```
In [1]: num1=eval(input("enter a number1:"))
        num2=eval(input("enter a number2:"))
        add=num1+num2
        print(add)
```

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

Types of erreors

```
In [2]: # error-1: Name error:  num3 not defined
        num1=eval(input("enter a number1:"))
        num2=eval(input("enter a number2:"))
        add=num1+num3
        print(add)
```

```
enter a number1:100
enter a number2:300

---------------------------------------------------------------------------
-
NameError                                 Traceback (most recent call las
t)
Cell In[2], line 4
      2 num1=eval(input("enter a number1:"))
      3 num2=eval(input("enter a number2:"))
----> 4 add=num1+num3
      5 print(add)

NameError: name 'num3' is not defined
```

```
In [3]: # error-2:
        num1=eval(input("enter a number1:")
        num2=eval(input("enter a number2:"))
        add=num1+num2
        print(add)
```

```
  Cell In[3], line 2
    num1=eval(input("enter a number1:")
             ^
SyntaxError: '(' was never closed
In [5]: # error-3:
num1=eval(input("enter a number1:"))
num2=eval(input("enter a number2:"))
add=num1/num2
print(add)
```

```
enter a number1:100
enter a number2:0

---------------------------------------------------------------------------
-
ZeroDivisionError                         Traceback (most recent call las
t)
Cell In[5], line 4
      2 num1=eval(input("enter a number1:"))
      3 num2=eval(input("enter a number2:"))
----> 4 add=num1/num2
      5 print(add)

ZeroDivisionError: division by zero
```

```
In [6]: # error-4:
        num1=input("enter a number1:")     # string
        num2=eval(input("enter a number2:")) # int
        add=num1/num2  # string/int
        print(add)
```

```
enter a number1:10
enter a number2:20

---------------------------------------------------------------------------
-
TypeError                                 Traceback (most recent call las
t)
Cell In[6], line 4
      2 num1=input("enter a number1:")
      3 num2=eval(input("enter a number2:"))
----> 4 add=num1/num2
      5 print(add)

TypeError: unsupported operand type(s) for /: 'str' and 'int'
```

```
In [7]: # error-5:
        num1=eval(input("enter a number1:"))
        num2=eval(input("enter a number2:"))
        add=num1/num2
        print add
```

```
  Cell In[7], line 5
    print add
    ^
SyntaxError: Missing parentheses in call to 'print'. Did you mean print
(...)?
```

```
In [9]: # error-6:
        num1=eval(input("enter a number1:"))
        num2=eval(input("enter a number2:"))
        add=num1/num2
        Print(add)
        print('hello')
```

```
enter a number1:100
enter a number2:200

---------------------------------------------------------------------------
-
NameError                                 Traceback (most recent call las
t)
Cell In[9], line 5
      3 num2=eval(input("enter a number2:"))
      4 add=num1/num2
----> 5 Print(add)
      6 print('hello')

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

Possible errors:

- Name error
- Syntax error:'(' was never closed
- Zero division error: division by zero
- Type error: unsupported operand type(s) for /: 'str' and 'int'
- Missing parentheses in call to 'print'. Did you mean print(...)?

suppose you have written 500 lines of code at 200 line you got an error this error is migh not impact your code so you want execute after 200 line also

Exception handling

try-except

- there are two blocks
- one try block
- another is exception block
- your code will be there in try block
- error capture will happens in except block
- indentation : space
- red : space/indentation not maintained properly
- indenation has min 4 charcters
- : is there, then indentation will be there

In [ ]:
```
try:
    num1=10
    num2=20
    num3=20
    add=num1+num2+num3
    print(add)

except:
    print("error is there,check code")
```

In [10]:
```
num1=10
num2=20
num3=20
add=num1+num2+num3
print(add)
```

```
50
```

In [ ]:
```
try:
    num=10
num2=20
   add=num+num2
   print(add)                # this will not work
```

In [14]:
```
# Method-1
try:
    num1=10
    num2=30
    num3=20
    add=num1/num2
    print(add)

except:
    print("error is there,check code")
```

```
0.3333333333333333
```

In [15]:
```
# Method-2
try:
    num1=10
    num2=0
    num3=20
    add=num1/num2
    print(add)

except:
    print("error is there,check code")
```

```
error is there,check code
```

In [13]:
```
num1=10
num2=0
add=num1/num2   # 10/0  ==== > zero
print(add)
```

```
---------------------------------------------------------------------------
-
ZeroDivisionError                         Traceback (most recent call las
t)
Cell In[13], line 3
      1 num1=10
      2 num2=0
----> 3 add=num1/num2   # 10/0  ==== > zero
      4 print(add)

ZeroDivisionError: division by zero
```

In [16]:
```
# wap ask the user enter 3 numbers and find the avergae
num1=eval(input("enter number-1:"))
num2=eval(input("enter number-2:"))
num3=eval(input("enter number-3:"))
avergae=(num1+num2+num3)/3
print("The average of {},{} and {} is {}".format(num1,num2,num3,avergae))
```

```
enter number-1:20
enter number-2:30
enter number-3:40
The average of 20 , 30 and 40 is 30.0
```

In [18]:
```
try:
    num1=eval(input("enter number-1:"))
    num2=eval(input("enter number-2:"))
    num3=eval(input("enter number-3:"))
    avergae=(num1+num2+num3333)/3
    print("The average of {},{} and {} is {}".format(num1,num2,num3,avergae

except:
    print("check the code")

# error-1: name error
```

```
enter number-1:20
enter number-2:30
enter number-3:40
check the code
```

In [21]:
```
try:
    num1=eval(input("enter number-1:"))
    num2=eval(input("enter number-2:"))
    num3=eval(input("enter number-3:"))
    avergae=(num1+num2+num3)/0
    print("The average of {},{} and {} is {}".format(num1,num2,num3,avergae

except Exception as e:
    print(e)
```

```
enter number-1:20
enter number-2:30
enter number-3:40
division by zero
```

In [22]:
```
# wap ask the user enter a number and find the square of the number

num=eval(input("enter a number:"))
print("the square of {} is {}".format(num,num*num))
```

```
enter a number:10
the square of 10 is 100
```

In [27]:
```
try:
    num=eval(input("enter a number:"))
    print("the square of {} is {}".format(num,num222*num))

except Exception as e:
    print(e)
```

```
enter a number:10
name 'num222' is not defined
```

In [ ]:
```
try:
    num=eval(input("enter a number:"))
    print("the square of {} is {}".format(num,num222*num))

except Exception as e:
    print(e)           # capture the error
    print('hello')
```

Case-1

In [32]:
```
try:
    num=eval(input("enter a number:"))
    print("the square of {} is {}".format(num,num222*num))

except Exception as e:
    print(e)           # capture the error
    print('hello')     # hello
```

```
enter a number:100
name 'num222' is not defined
hello
```

Case-2

In [33]:
```python
try:
    num=eval(input("enter a number:"))
    print("the square of {} is {}".format(num,num222*num))

except Exception as e:
    print('hai')
    print(e)
    print('hello')
```

```
enter a number:100
hai
name 'num222' is not defined
hello
```

Case-3

In [34]:
```python
try:
    num=eval(input("enter a number:"))
    print("the square of {} is {}".format(num,num222*num)) # error

except Exception as e:
    print('hai')
    print(e)
    print('hello')

print('python')
print(1)
print(2)

# num= 100
# error======= exception
              # hai
              # error
              # hello

# python
# 1
# 2
```

```
enter a number:100
hai
name 'num222' is not defined
hello
python
1
2
```

Case-4

```
In [37]: try:
             print('hello')
```

```
  Cell In[37], line 2
    print('hello')
                  ^
SyntaxError: incomplete input
```

- if your code is with out error , try block will execute
- if your code has errors,it will redirect to except block
- what ever you written inside except block that will run
- try and exception is together
- ones try and exceptions code are complete, it try to look any other lines of code are there
- Try-exception will not capture syntax error

Case-5

```
In [38]: print("=============================")
         print(3+5)
         print('****************************')

         try:
             num=eval(input("enter a number:"))
             print("the square of {} is {}".format(num,num222*num)) # error

         except Exception as e:
             print('hai')
             print(e)
             print('hello')

         print('python')
         print(1)
         print(2)
```

```
=============================
8
****************************
enter a number:20
hai
name 'num222' is not defined
hello
python
1
2
```

Case-6

In [39]:
```python
print("=============================")
print(3+5)
print('****************************')

try:
    num=eval(input("enter a number:"))
    print("the square of {} is {}".format(num,num222*num)) # error

print('out')

except Exception as e:
    print('hai')
    print(e)
    print('hello')

print('python')
print(1)
print(2)
```

```
  Cell In[39], line 9
    print('out')
    ^
SyntaxError: expected 'except' or 'finally' block
```

Case-7

In [41]:
```python
print("=============================")
print(3+5)
print('****************************')

try:
    num=eval(input("enter a number:"))
    print("the square of {} is {}".format(num,num222*num)) # error

 print('out')

except Exception as e:
    print('hai')
    print(e)
    print('hello')

print('python')
print(1)
print(2)
```

```
  File <tokenize>:9
    print('out')
    ^
IndentationError: unindent does not match any outer indentation level
```

All points

- there are two blocks
- one try block

- another is exception block
- your code will be there in try block
- error capture will happens in except block
- indentation : space
- red : space/indentation not maintained properly
- indenation has min 4 charcters
- : is there, then indentation will be there
- if your code is with out error , try block will execute
- if your code has errors,it will redirect to except block
- what ever you written inside except block that will run
- try and exception is together
- ones try and exceptions code are complete, it try to look any other lines of code are there
- Try-exception will not capture syntax errorm

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```