Exception Handling

Exception : Its run Time Error

In [ ]:

```
1. SyntaxError
2. NameError
3. ZeroDivisionError
4. AttributeError
5. IndentationError
6. FileNotFoundError
7. FileExistsError
8. ValueError
9. IndexError
10. TypeError

```

In [7]:

```
string="Python Trainning
string
```

```
  Cell In[7], line 1
    string="Python Trainning
           ^
SyntaxError: unterminated string literal (detected at line 1)
```

In [8]:

```
a=10
print(a)
```

```
10
```

In [9]:

```
a=10
print(b)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[9], line 2
      1 a=10
----> 2 print(b)

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

In [10]:

```
x=100
y=2
x/y  #  100/2  >>> Divison by 2 possible
```

Out[10]:

```
50.0
```

In [11]:

```
x=100
y=0
x/y  #  100/0  >>> Divison by zero not possible
```

```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
Cell In[11], line 3
      1 x=100
      2 y=0
----> 3 x/y  #  100/0  >>> Divison by zero not possible

ZeroDivisionError: division by zero
```

In [12]:

```
string='Machine Learning'
string.upper()
```

Out[12]:

```
'MACHINE LEARNING'
```

In [15]:

```
string='Machine Learning'
string.sort()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[15], line 2
      1 string='Machine Learning'
----> 2 string.sort()

AttributeError: 'str' object has no attribute 'sort'
```

In [16]:

```
1 print("Data Science")
2  print("Machine Learning Algorithms")
```

```
  Cell In[16], line 2
    print("Machine Learning Algorithms")
    ^
IndentationError: unexpected indent
```

In [20]:

```
1 if 100==100:
2     print("Hello")
```

```
Hello
```

In [21]:

```
1 if 100==100:
2 print("Hello")
```

```
  Cell In[21], line 2
    print("Hello")
    ^
IndentationError: expected an indented block after 'if' statement on line 1
```

In [22]:

```
import os
file=open('demo.txt','r')
data=file.read()
print(data)
file.close()
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
Cell In[22], line 2
      1 import os
----> 2 file=open('demo.txt','r')
      3 data=file.read()
      4 print(data)

File ~\AppData\Local\Programs\Python\Python311\Lib\site-packages\IPython\core
\interactiveshell.py:284, in _modified_open(file, *args, **kwargs)
    277 if file in {0, 1, 2}:
    278     raise ValueError(
    279         f"IPython won't let you open fd={file} by default "
    280         "as it is likely to crash IPython. If you know what you are do
ing, "
    281         "you can use builtins' open."
    282     )
--> 284 return io_open(file, *args, **kwargs)

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

In [23]:

```
with open('test.txt','w') as f :
    f.write("Python and Machine Learning")
```

In [24]:

```
with open('test2.txt','x') as f :
    f.write("Shri Software")
```

In [25]:

```
1 with open('test2.txt','x') as f :
2     f.write("Shri Software")
```

```
---------------------------------------------------------------------------
FileExistsError                           Traceback (most recent call last)
Cell In[25], line 1
----> 1 with open('test2.txt','x') as f :
      2     f.write("Shri Software")

File ~\AppData\Local\Programs\Python\Python311\Lib\site-packages\IPython\core
\interactiveshell.py:284, in _modified_open(file, *args, **kwargs)
    277 if file in {0, 1, 2}:
    278     raise ValueError(
    279         f"IPython won't let you open fd={file} by default "
    280         "as it is likely to crash IPython. If you know what you are do
ing, "
    281         "you can use builtins' open."
    282     )
--> 284 return io_open(file, *args, **kwargs)

FileExistsError: [Errno 17] File exists: 'test2.txt'
```

In [26]:

```
1 price=569.67
2 int(price)
```

Out[26]:

```
569
```

In [27]:

```
1 price='569'
2 int(price)
```

Out[27]:

```
569
```

In [28]:

```
1 price='569.67'
2 int(price)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[28], line 2
      1 price='569.67'
----> 2 int(price)

ValueError: invalid literal for int() with base 10: '569.67'
```

In [31]:

```
list1=[10,60,50,30]
print(list1[0])
print(list1[3])
print(list1[4])
```

```
10
30

---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[31], line 4
      2 print(list1[0])
      3 print(list1[3])
----> 4 print(list1[4])

IndexError: list index out of range
```

In [32]:

```
m=200
n='python'
m+n
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[32], line 3
      1 m=200
      2 n='python'
----> 3 m+n

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

In [35]:

```
m='200'
n='python'
m+n
```

Out[35]:

```
'200python'
```

In [34]:

```
m=10
n='python'
m*n
```

Out[34]:

```
'pythonpythonpythonpythonpythonpythonpythonpythonpythonpythonpython'
```

try - except

In [36]:

```
# Syntax :
try :
    print("Try Block")
except :
    print("Except Block")
```

```
Try Blcok
```

In [37]:

```
print("Python")
print("Machine Learning")
```

```
Python
Machine Learning
```

In [38]:

```
print("Python")
z=100/0
print("Machine Learning")
```

```
Python

---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
Cell In[38], line 2
      1 print("Python")
----> 2 z=100/0
      3 print("Machine Learning")

ZeroDivisionError: division by zero
```

In [39]:

```
print("Python")

try:
    z=100/0
except:
    print("Divison by zero not possible")

print("Machine Learning")
```

```
Python
Divison by zero not possible
Machine Learning
```

In [40]:

```
print("Python")

try:
    x=int(input("Enter 1st Integer Number : "))
    y=int(input("Enter 2nd Integer Number : "))
    z=x/y
    print(f"Division of {x} and {y} is {z}")
except:
    print("Divison by zero not possible")

print("Machine Learning")
```

```
Python
Enter 1st Integer Number : 20
Enter 2nd Integer Number : 3
Division of 20 and 3 is 6.666666666666667
Machine Learning
```

In [41]:

```
print("Python")

try:
    x=int(input("Enter 1st Integer Number : "))
    y=int(input("Enter 2nd Integer Number : "))
    z=x/y
    print(f"Division of {x} and {y} is {z}")
except:
    print("Divison by zero not possible")

print("Machine Learning")
```

```
Python
Enter 1st Integer Number : 80
Enter 2nd Integer Number : 0
Divison by zero not possible
Machine Learning
```

In [42]:

```
# Without Try Except
print("Python")


x=int(input("Enter 1st Integer Number : "))
y=int(input("Enter 2nd Integer Number : "))
z=x/y
print(f"Division of {x} and {y} is {z}")


print("Machine Learning")
```

```
Python
Enter 1st Integer Number : 20
Enter 2nd Integer Number : 4
Division of 20 and 4 is 5.0
Machine Learning
```

In [43]:

```
# Without Try Except
print("Python")


x=int(input("Enter 1st Integer Number : "))
y=int(input("Enter 2nd Integer Number : "))
z=x/y
print(f"Division of {x} and {y} is {z}")


print("Machine Learning")
```

```
Python
Enter 1st Integer Number : 56
Enter 2nd Integer Number : 0

---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
Cell In[43], line 7
      5 x=int(input("Enter 1st Integer Number : "))
      6 y=int(input("Enter 2nd Integer Number : "))
----> 7 z=x/y
      8 print(f"Division of {x} and {y} is {z}")
     11 print("Machine Learning")

ZeroDivisionError: division by zero
```

In [44]:

```
print("Tuple Program")
t1=(30,50,70,90)
t1.append(100)
print("Tuple Program Completed")
```

```
Tuple Program

---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[44], line 3
      1 print("Tuple Program")
      2 t1=(30,50,70,90)
----> 3 t1.append(100)
      4 print("Tuple Program Completed")

AttributeError: 'tuple' object has no attribute 'append'
```

In [45]:

```
print("Tuple Program")
try :
    t1=(30,50,70,90)
    t1.append(100)
except :
    print("append function is not in tuple ")

print("Tuple Program Completed")
```

```
Tuple Program
append function is not in tuple
Tuple Program Completed
```

In [46]:

```
print("List Program")
try :
    l1=[30,50,70,90]
    print(l1[10])
except :
    print("List out of index Error.... !!!!  ")

print("List Program Completed")
```

```
List Program
List out of index Error.... !!!!
List Program Completed
```

In [47]:

```
print("List Program")
try :
    l1=[30,50,70,90]
    print(l1[1])
except :
    print("List out of index Error.... !!!!  ")

print("List Program Completed")
```

```
List Program
50
List Program Completed
```

In [49]:

```
print("File Program")
with open('test2.txt','x') as f :
    f.write("Data Science")

a=10
b=20
c=a+b
print(f"Addition of {a} and {b} is {c} ")
```

```
File Program

---------------------------------------------------------------------------
FileExistsError                           Traceback (most recent call last)
Cell In[49], line 2
      1 print("File Program")
----> 2 with open('test2.txt','x') as f :
      3     f.write("Data Science")
      5 a=10

File ~\AppData\Local\Programs\Python\Python311\Lib\site-packages\IPython\core
\interactiveshell.py:284, in _modified_open(file, *args, **kwargs)
    277 if file in {0, 1, 2}:
    278     raise ValueError(
    279         f"IPython won't let you open fd={file} by default "
    280         "as it is likely to crash IPython. If you know what you are do
ing, "
    281         "you can use builtins' open."
    282     )
--> 284 return io_open(file, *args, **kwargs)

FileExistsError: [Errno 17] File exists: 'test2.txt'
```

In [50]:

```
print("File Program Opening file in x mode ")
try:
    with open('test2.txt','x') as f :
        f.write("Data Science")
except:
    print("File Already Exist Error ")

a=10
b=20
c=a+b
print(f"Addition of {a} and {b} is {c} ")
```

```
File Program
File Already Exist Error
Addition of 10 and 20 is 30
```

In [51]:

```
f_name="House_Rent_Dataset.csv"
import pandas as pd
df=pd.read_excel(f_name)
df
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[51], line 3
      1 f_name="House_Rent_Dataset.csv"
      2 import pandas as pd
----> 3 df=pd.read_excel(f_name)
      4 df

File ~\AppData\Local\Programs\Python\Python311\Lib\site-packages\pandas\io\exc
el\_base.py:478, in read_excel(io, sheet_name, header, names, index_col, useco
ls, dtype, engine, converters, true_values, false_values, skiprows, nrows, na_
values, keep_default_na, na_filter, verbose, parse_dates, date_parser, date_fo
rmat, thousands, decimal, comment, skipfooter, storage_options, dtype_backend)
    476 if not isinstance(io, ExcelFile):
    477     should_close = True
--> 478     io = ExcelFile(io, storage_options=storage_options, engine=engine)
    479 elif engine and engine != io.engine:
    480     raise ValueError(
    481         "Engine should not be specified when passing "
    482         "an ExcelFile - ExcelFile already has the engine set"
    483     )

File ~\AppData\Local\Programs\Python\Python311\Lib\site-packages\pandas\io\exc
el\_base.py:1500, in ExcelFile.__init__(self, path_or_buffer, engine, storage_
options)
   1496     ext = inspect_excel_format(
   1497         content_or_path=path_or_buffer, storage_options=storage_option
s
   1498     )
   1499     if ext is None:
-> 1500         raise ValueError(
   1501             "Excel file format cannot be determined, you must specify
"
   1502             "an engine manually."
   1503         )
   1505 engine = config.get_option(f"io.excel.{ext}.reader", silent=True)
   1506 if engine == "auto":

ValueError: Excel file format cannot be determined, you must specify an engine
manually.
```

In [52]:

```
f_name="House_Rent_Dataset.csv"
import pandas as pd
try :
    df=pd.read_excel(f_name)
except :
    df=pd.read_csv(f_name)

df
```

Out[52]:

	Posted On	BHK	Rent	Size	Floor	Area Type	Area Locality	City	Furnishing Status	Te Prefe
0	2022-05-18	2	10000	1100	Ground out of 2	Super Area	Bandel	Kolkata	Unfurnished	Bachelors/Fa
1	2022-05-13	2	20000	800	1 out of 3	Super Area	Phool Bagan, Kankurgachi	Kolkata	Semi-Furnished	Bachelors/Fa
2	2022-05-16	2	17000	1000	1 out of 3	Super Area	Salt Lake City Sector 2	Kolkata	Semi-Furnished	Bachelors/Fa
3	2022-07-04	2	10000	800	1 out of 2	Super Area	Dumdum Park	Kolkata	Unfurnished	Bachelors/Fa
4	2022-05-09	2	7500	850	1 out of 2	Carpet Area	South Dum Dum	Kolkata	Unfurnished	Bach
...	...	...	...	...	...	...	...	...	...	
4741	2022-05-18	2	15000	1000	3 out of 5	Carpet Area	Bandam Kommu	Hyderabad	Semi-Furnished	Bachelors/Fa
4742	2022-05-15	3	29000	2000	1 out of 4	Super Area	Manikonda, Hyderabad	Hyderabad	Semi-Furnished	Bachelors/Fa
4743	2022-07-10	3	35000	1750	3 out of 5	Carpet Area	Himayath Nagar, NH 7	Hyderabad	Semi-Furnished	Bachelors/Fa
4744	2022-07-06	3	45000	1500	23 out of 34	Carpet Area	Gachibowli	Hyderabad	Semi-Furnished	Fa
4745	2022-05-04	2	15000	1000	4 out of 5	Carpet Area	Suchitra Circle	Hyderabad	Unfurnished	Bach

4746 rows × 12 columns

In [53]:

```
print("Opening File in 'x' Mode ")
try :
    print("TRY Block")
    f=open('demo.txt', 'x')
    f.write("Samyak is Python Developer")
    f.close()

    f2=open('test3.txt','r')
    data=f2.read()
    print(data)
    f2.close()

except FileExistsError as error :
    print(error)

except FileNotFoundError as error :
    print(error)

except :
    print("Except Block : Custumized error ")


a=50
b=90
c=a+b
print(f"Addition of {a} and {b} is {c} ")
```

```
Opening File in 'x' Mode
TRY Block
[Errno 2] No such file or directory: 'test3.txt'
Addition of 50 and 90 is 140
```

In [54]:

```
print("Opening File in 'x' Mode ")
try :
    print("TRY Block")
    f=open('demo.txt', 'x')
    f.write("Samyak is Python Developer")
    f.close()

    f2=open('test3.txt','r')
    data=f2.read()
    print(data)
    f2.close()

except FileExistsError as error :
    print(error)

except FileNotFoundError as error :
    print(error)

except :
    print("Except Block : Custumized error ")


a=50
b=90
c=a+b
print(f"Addition of {a} and {b} is {c} ")
```

```
Opening File in 'x' Mode
TRY Block
[Errno 17] File exists: 'demo.txt'
Addition of 50 and 90 is 140
```

In [55]:

```
print("Opening File in 'x' Mode ")
try :
    print("TRY Block")
    f=open('demo123.txt', 'x')
    f.write("Samyak is Python Developer")
    f.close()

    f2=open('test.txt','r')
    data=f2.read()
    print(data)
    f2.close()

    div=400/0
    print(div)

except FileExistsError as error :
    print(error)

except FileNotFoundError as error :
    print(error)

except :
    print("Except Block : Custumized error ")


a=50
b=90
c=a+b
print(f"Addition of {a} and {b} is {c} ")
```

```
Opening File in 'x' Mode
TRY Block
Python and Machine Learning
Except Block : Custumized error
Addition of 50 and 90 is 140
```

try - except - else

In [56]:

```
try :
    print("try blcok ")

except :
    print("except block")

else:
    print("else block")

```

```
try blcok
else block
```

In [57]:

```
try :
    z=80/0

except :
    print("except block")

else:
    print("else block")
```

```
except block
```

In [60]:

```
string="Python"

try:
    print(string[2])

except IndexError as error :
    print(error)

else :
    print("Original String : ", string)

print("String Program Completed ")
```

```
t
Original String :  Python
String Program Completed
```

In [61]:

```
string="Python"

try:
    print(string[10])

except IndexError as error :
    print(error)

else :
    print("Original String : ", string)

print("String Program Completed ")
```

```
string index out of range
String Program Completed
```

try - except - finally

Finally Block will always excecute

In [63]:

```
try :
    print("try blcok ")

except :
    print("except block")

finally:
    print("Finally block")

print("Program Completed")
```

```
try blcok
Finally block
Program Completed
```

In [64]:

```
try :
    print("try blcok ")
    div=70/0

except :
    print("except block")

finally:
    print("Finally block")

print("Program Completed")
```

```
try blcok
except block
Finally block
Program Completed
```

raise

In [65]:

```
x=10
y=20
if type(x)== int and type(y)==int:
    add=x+y
    print(f"Addition of {x} and {y} is {add}")


```

```
Addition of 10 and 20 is 30
```

In [70]:

```

```

In [66]:

```
1 10+20
```

Out[66]:

```
30
```

In [67]:

```
1 10+'20'
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[67], line 1
----> 1 10+'20'

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

In [71]:

```
1 x=10
2 y='20'
3 if type(x)== int and type(y)==int:
4     add=x+y
5     print(f"Addition of {x} and {y} is {add}")
6 else:
7     raise TypeError("Error : x is not int or y is not int or both x and y are not int  "
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[71], line 7
      5     print(f"Addition of {x} and {y} is {add}")
      6 else:
----> 7     raise TypeError("Error : x is not int or y is not int or both x an
d y are not int  ")

TypeError: Error : x is not int or y is not int or both x and y are not int
```

In [72]:

```
x='10'
y='20'
if type(x)== int and type(y)==int:
    add=x+y
    print(f"Addition of {x} and {y} is {add}")
else:
    raise TypeError("Error : x is not int or y is not int or both x and y are not int  "
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[72], line 7
      5     print(f"Addition of {x} and {y} is {add}")
      6 else:
----> 7     raise TypeError("Error : x is not int or y is not int or both x an
d y are not int  ")

TypeError: Error : x is not int or y is not int or both x and y are not int
```

In [73]:

```
x=10
y=20
if type(x)== int and type(y)==int:
    add=x+y
    print(f"Addition of {x} and {y} is {add}")
else:
    raise TypeError("Error : x is not int or y is not int or both x and y are not int  "
```

```
Addition of 10 and 20 is 30
```

Nested try - except

In [74]:

```
try :
    print("Outer try block ")
    try :
        print("inner try block")
    except :
        print("Inner Except")
except:
    print("Outer Except")

```

```
Outer try block
inner try block
```

In [75]:

```
try :
    100/0
    try :
        print("inner try block")
    except :
        print("Inner Except")
except:
    print("Outer Except")

```

```
Outer Except
```

In [76]:

```
try :
    print("Outer try block ")
    try :
        100/0
    except :
        print("Inner Except")
except:
    print("Outer Except")

```

```
Outer try block
Inner Except
```

In [77]:

```
try :
    print("Outer Try")
    print("Opening file in x mode")
    f=open("Demo555.txt",'x')
    f.write("Vinod is Data Analyst")
    f.close()

    try :
        print("Inner Try1 ")
        f2=open('test555.txt','r')
        data=f2.read()
        print(data)
        f2.close()

    except FileNotFoundError as error:
        print(error)


    try :
        print("Inner Try2 ")
        div=100/0
        print(div)
    except ZeroDivisionError as error:
        print(error)


except FileExistsError as error:
    print(error)

finally:
    print("This code always excecute .....")
    a=10
    b=30
    print(f"addition of {a} and {b} is {a+b}")
```

```
Outer Try
Opening file in x mode
Inner Try1
[Errno 2] No such file or directory: 'test555.txt'
Inner Try2
division by zero
This code always excecute .....
addition of 10 and 30 is 40
```

In [78]:

```
try :
    print("Outer Try")
    print("Opening file in x mode")
    f=open("Demo555.txt",'x')
    f.write("Vinod is Data Analyst")
    f.close()

    try :
        print("Inner Try1 ")
        f2=open('test555.txt','r')
        data=f2.read()
        print(data)
        f2.close()

    except FileNotFoundError as error:
        print(error)


    try :
        print("Inner Try2 ")
        div=100/0
        print(div)
    except ZeroDivisionError as error:
        print(error)


except FileExistsError as error:
    print(error)

finally:
    print("This code always excecute .....")
    a=10
    b=30
    print(f"addition of {a} and {b} is {a+b}")


```

```
Outer Try
Opening file in x mode
[Errno 17] File exists: 'Demo555.txt'
This code always excecute .....
addition of 10 and 30 is 40
```

In [ ]:

```

```