In [1]:
```
def func1():
    print('This is a function')
func1()
```

```
This is a function
```

In [2]:
```
def func1():
    def func2():
        print('This is a function inside a function. Second function')
    print('This is an outside function. First function.')
    func2()
func1()
```

```
This is an outside function. First function.
This is a function inside a function. Second function
```

In [3]:
```
def add(a, b):
    sum = a+b
    return sum
    print(sum)
    def mul(a, b):
        pro = a*b
        return pro
    print(pro)
    mul(2,3)
add(2,3)
```

Out[3]: 5

In [4]:
```
def add(a, b):
    sum = a+b
    return sum
    def mul(a, b):
        pro = a*b
        return pro
res1 = add(2,3)
res2 = mul(2,3)
print(res1)
print(res2)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
C:\Users\PRATIK~1\AppData\Local\Temp/ipykernel_5332/3400415773.py in <module>
      7     mul(2,3)
      8 res1 = add(2,3)
----> 9 res2 = mul(2,3)
     10 print(res1)
     11 print(res2)

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

In [5]:
```
def add(a, b):
    sum = a+b
    return sum
    def mul(a, b):
        pro = a*b
        return pro
res1 = add(2,3)
res2 = mul(2,3)
print(res1)
print(res2)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
C:\Users\PRATIK~1\AppData\Local\Temp/ipykernel_5332/4014288858.py in <module>
      6         return pro
      7 res1 = add(2,3)
----> 8 res2 = mul(2,3)
      9 print(res1)
     10 print(res2)

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

In [6]:
```
def add(a, b):
    sum = a+b
    return sum
    def mul(a, b):
        pro = a*b
        return pro
    mul()
add()
res1 = add(2,3)
res2 = mul(2,3)
print(res1)
print(res2)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
C:\Users\PRATIK~1\AppData\Local\Temp/ipykernel_5332/2520525657.py in <module>
      6         return pro
      7     mul()
----> 8 add()
      9 res1 = add(2,3)
     10 res2 = mul(2,3)

TypeError: add() missing 2 required positional arguments: 'a' and 'b'
```

In [7]:
```
def add(a, b):
    sum = a+b
    return sum
    def mul(a, b):
        pro = a*b
        return pro
    mul(2,3)
add(2,3)

print(add())
print(mul())
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
C:\Users\PRATIK~1\AppData\Local\Temp/ipykernel_5332/2836380321.py in <module>
      8 add(2,3)
      9
---> 10 print(add())
     11 print(mul())

TypeError: add() missing 2 required positional arguments: 'a' and 'b'
```

In [8]:
```
def add(a, b):
    sum = a+b
    return sum
    def mul(a, b):
        pro = a*b
        return pro
    mul(2,3)
add(2,3)

print(add(2,3))
print(mul(2,3))
```

```
5

---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
C:\Users\PRATIK~1\AppData\Local\Temp/ipykernel_5332/498508623.py in <module>
      9
     10 print(add(2,3))
---> 11 print(mul(2,3))

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

In [9]:
```
price = 10000
if(price>3000)
    print('No!')
else:
    print('Yes!')
```

```
  File "C:\Users\PRATIK~1\AppData\Local\Temp/ipykernel_5332/2194114329.py", line 2
    if(price>3000)
                  ^
SyntaxError: expected ':'
```

In [10]:
```
price = 10000
if(price>3000):
    print('No!')
else:
    print('Yes!')
```

```
No!
```

In [11]:
```
price = 10000
print(price/0)
```

```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
C:\Users\PRATIK~1\AppData\Local\Temp/ipykernel_5332/2930919672.py in <module>
      1 price = 10000
----> 2 print(price/0)

ZeroDivisionError: division by zero
```

In [12]:
```
arr = [1, 2, 3]
```

In [13]:
```
arr = [1, 2, 3]
try:
    print("First item", arr[0])
```

```
    print("Second item", arr[1])
except:
    print("Error occured!")
```

```
First item 1
Second item 2
```

In [14]:
```
arr = [1, 2, 3]
try:
    print("First item" % arr[0])
    print("Second item "% arr[1])
except:
    print("Error occured!")
```

```
Error occured!
```

In [15]:
```
arr = [1, 2, 3]
try:
    print("First item", arr[0])
    print("Second item", arr[3])
except:
    print("Error occured!")
```

```
First item 1
Error occured!
```

In [16]:
```
def func1(a):
    if(a>3):
        b = a/(a-1)
        print("Value of b=", b)
try:
    #func1(int(input('enter num1:')))
    func1(int(input('enter num2:')))
except ZeroDivisionError:
    print('ZeroDivisionError occured here!')
except NameError:
    print('NameError occured here!')
```

```
enter num1:2
enter num2:3
```

In [17]:
```
def fun(a):
    if a < 4:

        # throws ZeroDivisionError for a = 3
        b = a/(a-3)

    # throws NameError if a >= 4
    print("Value of b = ", b)

try:
    fun(3)
    fun(5)
```

```
  File "C:\Users\PRATIK~1\AppData\Local\Temp/ipykernel_5332/3934711978.py", line 12
    fun(5)
        ^
SyntaxError: expected 'except' or 'finally' block
```

In [18]:
```
def AbyB(a , b):
    try:
        c = ((a+b) / (a-b))
    except ZeroDivisionError:
        print ("a/b result in 0")
    else:
        print (c)

# Driver program to test above function
AbyB(2.0, 3.0)
AbyB(3.0, 3.0)
```

```
-5.0
a/b result in 0
```

In [19]:
```
try:
    a = 5/0
    print(a)
except ZeroDivisionError:
    print('Divisor cannot be Zero!')
finally:
    print('This will always be printed!')
```

```
Divisor cannot be Zero!
This will always be printed!
```

In [20]:
```
try:
    a = 5/0
    print(a)
except ZeroDivisionError:
    print('Divisor cannot be Zero!')
finally:
    print('Error occured here')
```

```
Divisor cannot be Zero!
Error occured here
```

In [21]:
```
try:
    raise NameError('Hi there!')
except NameError:
    print('Error occured!')
```

```
Error occured!
```

In [ ]:
```

```