bill is: 1100.0
```

```
bill_pay(
 eval(input("enter a
number:")),
eval(input("enter a
number:"))  )
```

```
enter a number:1000
enter a number:10
the total bill is: 1100.0
```

```
def even_odd():
 num=eval(input("enter the
number:"))  if num%2==0:
 print("even")
 else:
 print("odd")
```

```
In [1]: In [ ]:
```

```
def even_odd(num):
 if num%2==0:
 print("even")
 else:
 print("odd")

even_odd(10)

even

- Function with out arguments

- Function with arguments

- Default arguments

- Local vs global

- return

- function in function

# Practice every day
# Attend the classes

# 9am to 10.30 lab
# 10.30 to 12pm



# Functions with out arguments def even_odd():
 num=eval(input("enter the number:"))  if num%2==0:
 print("even")
 else:
 print("odd")


# Functions with arguments
def even_odd(num):
 if num%2==0:
 print("even")
 else:
 print("odd")

even_odd(20)
```

```
even
```

Default arguments

```
# we can fix the argument values # is called default arguments
In [5]: In [10]:
```

```
summ(20)

# num1=20 num2=100

num1: 20
num2: 100
120

def
summ(num1=200,num2
=100):
print("num1:",num1
)
print("num2:",num2
)  add=num1+num2
 print(add)

summ()

summ()
```

```
def
summ(num1,num2=100
):
print("num1:",num1
)
print("num2:",num2
)  add=num1+num2
 print(add)
```

```
Out[10]: 'summ()'
```

```
 In [ ]:
 def summ():
  num1=20
  num2=100
  add=num1+num2
  print(add)

 summ()

 def
 summ(num1,num2):
 add=num1+num2
In [14]:
```

```
 print(add)

summ(20,100)

def
summ(num1,num2=1
00):
add=num1+num2
 print(add)

summ(20)
```

In [ ]: In [15]:

In [ ]: In [17]:

```
def summ(num1=100,num2):
 add=num1+num2
 print(add)

summ(100)
```

```
Cell In[14], line 1
 def summ(num1=100,num2):
 ^
SyntaxError: non-default argument
follows default argument
```

Note: Default arguments always at last

```
avg(n1,n2,n3=100) # Works
avg(n1,n2=100,n3=100) # Works
avg(n1=100,n2=100,n3=100) # Works
avg(n1=100,n2,n3) # Fail
avg(n1=100,n2=100,n3) # Fail
avg(n1=100,n2,n3=100) # Fail
avg(n1,n2=100,n3=100) # Works
```

```
def avg(n1,n2,n3=100):
 average=(n1+n2+n3)/3
 print(average)

avg(200,300) # n1=200 n2=300
n3=100
```

```
200.0
```

```
def avg(n1,n2,n3=100):
 average=(n1+n2+n3)/3
 print(average)

avg(200,300)
```

```
def avg(n1=100,n2=100,n3=100):
 average=(n1+n2+n3)/3
 print(average)

avg() # n1=100 n2=100 n3=100
```

```
100.0
```

In [18]: In [19]: In [20]: In

[21]:

```
def avg(n1=100,n2,n3):
 average=(n1+n2+n3)/3
 print(average)

avg(200,300)
```

```
Cell In[18], line 1
 def avg(n1=100,n2,n3):
 ^
SyntaxError: non-default argument
follows default argument
```

```
def avg(n1=100,n2=100,n3):
 average=(n1+n2+n3)/3
 print(average)

avg(200)
```

```
Cell In[19], line 1
 def avg(n1=100,n2=100,n3):
 ^
SyntaxError: non-default argument
follows default argument
```

```
def avg(n1=100,n2,n3=100):
 average=(n1+n2+n3)/3
 print(average)

avg(200)
```

```
Cell In[20], line 1
 def avg(n1=100,n2,n3=100):
 ^
SyntaxError: non-default argument
follows default argument
```

```
def avg(n1,n2=100,n3=100):
 average=(n1+n2+n3)/3
 print(average)

avg(200)
```

```
133.33333333333334
In [27]:
```

```
In [26]: In [28]:
```

```
def summ(num1,num2=100):
 add=num1+num2
 print(add)

summ(num1=200)

# num2=100
# num1=200
# 300

# Have you provided any value
before define the function # what
are you provided while define the
function  # what are you provided
while calling the function  #
what are you provided while
running the function
```

```
300
```

```
def summ(num1,num2=100):
 add=num1+num2
 print(add)

summ(num1=200,num2=500)

# num2=100
# num1=200
# num2=500
# 500+200=700
```

```
700

def summ(num1,num2=100):
 num2=1000
 add=num1+num2
 print(add)
```

In [30]:

In [32]: In [33]:

```
num2=5000

def summ(num1,num2=3000):
 num2=4000
 add=num1+num2
 print(add)

num2=7000

summ(num1=200,num2=2000) #
4200

# num2=5000
# num2=3000 define
# num2=7000
# num2=2000
```

```
      summ(num1=200)

      # num2=100
      # num1=200
      # num2=1000
      # 1200

      1200

# num2=4000

print(num2)

4200
7000


num2=5000

def summ(num1,num2=3000):
 num2=4000
 add=num1+num2
 print(add)

num2=7000

summ(num1=200,num2=2000)

4200
```

```
# Function with out
arguments
def bill_pay1():
 bill=eval(input("enter the
bill amount:"))
tip_per=eval(input("enter
the tip in %:"))
tip_amount=bill*tip_per/100

total_amount=bill+tip_amount
 print(f"the total bill is:
{total_amount}")
# Function with arguments
def bill_pay2(bill,tip_per):
 tip_amount=bill*tip_per/100

total_amount=bill+tip_amount
 print(f"the total bill is:
{total_amount}")
# Create default argument as
tip_per=10 def
bill_pay3(bill,tip_per=10):
 tip_amount=bill*tip_per/100

total_amount=bill+tip_amount
 print(f"the total bill is:
{total_amount}")
bill_pay3(1000)
```

```
the total bill is: 1100.0
```

```
In [35]: In [ ]:
```

```
In [38]:
def bill_pay3(tip_per=10):
 print("per:",tip_per)
 bill=1000
 tip_amount=bill*tip_per/100

total_amount=bill+tip_amount
 print(f"the total bill is:
{total_amount}")
bill_pay3(1000)

# 1000
# 1000*1000/10=

per: 1000
the total bill is: 11000.0


def
bill_pay4(bill,tip_per=10):
 tip_amount=bill*tip_per/100

total_amount=bill+tip_amount
 print(f"the total bill is:
{total_amount}")
bill_pay4(1000) # bill=1000
tip=10



def bill_pay5(tip_per=10):
 print("per:",tip_per)
 bill=1000
 tip_amount=bill*tip_per/100
```

```
total_amount=bill+tip_amount
 print(f"the total bill is:
{total_amount}")
bill_pay5(1000) # bill=1000
tip=1000

def bill_pay6(tip_per=10):
 print("per:",tip_per)
 bill=1000
 tip_amount=bill*tip_per/100

total_amount=bill+tip_amount
 print(f"the total bill is:
{total_amount}")
bill_pay6() # bill=1000
tip_per=10


def summ(n2=50):
 n1=20
 print(n1+n2)

summ() # n2=50 n1=20

70
In [ ]:
```

```
In [ ]: In [40]: In [41]: In

[43]:

In [45]: In [55]:

In [56]:
# wap ask the user enter
salary: 100000
```

```
# ask the user enter tax per
: 10
# calculate tax

# Implement functio with out
argument
# function with argument
# function default argument
: tax_per=10


def tax():
 salary=eval(input("enter
the salary:"))
tax_per=eval(input("enter
the tax percentage:"))
tax_pay=(salary*tax_per)/100
 print(f"total tax pay is
{tax_pay}") tax()


def tax(salary,tax_per):

tax_pay=(salary*tax_per)/100
 print(f"total tax pay is
{tax_pay}") tax(100000,10)

total tax pay is 10000.0


def tax(salary,tax_per=10):

tax_pay=(salary*tax_per)/100
 print(f"total tax pay is
{tax_pay}") tax(100000)

total tax pay is 10000.0
```

```
def tax1(salary,tax_per=10):

tax_pay=(salary*tax_per)/100
 print(f"total tax pay is
{tax_pay}") tax1(100000)

total tax pay is 10000.0

�������������


#tax_pay
# tax_pay we are getting
inside the function   # if
you want to use outside then
use return


def tax1(salary,tax_per=10):

tax_pay=(salary*tax_per)/100
 return(tax_pay)

# we are askin function to


return(give me) some value


tax_pay=tax1(10000) # is


giving tax_pay
```

```
In [57]: In [ ]:     0):
                     tax_pay=(salary*tax_p
                     er)/100
                     return(tax_pay)

                     tax_pay=tax1(10000)
                     print(tax_pay)


In [71]: In [70]:
                     def
                     tax1(salary,tax_per=1
                     0):
                     tax_pay1=(salary*tax_
                     per)/100
                     return(tax_pay1,salar
                     y)
                     tax_pay1,salary=tax1(
                     10000)

                     print(tax_pay1)
                     print(salary)
print(tax_pay)
                     1000.0
1000.0               10000


def                  # wap take three
tax1(salary,tax_per=1numbers
# do the sum           # return sum and
# do the avergae       avergae

Out[70]: (1, 2, 3)
                        summ=n1+n2+n3
In [73]:                avg=summ/3
                        return(summ,avg)
                       summ,avg=sumavg(10,2
                       0,30) print(summ)
                       print(avg)

                       60
                       20.0


                       add=0

In [77]: In [ ]:       def summ(n1,n2):
                        add=n1+n2 #
                       add=10+20=30  return
                       add
                       add=summ(10,20)
                       print(add)

                       30


                       # 9 to 10.30
def                    # 10.30 to 12
sumavg(n1,n2,n3):
 In [ ]: In [ ]: In [1]:
```

- Function with default

argument - return

```
                             # game program
                             # ask the user get a
                             random number 1,10:n1 #
                             ask the user enter number
                             :n2
                             # if n1==n2
                             # print("in")
                             # else:
                             # print("out")

 In [2]: In [3]:

                             import random
                             def game():
                              n1=random.randint(1,10)
                              n2=eval(input("enter the
                             number:"))  if n1==n2:
                              print("in")
                              else:
                              print("out")

                             game()

                             enter the number:6
                             out
```

- Functions with out arguments

- Function with arguments

```
import random
def game1(n2):
 n1=random.randint(1,10)
 if n1==n2:
 print("in")
 else:
 print("out")

game1(7)

out
```

```
def game1(n2=7):
 n1=random.randint(1,10)
 if n1==n2:
 print("in")
 else:
 print("out")

game1()
```

```
import random

        out
In [12]:
```

```
 if n1==n2:
 print("in")
 print("hello")
 return(10000,'award')
 else:
 print("out")
 return(0,'no award')

money,award=game1()
print(money)
print(award)

out
0
no award
```

```
sir in that assignemnet reverse
of the numerical value and sum of
firt two last two digits was bit
confusing
```

```
In [ ]: In [7]:
```

```
n1=12345 # 54321
o1=n1%10
new_n2=12345//10
new_n2
```

```
import random
def game1(n2=7):
 n1=random.randint(1,10)

 Out[7]: 1234
           5)
In [10]:   int(n1[::-1
n1=str(1234 ])

Out[10]: 54321
```

```
In [14]:
import random
def game1(n2):

n1=random.randint(
1,10)  if n1==n2:
```

```
print("in")
else:
print("out")

game1(6)

out
```

```
In [ ]: def summ():
         num1=eval(input("enter the number1:"))
         num2=eval(input("enter the number2:"))
         print(num1+num2)

        # two variables are intilaised num1 and num2
        # two variables are intilised inside the function
        # these varuables are called local variables
```

```
#################################################################
def average(): # NO
 n1=eval(input("enter number1:"))
 n2=eval(input("enter number2:"))
 n3=eval(input("enter number3:"))
 avg=(n1+n2+n3)/3 # round((n1+n2+n3)/3,2)
 out=round(avg,2)
   print(f"the average of {n1},{n2} and {n3} is {out}")
```

```
#######################################################################

def bill_pay(): # NO
 bill=eval(input("enter the bill amount:"))
 tip_per=eval(input("enter the tip in %:"))
 tip_amount=bill*tip_per/100 # 1000*10/100=100
 total_amount=bill+tip_amount
 print(f"the total bill is: {total_amount}")
```

```
#########################################################################

def even_odd():
 num=eval(input("enter the number:"))
 if num%2==0:
 print("even")
 else:
 print("odd")

###########################################################################
def game():# NO
 n1=random.randint(1,10)
 n2=eval(input("enter the number"))
 if n1==n2:
 print("in")
 else:
 print("out")
```

In [15]: In [18]:

```
num1=eval(input("enter the number1:"))
num2=eval(input("enter the number2:"))
def summ():
 print(num1+num2)

summ()

enter the number1:10
enter the number2:20
30
```

```
n1=random.randint(1,10) # global
def game():
 n2=eval(input("enter the number")) # local
 if n1==n2:
 print("in")
 else:
 print("out")

game()
print("random:",n1)
print("user:",n2)

# step-1: some random number strored in n1
# step-2: you define the function
# step-3: call the function
# it is going to step-2 and execute the lines
# user value will store in n2
# apply the condition print in or out
# step-4: you are print the random n1:
global variable # step-5: you are print
the n2: local variable
```

```
enter the number6
out
random: 1
```

```
-----------------------------------------
--------------------------------- -
NameError Traceback (most recent call las
t)
Cell In[18], line 11
 9 game()
 10 print("random:",n1)
---> 11 print("user:",n2)

NameError: name 'n2' is not defined
In [19]: In [ ]:
```

```
In [20]:
def even_odd(num):
 if num%2==0:
 print("even")
 else:
 print("odd")

num=eval(input("enter the number"))
even_odd(num)
```

```
enter the number20
```

```
even
```

```
def even_odd(num):
 if num%2==0:
 print("even")
 else:
 print("odd")


even_odd(20)

#####################################

Local even_odd(eval(input("enter the

number"))) # Local

num=eval(input("enter the number")) #
global
even_odd(num)

num=random.randint(1,50) # global
even_odd(num)


def even_odd(num11):
 if num11%2==0:
 print("even")
 else:
 print("odd")

even_odd(eval(input("enter the number")))
print(num11)
```

```
enter the number20
even
```

```
-----------------------------------------
--------------------------------- -
NameError Traceback (most recent call las
t)
Cell In[20], line 8
 5 print("odd")
 7 even_odd(eval(input("enter the
number")))
----> 8 print(num11)

NameError: name 'num11' is not defined
```

variables Inside the function is called as local variable
variables outside the function is called as global variable
If you want to use a global variable inside the function, intilaize that before call the function

```
                num11%2==0:
In [25]:        print("even")
def             else:
even_odd(num11)  print("odd")
:   if
```

```
even_odd(num11) 10
num11=20
```

Out[25]: 10

In [ ]: In [28]:

```
def bill_pay1():
 global total_amount
 bill=eval(input("enter the bill
amount:"))
 tip_per=eval(input("enter the tip
in %:"))
 tip_amount=bill*tip_per/100 #
1000*10/100=100
total_amount=bill+tip_amount
 print(f"the total bill is:
{total_amount}")

bill_pay1()
```

```
enter the bill amount:1000
enter the tip in %:10
the total bill is: 1100.0
```

In [29]:

```
use local variables, outside the

function with out using return

global
```

```
total_amount
```

Out[29]: 1100.0

In [ ]: In [31]:

```
def ar_oper(n1,n2,n3):
 global summ,avg
 summ=n1+n2+n3
 avg=round(summ/3,2)
ar_oper(2,4,5)
print(summ)
print(avg)
```

```
# wap ask take three numbers
as arguments # create add and
avg variables inside the func
# calculate that add and avg
# print outside the function
with out uisng return
```

```
11
3.67
```

Functions in Functions

In [32]: In [33]: In [34]:

```
def greet1():
 print("hello")
def greet2():
 print("how are
you")
```

```
greet1()
greet2()

hello
how are you

def greet1():
 print("hello")
def greet2():
 greet1()
 print("how are
you")
greet2()
```

```
how are you

def greet1():
 greet2()
 print("hello")
def greet2():
 print("how are
you")
greet1()

how are you
hello
```

```
hello
In [35]:
def greet1():
 greet2()
 print("hello")

def greet2():
 greet1()
 print("how are you")

greet1()

-------------------------------------------
--------------------------------- -
RecursionError Traceback (most recent
call las t)
Cell In[35], line 9
 6 greet1()
 7 print("how are you")
----> 9 greet1()

Cell In[35], line 2, in greet1()
 1 def greet1():
----> 2 greet2()
 3 print("hello")

Cell In[35], line 6, in greet2()
 5 def greet2():
----> 6 greet1()
```

```
 7 print("how are you")

Cell In[35], line 2, in greet1()
 1 def greet1():
----> 2 greet2()
 3 print("hello")

Cell In[35], line 6, in greet2()
 5 def greet2():
----> 6 greet1()
 7 print("how are you")

 [... skipping similar frames: greet1 at
line 2 (1484 times), greet2 at line 6
(1484 times)]

Cell In[35], line 2, in greet1()
 1 def greet1():
----> 2 greet2()
 3 print("hello")

Cell In[35], line 6, in greet2()
 5 def greet2():
----> 6 greet1()
 7 print("how are you")

RecursionError: maximum recursion depth
exceeded
```

```
 In [ ]: def greet1():
          print("hello")

         def greet2():
          print("how are you")

         greet1()
         greet2()
         ###############################################################################

         def greet1():
          print("hello")

         def greet2():
          greet1()
          print("how are you")

         greet2()
         ###############################################################################
```

```
def greet1():
 greet2()
 print("hello")

def greet2():
 print("how are you")

greet1()
###########################################################################

def greet1():
 greet2()
 print("hello")

def greet2():
 greet1()
 print("how are you")

greet1()
```

In [ ]:

In [ ]: In [ ]:

```
# calcultor program
# create 4 functions with
arguments
# fun1: add
# fun2: sub
# fun3: mul
# fun4: div

# print("if you want use add
opertion pls enter 1") #
print("if you want use sub
opertion pls enter 2") #
print("if you want use mul
opertion pls enter 3") #
print("if you want use div
opertion pls enter 4")

# option=eval(input("choose
option 1,2,3,4")) # if
option==1:
```

```
# n1=
# n2=
# add(n1,n2)
# elif option==2:
# sub()
```

In [37]:

```
def add():
 total=num1+num2
 print(f"sum is {total}")


def sub():
 diff=num1-num2
 print(f"subtraction is
{diff}")
def mul():
 fm=num1*num2
 print(f"multiplication is
{fm}")
def div():
 fd=round(num1/num2,2)
 print(f"division is {fd}")

print ("if you want to you
add operation pls enter 1")
print ("if you want to you
sub operation pls enter 2")
print ("if you want to you
mul operation pls enter 3")
print ("if you want to you
div operation pls enter 4")

option=eval(input("choose
option 1,2,3,4: "))
num1=eval(input("enter num1:
"))
num2=eval(input("enter num2:
"))
if option==1:
 add()
elif option==2:
 sub()
elif option==3:
 mul()
elif option==4:
 div()
else:
 print("Please select
appropriate option")
```

```
if you want to you add
operation pls enter 1 if you
want to you sub operation pls
enter 2 if you want to you
mul operation pls enter 3 if
you want to you div operation
pls enter 4 choose option
1,2,3,4: 1
enter num1: 10
enter num2: 20
sum is 30
```

```
In [38]:
def add(n1,n2):
 n3 = n1+n2
 print(f"{n3}")
def sub(n1,n2):
 n3= n1-n2
 print(f"{n3}")
def mul(n1,n2):
 n3 =n1*n2
 print(f"{n3}")
def div(n1,n2):
 n3=n1/n2
 print(f"{n3}")

print("if you want to use add
function enter 1") print("if
you want to use sub function
enter 2") print("if you want
to use mul function enter 3")
print("if you want to use div
function enter 4") option =
eval(input("Enter you
option"))

if option == 1:
 n1 = eval(input("Enter n1 :
"))
 n2 = eval(input("Enter n2 :
"))
 add(n1,n2)
elif option == 2:
 n1 = eval(input("Enter n1 :
"))
 n2 = eval(input("Enter n2 :
"))
 sub(n1,n2)
elif option == 3:
 n1 = eval(input("Enter n1 :
"))
 n2 = eval(input("Enter n2 :
"))
 mul(n1,n2)
else:
 n1 = eval(input("Enter n1 :
"))
 n2 = eval(input("Enter n2 :
"))
 div(n1,n2)
```

```
if you want to use add
function enter 1 if you want
to use sub function enter 2
if you want to use mul
function enter 3 if you want
to use div function enter 4
Enter you option3
Enter n1 : 7
Enter n2 : 8
56
```

```
In [39]:
def summ():
 add=n1+n2
 return(add)
def sub():
 diff=n1-n2
 return(diff)
def mul():
 product=n1*n2
 return(product)
def div():
 division=n1/n2
 return(division)
print('if you want to use add
enter 1')
print('if you want to use sub
enter 2')
print('if you want to use mul
enter 3') print('if you want
to use div enter 4')
option=eval(input('choose
option 1,2,3,4'))
n1=eval(input('enter num1 :
'))
n2=eval(input('enter num2 :
'))
if option==1:
 result=summ()
 print(f'sum of {n1} and {n2}
is {result}') elif option==2:
 result=sub()
 print(f'difference of {n1}
and {n2} is {result}') elif
option==3:
 result=mul()
 print(f'product of {n1} and
{n2} is {result}') else:
 result=div()
 print(f'division of {n1} and
{n2} is {result}')
```

```
if you want to use add enter
1
if you want to use sub enter
2
if you want to use mul enter
3
if you want to use div enter
4
```

```
choose option 1,2,3,44
enter num1 : 5
enter num2 : 6
division of 5 and 6 is 0.8333333333333334
```

In [40]:

```python
# calculator
# create 4 functions
# fun1: add
# fun2: sub
# fun3: mul
# fun4: div
# print("Ask user for option to choose option") # 1 for add, 2 for sub, 3 for mul, 4 for div
# ask num1 and num2


def add(n1, n2):
 return n1+n2

def sub(n1, n2):
 return n1-n2

def mul(n1, n2):
 return n1*n2

def div(n1, n2):
 return n1/n2

print("Enter your choose your option: 1 for Addition")
print("Enter your choose your option: 2 for Substraction")
print("Enter your choose your option: 3 for Multiplication") print("Enter your choose your option: 4 for Division")
option=eval(input("Enter option:"))
num1=eval(input("Enter num1:"))
num2=eval(input("Enter num2:"))

if option==1:
 result=add(num1, num2)
elif option==2:
 result=sub(num1, num2)
elif option==3:
 result=mul(num1, num2)
elif option==4:
 result=div(num1, num2)
else:
 print("Kindly provide proper option")
print("Result :", result)
```

```
Enter your choose your option: 1 for Addition
Enter your choose your option: 2 for Substraction
Enter your choose your option: 3 for Multiplication
Enter your choose your option: 4 for Division
Enter option:3
Enter num1:10
Enter num2:2
Result : 20
```

```
In [42]: In [ ]:
```

```
n1 = eval(input("Enter 1st
number :")) n2 =
eval(input("Enter 2nd
number :")) option =
eval(input("Enter option
1,2,3,4 :")) def add():
 return(n1+n2)
def sub():
 return(n1-n2)
def mul():
 return(n1*n2)
def div():
 return(n1/n2)
if option==1:
 ans=add()
elif option==2:
 ans=sub()
elif option==3:
 ans=mul()
elif option==4:
 ans=div()
else:
 print("use valid option")

print(ans)
```

```
Enter 1st number :20
Enter 2nd number :20
Enter option 1,2,3,4 :2
0
```

```
# for loop
if-else
```