Why Try Except

Can we handle our program without using try and except?

Yes

Why try except is required

ZeroDivisionError

a=int(input('enter first number'))#taking integer input and storing it into a

b=int(input('enter second number')

res=a//b # floor div but this div may cause error if a number is divided by zero

print(res)

#this program may cause zero division error

a=int(input('enter first number'))#taking integer input and storing it into a

b=int(input('enter second number')

c=div(a, b)

If c==-1

print('zero division error')

else:

print(c)

It's working without Try-Except also

But problem arises when we use codes

If a function able to divide it will

If it's not able to divide then it raises the exception

So, there are two possibilities so, here div function is also having two possibilities

```python
def div(a, b):
    if b != 0:
        c = a // b
        return c
    else:
        return -1



a = int(input('Enter first number'))
b = int(input('Enter second number'))

c = div(a, b)
if c == -1:
    print('Zero division error')
else:
    print(c)
```

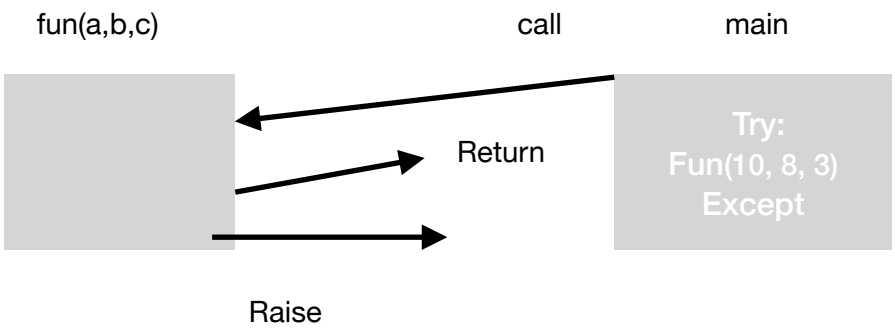
```python
def div(a, b):
    if b != 0:
        c = a // b
        return c
    else:
        raise ZeroDivisionError



a = int(input('Enter first number'))
b = int(input('Enter second number'))

try:
    c = div(a, b)
    print(c)
except:
    print('Zero division error')
```

The output will be same but the internal working program will chang



When you call a function there are two things possible:

Values return or

Exception raised

Main block should try and except

So this is the reason why try and except is introduced

So, when we communicate between two functions it’s difficult to work using if and so, try and except is suitable.