[Day8]

Exception Handelling - Allows to handle errors gracefully and take corrective actions without stopping the execution of the program

Exceptions : Events that disrupts the normal flow of a program. They occur when an error is encountered during program execution

Example :
- ZeroDivisionError : dividing by zero
- FileNotFoundError : File not found
- ValueError : Invalid type
- TypeError : Invalid type

try-except

```
try:
    a=b
except NameError as en
    print(en)
```

```
try:
    result = 1/0
except ZeroDivisionError as ex:
    print(ex)
    print("Please enter denominator
          greater than Zero")
```

```
try:
    #code (error code)
except exception as ext1
    #code exception
```

try-except-else block

```
try:
    num = int(input("Enter a number:"))
    result = 10/num
except ValueError:
    print("Not valid number")
except ZeroDivisionError:
    print("You can't divide by zero!")
except Exception as ex:
    print(ex)
else:
    print(f"the result is {result}")
```

try-except-else-finally

```
    # Some error code then
finally:
    print("Execution
          completed")
```

It is executed even if the error doesn't encountered.

File handling & Exception Handling

```
try:
    file = open('example.txt', 'r')
    Content = file.read()
    a=b
    print(content)
except FileNotFoundError:
    print("File does not exists")
except Exception as ex:
    print(ex)
finally:
    if 'file' in locals() or not file.closed:
        file.close()
```