There are 2 stages where error may happen in a program

- During compilation -> Syntax Error
- During execution -> Exceptions

Syntax Error

- Something in the program is not written according to the program grammar.
- Error is raised by the interpreter/compiler
- You can solve it by rectifying the program

```
# Example of syntax error
print 'hello world'
```

```
  Cell In[8], line 2
    print 'hello world'
    ^
SyntaxError: Missing parentheses in call to 'print'. Did you mean
print(...)?
```

Other examples of syntax error

- Leaving symbols like colon,brackets
- Misspelling a keyword
- Incorrect indentation
- empty if/else/loops/class/functions

```
a=5
if a==3
    print(a)
```

```
  Cell In[10], line 2
    if a==3
           ^
SyntaxError: expected ':'
```

```
a=5
iff a==3:
    print(a)
```

```
  Cell In[12], line 2
    iff a==3:
        ^
SyntaxError: invalid syntax
```

```
a=5
if a==3:
print(a)
```

```
  Cell In[14], line 3
    print(a)
    ^
IndentationError: expected an indented block after 'if' statement on
line 2


# Index Error
# IndexError
# The IndexError is thrown when trying to access an item at an invalid
index.
L = [1,2,3]
L[100]

---------------------------------------------------------------------
-----
IndexError                                Traceback (most recent call
last)
Cell In[16], line 5
      1 # Index Error
      2 # IndexError
      3 # The IndexError is thrown when trying to access an item at an
invalid index.
      4 L = [1,2,3]
----> 5 L[100]

IndexError: list index out of range

# ModuleNotFoundError
# The ModuleNotFoundError is thrown when a module could not be found.
import mathi
math.floor(5.3)

---------------------------------------------------------------------
-----
ModuleNotFoundError                       Traceback (most recent call
last)
Cell In[18], line 3
      1 # ModuleNotFoundError
      2 # The ModuleNotFoundError is thrown when a module could not be
found.
----> 3 import mathi
      4 math.floor(5.3)

ModuleNotFoundError: No module named 'mathi'

# KeyError
# The KeyError is thrown when a key is not found

d = {'name':'nitish'}
d['age']
```

```
---------------------------------------------------------------------
-----
KeyError                                  Traceback (most recent call
last)
Cell In[20], line 5
      1 # KeyError
      2 # The KeyError is thrown when a key is not found
      4 d = {'name':'nitish'}
----> 5 d['age']

KeyError: 'age'
```

```
# TypeError
# The TypeError is thrown when an operation or function is applied to
an object of an inappropriate type.
1 + 'a'
```

```
---------------------------------------------------------------------
-----
TypeError                                 Traceback (most recent call
last)
Cell In[22], line 3
      1 # TypeError
      2 # The TypeError is thrown when an operation or function is
applied to an object of an inappropriate type.
----> 3 1 + 'a'

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

```
# ValueError
# The ValueError is thrown when a function's argument is of an
inappropriate type.
int('a')
```

```
---------------------------------------------------------------------
-----
ValueError                                Traceback (most recent call
last)
Cell In[24], line 3
      1 # ValueError
      2 # The ValueError is thrown when a function's argument is of an
inappropriate type.
----> 3 int('a')

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

```
# NameError
# The NameError is thrown when an object could not be found.
print(k)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[26], line 3
      1 # NameError
      2 # The NameError is thrown when an object could not be found.
----> 3 print(k)

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

```
# AttributeError
L = [1,2,3]
L.upper()

# Stacktrace
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[28], line 3
      1 # AttributeError
      2 L = [1,2,3]
----> 3 L.upper()
      5 # Stacktrace

AttributeError: 'list' object has no attribute 'upper'
```

Exceptions

If things go wrong during the execution of the program(runtime). It generally happens when something unforeseen has happened.

- Exceptions are raised by python runtime
- You have to takle is on the fly

Examples

- Memory overflow
- Divide by 0 -> logical error
- Database error

```
# Why is it important to handle exceptions
# how to handle exceptions
# using-> Try except block

# ye jo error aata hai usse Stacktrace bolte hain  --> stacktrace me sabse pahle bataya jata hai ki kno se type ka error hai and kya error hai wo
# bataya jata and kon se line me error hai ye bhi batata hai
```

```python
# error ke handle karna jaruri hai qki iss tarah ka error user ko nahi dikhna chahaiye nahi to usse presani ho sakti hai and second is for security
# purpose iss tarah ke error me saaf dikhta hai ki kon se line pe kon sa code likha hua hai



# let's create a file
with open('sample.txt','w') as f:
  f.write('hello world')

# try catct demo
with open('sample.txt','r') as f:
  print(f.read())
```

```
hello world
```

```python
# jab bhi aap kuch external chizo ko aone code me laa rahe ho (like koi file, database connection, bluetooth connection etc) to aap exception
# hendle karte ho

# suppose ki koi file ka name koi sample se change karke sample1 kar dya
with open('sample1.txt','r') as f:
  print(f.read())
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
Cell In[42], line 2
      1 # suppose ki koi file ka name koi sample se change karke sample1 kar dya
----> 2 with open('sample1.txt','r') as f:
      3   print(f.read())

File ~\AppData\Roaming\Python\Python312\site-packages\IPython\core\interactiveshell.py:324, in _modified_open(file, *args, **kwargs)
    317 if file in {0, 1, 2}:
    318     raise ValueError(
    319         f"IPython won't let you open fd={file} by default "
    320         "as it is likely to crash IPython. If you know what you are doing, "
    321         "you can use builtins' open."
    322     )
--> 324 return io_open(file, *args, **kwargs)
```

```
FileNotFoundError: [Errno 2] No such file or directory: 'sample1.txt'
```

```
# ye ajib saa error aayega to aaisi error user ko nn dikhee to hum
exception ko handle karte hain
try:
    with open('sample.txt','r') as f:
      print(f.read())
except:
    print('sorry file not found')
```

```
hello world
```

```
try:
    with open('sample1.txt','r') as f:
      print(f.read())
except:
    print('sorry file not found')
```

```
sorry file not found
```

```
# abhi aap dekhoge ki koi error nahi aaya but ek pyara sa msg aaya hai
```

```
# catching specific exception
try :
    f=open('sample1.txt','r')
    print(f.read())
    print(m)

except:
    print('some error occured')
```

```
some error occured
```

```
# try ke andar gya and first line pe hi error to except ko print kara
dya
try :
    f=open('sample.txt','r')
    print(f.read())
    print(m)

except:
    print('some error occured')
```

```
hello world
some error occured
```

```
# abhi try ke andar gya and 3rd line pe error qki koi m name ka
variable hai hi nahi to 1st 2nd line to execute ho jayega but
# 3rd line ke karan except ka code bhi run hoga
```

```python
# ek baat socho as a user ki kya kabhi aap chahoge ki kuch bhi error
aaye to aap humesa same message print kara rahe ho ofcourse not qki
aapko samajh
# me bhi aana chahaiye ki kya problem aa rahi hai taki proper
understanding bane(matlab ki hume nature of error pata chale)


try :
    f=open('sample.txt','r')
    print(f.read())
    print(m)

except Exception as e:
    print(e)

# abb kiss tarah ka error aa raha hai kuch kuch pata chal raha hai
```

```
hello world
name 'm' is not defined
```

```python
try :
    f=open('sample1.txt','r')
    print(f.read())
    print(m)

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

```
[Errno 2] No such file or directory: 'sample1.txt'
```

```python
try :
    f=open('sample1.txt','r')
    print(f.read())
    print(m)

except Exception as e:
    print(e.with_trackback)

# isse aap proper error name type check kar sakte ho
```

```
---------------------------------------------------------------------
-----
FileNotFoundError                         Traceback (most recent call
last)
Cell In[64], line 2
      1 try :
----> 2     f=open('sample1.txt','r')
      3     print(f.read())

File ~\AppData\Roaming\Python\Python312\site-packages\IPython\core\
interactiveshell.py:324, in _modified_open(file, *args, **kwargs)
```

```
    318     raise ValueError(
    319         f"IPython won't let you open fd={file} by default "
    320         "as it is likely to crash IPython. If you know what
you are doing, "
    321         "you can use builtins' open."
    322     )
--> 324 return io_open(file, *args, **kwargs)

FileNotFoundError: [Errno 2] No such file or directory: 'sample1.txt'

During handling of the above exception, another exception occurred:

AttributeError                            Traceback (most recent call
last)
Cell In[64], line 7
      4     print(m)
      6 except Exception as e:
----> 7     print(e.with_trackback)
      9 # isse aap proper error name type check kar sakte ho

AttributeError: 'FileNotFoundError' object has no attribute
'with_trackback'
```

```
try :
    f=open('sample.txt','r')
    print(f.read())
    print(m)

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

```
hello world

---------------------------------------------------------------------------
-----
NameError                                 Traceback (most recent call
last)
Cell In[66], line 4
      3     print(f.read())
----> 4     print(m)
      6 except Exception as e:

NameError: name 'm' is not defined

During handling of the above exception, another exception occurred:

AttributeError                            Traceback (most recent call
last)
Cell In[66], line 7
      4     print(m)
      6 except Exception as e:
```

```
----> 7     print(e.with_trackback)

AttributeError: 'NameError' object has no attribute 'with_trackback'
```

```
try :
    f=open('sample1.txt','r')
    print(f.read())
    print(m)
except FileNotFoundError:
    print('file not found')
except NameError:
    print('variable not defined')
```

```
file not found
```

```
try :
    f=open('sample.txt','r')
    print(f.read())
    print(m)
except FileNotFoundError:
    print('file not found')
except NameError:
    print('variable not defined')
```

```
hello world
variable not defined
```

```
try :
    m=5
    f=open('sample.txt','r')
    print(f.read())
    print(m)
    print(5/0)
except FileNotFoundError:
    print('file not found')
except NameError:
    print('variable not defined')
except Exception as e:
    print(e.with_trackback)
```

```
hello world
5

---------------------------------------------------------------------------
-----
ZeroDivisionError                         Traceback (most recent call
last)
Cell In[72], line 6
      5     print(m)
----> 6     print(5/0)
      7 except FileNotFoundError:
```

```
ZeroDivisionError: division by zero

During handling of the above exception, another exception occurred:

AttributeError                            Traceback (most recent call
last)
Cell In[72], line 12
     10     print('variable not defined')
     11 except Exception as e:
---> 12     print(e.with_trackback)

AttributeError: 'ZeroDivisionError' object has no attribute
'with_trackback'
```

```
try :
    m=5
    f=open('sample.txt','r')
    print(f.read())
    print(m)
    print(5/0)
except FileNotFoundError:
    print('file not found')
except NameError:
    print('variable not defined')
except ZeroDivisionError:
    print("can't devide by zero")
```

```
hello world
5
can't devide by zero
```

```
# aap ek  general except block bhi banate ho suppose agar kuch samajh
nahi aa rha hai to wo print kara de(ye ek general line hai jo jitna
hum
# banaye hai exception handler usse match nahi ho raha hai to eo last
bale me jo hai usse execute kara dega)
try :
    m=5
    f=open('sample.txt','r')
    print(f.read())
    print(m)
    print(5/2)
    L = [1,2,3]
    L[100]
except FileNotFoundError:
    print('file not found')
except NameError:
    print('variable not defined')
```

```
except ZeroDivisionError:
    print("can't devide by zero")

except Exception as e:
    print(e)

# generic message humesa last me likhna chahaiye otherwise ye overtake
kar lega
```

```
hello world
5
2.5
list index out of range
```

```
# else
try:
    f=open('sample.txt','r')
except FileNotFoundError:
    print('file nahi mili')
except Exception:
    print('Kuch to lafda hai')

else:
    print(f.read())


# else me wo code likha jata hai jiske bare me aap super sure ho ki ye
code fatne bala nahi hai and else tabhi execute hoga jabki koi bhi
except block
# execute nahi hoga
```

```
hello world
```

```
try:
    f=open('sample1.txt','r')
except FileNotFoundError:
    print('file nahi mili')
except Exception:
    print('Kuch to lafda hai')

else:
    print(f.read())
```

```
file nahi mili
```

```
# finally  -> kuch bhi ho finally to execute hoga hi

try:
  f = open('sample.txt','r')
except FileNotFoundError:
  print('file nai mili')
```

```
except Exception:
  print('kuch to lafda hai')
else:
  print(f.read())
finally:
  print('ye to print hoga hi')
```

```
hello world
ye to print hoga hi
```

```
try:
  f = open('sample1.txt','r')
except FileNotFoundError:
  print('file nai mili')
except Exception:
  print('kuch to lafda hai')
else:
  print(f.read())
finally:
  print('ye to print hoga hi')
```

```
file nai mili
ye to print hoga hi
```

```
# geenerally finally ke andar wo kaam karte hai jo ki hume har haal me
karna hi hai chahae exception aaye ya phir nahi aaye
# like databse connectivity ko band karna , socket connection ko band
karna hai etc.

# raise Exception
# In Python programming, exceptions are raised when errors occur at
runtime.
# We can also manually raise exceptions using the raise keyword.

# We can optionally pass values to the exception to clarify why that
exception was raised

raise NameError
```

```
---------------------------------------------------------------------
-----
NameError                                 Traceback (most recent call
last)
Cell In[96], line 1
----> 1 raise NameError

NameError:
```

```
# hum manullay ek error kahi pe ,kisi bhi time pe   throw kara sakte
hain
raise NameError('aaise hi try kar raha hu')
```

```
---------------------------------------------------------------------
-----
NameError                                 Traceback (most recent call
last)
Cell In[98], line 2
      1 # hum manullay ek error kisi bhi time pe throw kara sakte hain
----> 2 raise NameError('aaise hi try kar raha hu')

NameError: aaise hi try kar raha hu
```

```
raise abc('aaise hi try kar raha hu')
# abc koi proper error type nahi hai so ye kaam nahi karega
```

```
---------------------------------------------------------------------
-----
NameError                                 Traceback (most recent call
last)
Cell In[100], line 1
----> 1 raise abc('aaise hi try kar raha hu')

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

```
raise FileNotFoundError('aaise hi try kar raha hu')
```

```
---------------------------------------------------------------------
-----
FileNotFoundError                         Traceback (most recent call
last)
Cell In[102], line 1
----> 1 raise FileNotFoundError('aaise hi try kar raha hu')

FileNotFoundError: aaise hi try kar raha hu
```

```
# kya fayda --->   raise ek error throw karta hai and except usse
handle karta hai

# python   c++
#  try   -> try
# except --> catch
# raise --> throw

class Bank:

  def __init__(self,balance):
    self.balance = balance

  def withdraw(self,amount):
    if amount < 0:
      raise Exception('amount cannot be -ve')
    if self.balance < amount:
      raise Exception('paise nai hai tere paas')
```

```
    self.balance = self.balance - amount

obj = Bank(10000)
try:
  obj.withdraw(5000)
except Exception as e:
  print(e)
else:
  print(obj.balance)

# jitne bhi error occur hote hain wo sab ek class hota hai and class
ke andar se jo raise exception create karega wo ek object hoga jo ki
class ke baha
# except ko milega
```

```
5000
```

```
class Bank:

  def __init__(self,balance):
    self.balance = balance

  def withdraw(self,amount):
    if amount < 0:
      raise Exception('amount cannot be -ve')
    if self.balance < amount:
      raise Exception('paise nai hai tere paas')
    self.balance = self.balance - amount

obj = Bank(10000)
try:
  obj.withdraw(-5000)
except Exception as e:
  print(e)
else:
  print(obj.balance)
```

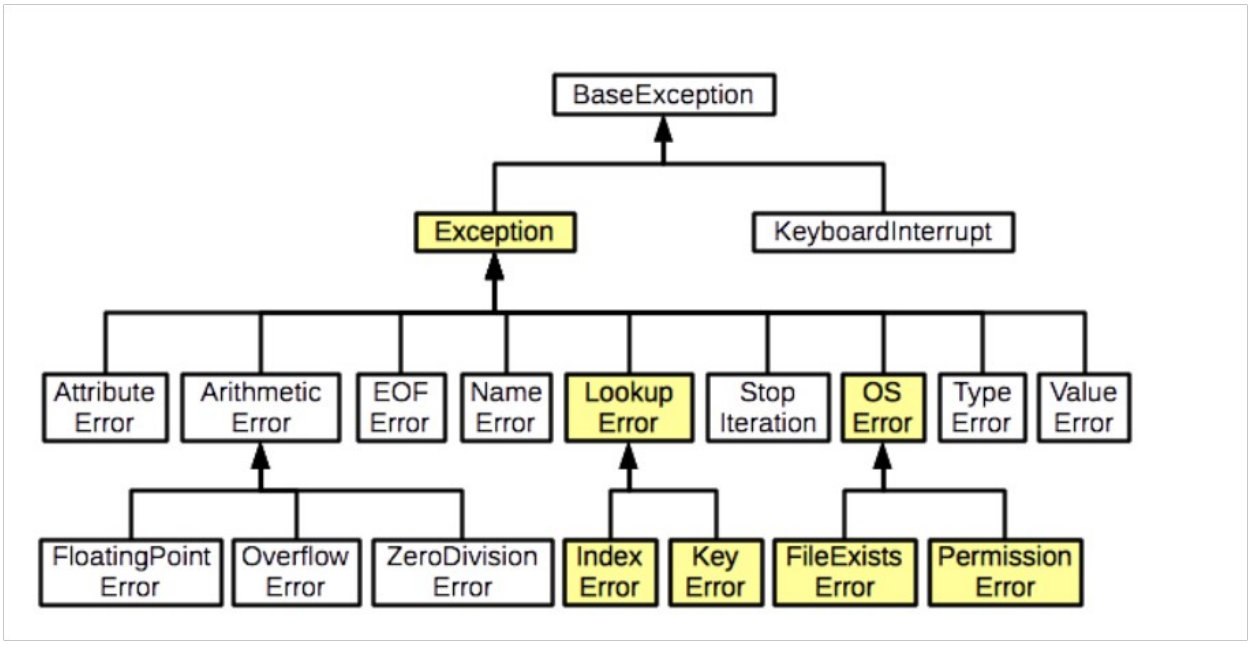
```
amount cannot be -ve
```

```
class Bank:

  def __init__(self,balance):
    self.balance = balance

  def withdraw(self,amount):
    if amount < 0:
      raise Exception('amount cannot be -ve')
    if self.balance < amount:
      raise Exception('paise nai hai tere paas')
    self.balance = self.balance - amount
```

```
obj = Bank(10000)
try:
  obj.withdraw(15000)
except Exception as e:
  print(e)
else:
  print(obj.balance)
```

```
paise nai hai tere paas
```

```
# creating custom exceptions  (abhi tak sare exception wo aa rahe the jo ki predefined hai aap custom bhi create kar sakte ho)
# search on google (exception hierarchy in python) and go on image section
```




```
# aap apna custom exception class create kar sakte ho and usse use kar sakte ho\
# jab aap apna exception class create karte ho to usse main exception class se inherit karna hota hai
class MyException(Exception):
    def __init__(self,message):
        print(message)



class Bank:

  def __init__(self,balance):
    self.balance = balance
```

```
  def withdraw(self,amount):
    if amount < 0:
      raise MyException('amount cannot be -ve')
    if self.balance < amount:
      raise MyException('paise nai hai tere paas')
    self.balance = self.balance - amount

obj = Bank(10000)
try:
  obj.withdraw(-5000)
except MyException as e:
  pass   # qki already humne MyException class me print kara dya hai
else:
  print(obj.balance)
```

```
amount cannot be -ve
```

```
# kya fayda ye karne se

class SecurityError(Exception):

  def __init__(self,message):
    print(message)

  def logout(self):
    print('logout')

class Google:

  def __init__(self,name,email,password,device):
    self.name = name
    self.email = email
    self.password = password
    self.device = device

  def login(self,email,password,device):
    if device != self.device:
      raise SecurityError('bhai teri to lag gayi')
    if email == self.email and password == self.password:
      print('welcome')
    else:
      print('login error')



obj = Google('nitish','nitish@gmail.com','1234','android')

try:
  obj.login('nitish@gmail.com','1234','windows')
except SecurityError as e:
```

```
  e.logout()
else:
  print(obj.name)
finally:
  print('database connection closed')


bhai teri to lag gayi
logout
database connection closed
```