

Python_s_[Exceptions Handling with Try-Except-ELSE-FINALLY]_[Day 26](DATA MINDS)

September 25, 2023

EXCEPTION HANDLING - WE HAVE GENERALLY BLOCKS like “Try” and “Except” IN EXCEPTION HANDLING

1) THE LINE GIVES THE ERROR OR WE THINK IT GIVES THE ERROR IN FUTURE SO WE PUT THAT IN BETWEEN THE TRY AND EXCEPT BLOCK , AND BY USING THE TRY () FUNCTION WE PUT ERROR LINE BELOW THE IN TRY AND CLOSE IT BY USING EXCEPT () FUNCTION WE HANDLE THE EXCEPTIONS AND IN EXCEPT F () FUNCTION WE GIVEN ANY NAME OF ERROR LINE OF CODE AND PRINT IT BY USING PRINT IN PRINT () FUNCTION UNDER THE EXCEPT () FUNCTION

2) THERE ARE GENERALLY FOUR(4) BLOCKS IN EXCEPTION HANDLING -:

(A) TRY () BLOCK,

(B) EXCEPT () BLOCK,

(C) ELSE () BLOCK ,

(D) FINALLY () BLOCK

3) THERE IS NO HEIRACHY ISSUE IN TRY OR EXCEPT FUNCTIONS LIKE WE PUT TRY UNDER THE TRY AND EXCEPT UNDER THE EXCEPT SOMETHING LIKE THAT IN EXCEPTIONS HANDING

```
[1]: a=10
```

```
[2]: a/0
```

```
-----  
ZeroDivisionError                                Traceback (most recent call last)  
Cell In[2], line 1  
----> 1 a/0  
  
ZeroDivisionError: division by zero
```

```
[4]: f=open("test.txt","r")  
     print("This is my Print")
```

```

-----
FileNotFoundError                                Traceback (most recent call last)
Cell In[4], line 1
----> 1 f=open("test.txt","r")
      2 print("This is my Print")

File /opt/conda/lib/python3.10/site-packages/IPython/core/interactiveshell.py:
  282, in _modified_open(file, *args, **kwargs)
    275 if file in {0, 1, 2}:
    276     raise ValueError(
    277         f"IPython won't let you open fd={file} by default "
    278         "as it is likely to crash IPython. If you know what you are_
  doing, "
    279         "you can use builtins' open."
    280     )
--> 282 return io_open(file, *args, **kwargs)

FileNotFoundError: [Errno 2] No such file or directory: 'test.txt'

```

NOTE :-

- 1) IN PYTHON CODE IS EXECUTE LINE BY LINE IF THERE IS ANY ERROR IN THE PREVIOUS LINE SO NEXT LINE WILL AUTOMATICALLY GIVES YOU ERROR
- 2) SO FOR HANDLING SUCH KIND OF CODE WE USE EXCEPTION HANDLING
- 3) IN GENERAL IN SUSPICIOUS CODE OR PRODUCTION GRADE CODE WE USE TO PUT OUR CODE IN BETWEEN TRY AND EXCEPT FOR HANDLING SUCH KIND OF PROBLEMS
- 4) IMPORTANT WE ALWAYS USE LOGGING () FUNCTION AND EXCEPTION HANDLING IN PROJECTS AND CODING TASKS ALWAYS

LOOK AT THE " BELOW EXAMPLE " OF EXCEPTION HANDLING BY USING TRY AND EXCEPT :-

- 1) THE LINE GIVES THE ERROR OR WE THINK IT GIVES THE ERROR IN FUTURE SO WE PUT THAT IN BETWEEN THE TRY AND EXCEPT BLOCK , AND BY USING THE TRY () FUNCTION WE PUT ERROR LINE BELOW THE IN TRY AND CLOSE IT BY USING EXCEPT () FUNCTION WE HANDLE THE EXCEPTIONS AND IN EXCEPT F () FUNCTION WE GIVEN ANY NAME OF ERROR LINE OF CODE AND PRINT IT BY USING PRINT IN PRINT () FUNCTION UNDER THE EXCEPT () FUNCTION

```

[6]: try:
      f=open("test.txt","r")
    except Exception as e:
      print("This is my except block",e)
a=5
b=15
a+b

```

This is my except block [Errno 2] No such file or directory: 'test.txt'

[6]: 20

```
[9]: try:
      f=open("test.txt","w")
      f.write("This is Virat Tiwari")
      f.close()
    except Exception as e:
      print("This is my except block",e)
```

NOTE :-

ELSE () BLOCK IS THAT BLOCK WHICH EXECUTE ONLY WHEN TRY () FUNCTION EXECUTE THEMSELVES SUCCESSFULLY, IF IN ANY CASE TRY () BLOCK IS CRASHED OR IT DOES NOT EXECUTE THEN ELSE () BLOCK IS NOT EXECUTE

```
[10]: # THIS EXAMPLE OF WHERE ELSE ( ) BLOCK SUCCESSFULLY EXECUTED

try:
    f=open("test.txt","w")
    f.write("This is Virat Tiwari")
except Exception as e:
    print("This is my except block",e)
else:
    f.close()
    print("This will be executed once our 'try' will execute without error")
```

This will be executed once our 'try' will execute without error

```
[11]: # THIS EXAMPLE OF WHERE ELSE ( ) BLOCK IS NOT EXECUTED DUE TO CRASHED CODE IN
      ↪ TRY ( ) BLOCK

try:
    f=open("test1.txt","r")
    f.write("This is Virat Tiwari")
except Exception as e:
    print("This is my except block",e)
else:
    f.close()
    print("This will be executed once our 'try' will execute without error")
```

This is my except block [Errno 2] No such file or directory: 'test1.txt'

```
[13]: # IN THIS CODE OF LINE WE GET THE OUTPUT BECAUSE WE WRITE THE FILE AND THERE IS
      ↪ NO USE OF EXCEPT ( ) BLOCK BECAUSE WE JUST WRITE THE FILE WITHOUT READING IT

try :
```

```

f=open("test1.txt","w")
f.write("Aspiring Data Scientist")
finally:
    print("Finally will execute itself in any situation")

```

Finally will execute itself in any situation

[14]: *# IN THIS LINE OF CODE WE DON'T DEFINE THE TXT FILE AND WITHOUT WRITING IT WE
 ↳ READ IT AND ALSO IN THIS CODE OF LINE WE CAN NOT USE EXCEPT () FUNCTION
 ↳ FOR HANDLING THE ERROR THAT'S WHY WE GOT THE ERROR AS A OUTPUT*

```

try :
    f=open("test2.txt","r")
    f.write("Aspiring Data Scientist")
finally:
    print("Finally will execute itself in any situation")

```

Finally will execute itself in any situation

```

-----
FileNotFoundError                                Traceback (most recent call last)
Cell In[14], line 2
      1 try :
----> 2     f=open("test2.txt","r")
      3     f.write("Aspiring Data Scientist")
      4 finally:

File /opt/conda/lib/python3.10/site-packages/IPython/core/interactiveshell.py:
↳282, in _modified_open(file, *args, **kwargs)
    275 if file in {0, 1, 2}:
    276     raise ValueError(
    277         f"IPython won't let you open fd={file} by default "
    278         "as it is likely to crash IPython. If you know what you are_
↳doing, "
    279         "you can use builtins' open."
    280     )
--> 282 return io_open(file, *args, **kwargs)

FileNotFoundError: [Errno 2] No such file or directory: 'test2.txt'

```

THANK YOU SO MUCH :)

YOURS VIRAT TIWARI