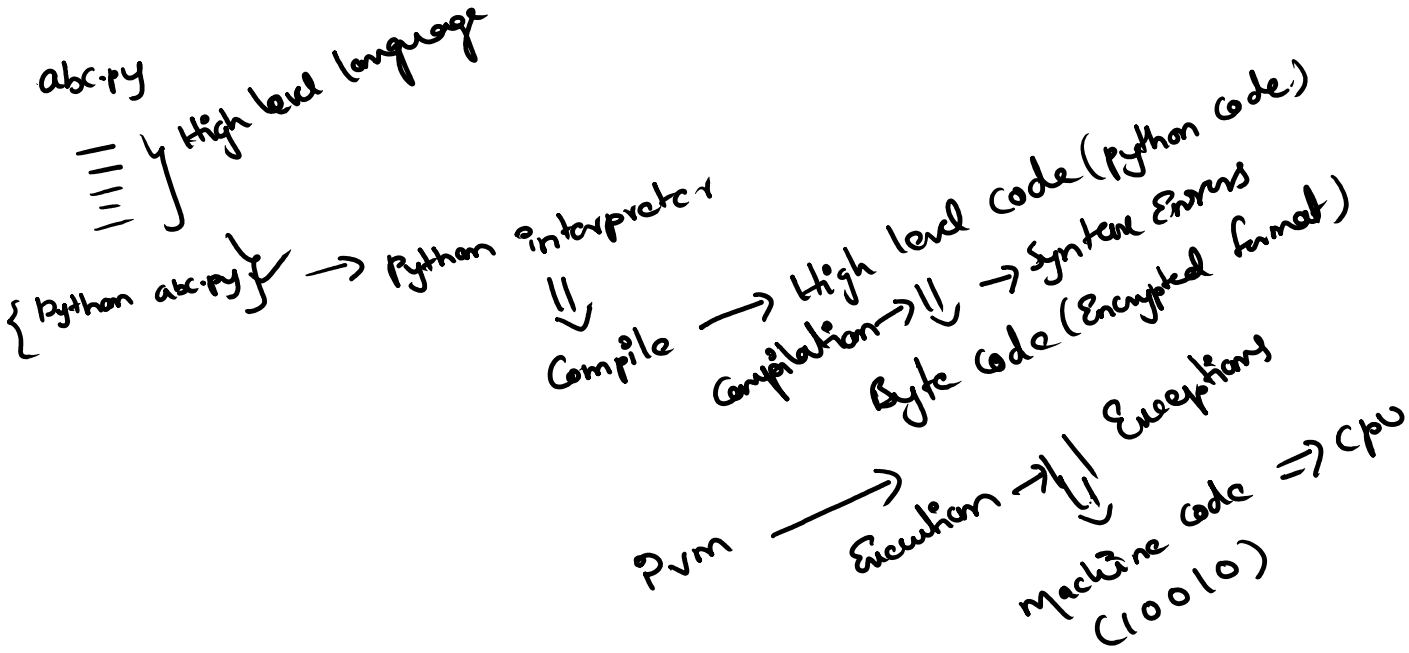
Exception Handling Intro

30 January 2025 22:30

There are 2 stages where error can occur in a program
- During compilation -> Syntax Error
- During execution -> Exceptions



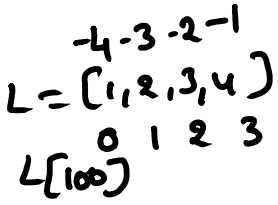
Syntax error & exceptions

30 January 2025 23:29

Syntax error:

--> The code written doesn't follow programming syntax/grammer.
--> This Error is raised by the interpreter/compiler
--> This can be solved by correcting the syntax or grammer

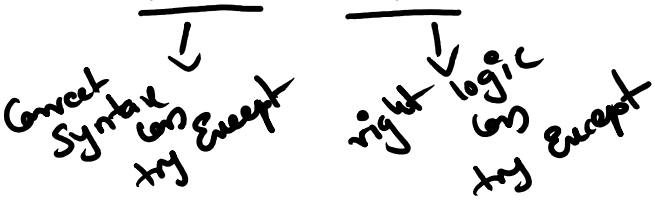
Examples of syntax error
--> Forgetting symbols like colon, brackets
--> Misspelling keywords
--> Incorrect indentation
--> IndexError:
The IndexError is thrown when trying to access an item at an invalid index.
--> ModuleNotFoundError
The ModuleNotFoundError is thrown when a module could not be found.
--> KeyError
The KeyError is thrown when a key is not found in dictionary
--> ValueError
The ValueError is thrown when a function's argument is of an inappropriate data type.
--> NameError
The NameError is thrown when accessing an object/variable that is not defined/initialized.
--> AttributeError
It is raised when calling inappropriate method



Exceptions:

If any error occurs during the execution of the program called exceptions.
- Exceptions are raised in runtime
- We can handle by writing right logic or through try except block

Note: Both syntax errors and exceptions can be handled via try except block



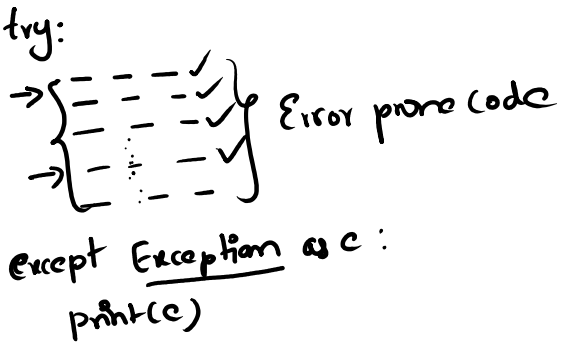
Why it is important to handle?
-->User experience (Enhancing)
-->security reasons (Hackers can't hack it)

try except block

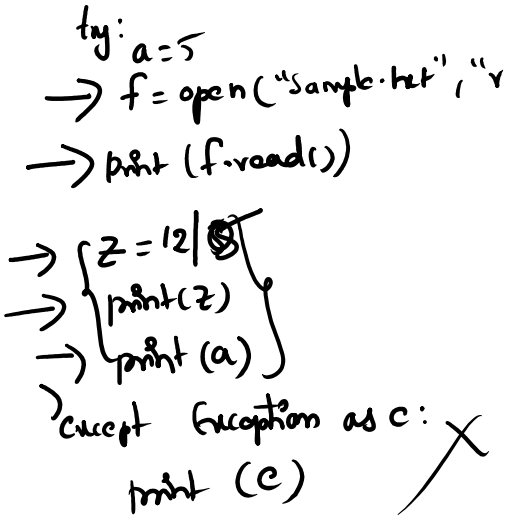
30 January 2025 23:31

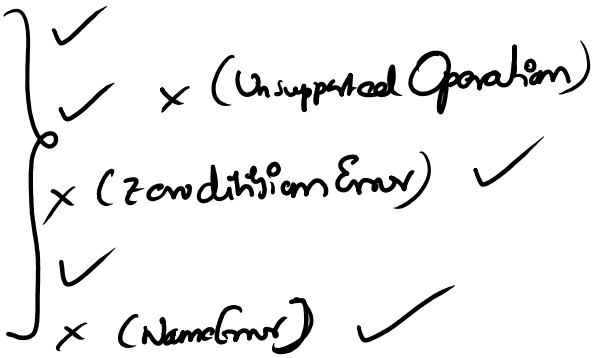
1. Understanding try except block
2. Catching specific exceptions
3. Usage of else with try except
4. Finally with try except else
5. raise keyword

Understanding try except block



Catching specific exceptions





Note :- Remember atleast 10 Errors:

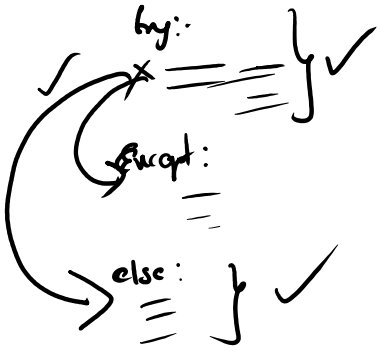
-> go with specific Exception which can handle those Errors

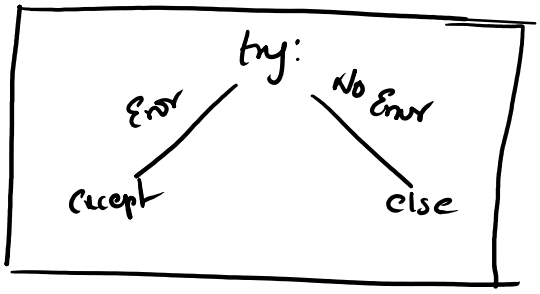
-> At the End u keep one General Exception

try:-

```
except Err1
except Err2
  :
except Exception as e:
    print(e)
```

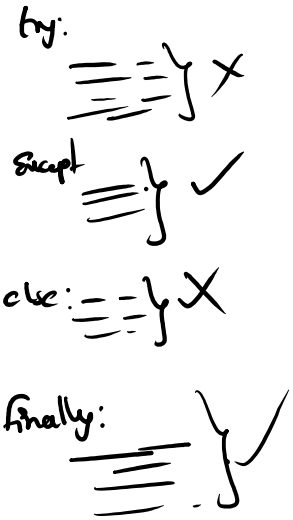
Usage of else with try except

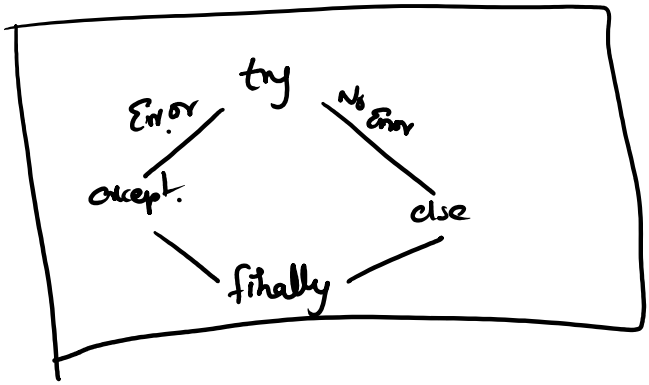




Note:- put your Error free code inside else block, which you think should get Executed after successful code Execution in try block.

Finally with try except else





Scenario:-

Scenario:-

```
try:
    conn = cursor(user, pass, ip, port)
    prod_table = conn.connect("products")
    prod_table.insert("—", " ", " ", " ")
    prod_table.save()
except:
    print(e)
else:
    prod_table.Entries(id = 1)
finally:
    conn.close()
```

local → DB
product
insert
print(e)
closed

Bluetooth

Eg:-

Ex:- DB connections, file access, cloud services etc.

raise keyword

→ It will allow the developer to raise specific Exceptions with Execution of code

Ex:-

```
def withdraw(amount, balance = 10000):
    if amount > balance:
        raise Exception("your balance is low")
    elif amount < 0:
        raise Exception("your amount should be positive")
    else:
        amount_withdrawn = balance - amount
        return amount_withdrawn
```

10 positive
10

definition

```
try:
    a = withdraw(-100)
except Exception as e:
    print(e) # "your amount should be positive"
```

except Exception as e:
    print(e) # "your num should be positive"
else:
    print(a)