```
#file handling file ma banako kura haru hard disk ma basxa

f = open("hello.txt","w") # file banauna lai open w for file ma kei lekhna lai
f.write("Hello World") # file bhitra write garna lai
f.close()# file open garisake paxi close pani garna paryo
```

```
f = open("hello.txt","r")# file read garna lai
print(f.read())
f.close()
```

```
Hello World
```

```
1/0
```

```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
Cell In[4], line 1
----> 1 1/0

ZeroDivisionError: division by zero
```

```
print('hi')
1/0 # error ayera tala ko code run huna payena
print('helllo')
```

```
hi
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
Cell In[5], line 2
      1 print('hi')
----> 2 1/0
      3 print('helllo')

ZeroDivisionError: division by zero
```

```
f = open("hello.txt","r")# file read garna lai
print(f.read())
1/0
f.close()
```

```
Hello World
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
Cell In[6], line 3
      1 f = open("hello.txt","r")# file read garna lai
      2 print(f.read())
----> 3 1/0
      4 f.close()

ZeroDivisionError: division by zero
```

```
f.closed() # error ayo
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[7], line 1
----> 1 f.closed()

TypeError: 'bool' object is not callable
```

```
#Error handling in file/error/exception handling
```

```
def division(a,b):
    div = a/b
    return div
```

```
division(4,2)
```

```
2.0
```

```
division(4,0)
```

```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
Cell In[13], line 1
----> 1 division(4,0)

Cell In[10], line 2, in division(a, b)
      1 def division(a,b):
----> 2     div = a/b
      3     return div

ZeroDivisionError: division by zero
```

```
#error ayo aba error handle garna try catch use garne
def division(a,b):
    try:
        div = a / b
        return div
    except ZeroDivisionError: # zero division error matra handle gareko
        print("cannot divide by 0.")
```

```
division(4,2)
```

```
2.0
```

```
division(4,0)
print('hello')
```

```
cannot divide by 0.
hello
```

```
division(4,'a')
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[26], line 1
----> 1 division(4,'a')

Cell In[22], line 4, in division(a, b)
      2 def division(a,b):
      3     try:
----> 4         div = a / b
      5         return div
      6     except ZeroDivisionError:

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

```
def division(a,b):
    try:
        div = a / b
        return div
    except ZeroDivisionError: # zero division error matra handle gareko
        print("cannot divide by 0.")
    except TypeError: # farak datatype aune wala error handle gareko
        print("Invalid datatype")
```

```
division(4,'a')
```

```
Invalid datatype
```

```
#jasto type po exception pani handle garna lai

def division(a,b):
    try:
        div = a / b
        return div
```

```
    except ZeroDivisionError: # zero division error matra handle gareko
        print("cannot divide by 0.") # yesari lekhne user lai bujhauna lai natra pytho vdeko error bujnna garo hunxa
    except TypeError: # farak datatype aune wala error handle gareko
        print("Invalid datatype")
    except Exception as e:
        print(e) # Exception as e garera jasto error pani handle garna sakinxa


def division(a,b):
    try:
        div = a/b
        return div
    except Exception as e:
        print(e)



division(4,2)
```

```
2.0
```

```
division(4,0)
```

```
division by zero
```

```
division('a',2)
```

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

```
def division(a,b):
    try:
        div = a / b

    except ZeroDivisionError: # zero division error matra handle gareko
        print("cannot divide by 0.") # yesari lekhne user lai bujhauna lai natra pytho vdeko error bujnna garo hunxa
    except TypeError: # farak datatype aune wala error handle gareko
        print("Invalid datatype")
    except Exception as e:
        print(e)
    else: # try wala part run vayo vane yo code run hunxa
        print("I will run if there is no exception/error")
        return div


division(4,2)
```

```
I will run if there is no exception/error
2.0
```

```
division(4,'a')
```

```
Invalid datatype
```

```
#finally

def division(a,b):
    try:
        div = a / b

    except ZeroDivisionError: # zero division error matra handle gareko
        print("cannot divide by 0.") # yesari lekhne user lai bujhauna lai natra pytho vdeko error bujnna garo hunxa
    except TypeError: # farak datatype aune wala error handle gareko
        print("Invalid datatype")
    except Exception as e:
        print(e)
    else: # try wala part run vayo vane yo code run hunxa
        print("I will run if there is no exception/error")
        return div

    finally: #error aye pani naye ani finallu run hunxa jaile pani
        print("I will run no matter what happens in the code")



division(4,2)
```

```
I will run if there is no exception/error
I will run no matter what happens in the code
2.0
```

```
division(2,0)
```

```
cannot divide by 0.
I will run no matter what happens in the code
```

```
f = open("hello.txt","w")
f.write("Hello world")

try:
   1/0

except Exception as e:
    print(e)
f.close()
```

```
division by zero
```

```
#python ma file open close garna with use garne automatically open ra close garxa file lai

with open("hello.txt","r") as f:
    print(f.read())
```

```
Hello world
```

```
f.closed
```

```
True
```

```
with open("hello.txt","r") as f:
    print(f.closed)
    print(f.read())
    print(f.closed)

print(f.closed) # file closed garna lai with ko sida linema huna paryo
```

```
False
Hello world
False
True
```

```
with open("hello.txt","r") as f:
    print(f.closed)
    print(f.read())
    1/0
```

```
False
Hello world
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
Cell In[63], line 4
      2 print(f.closed)
      3 print(f.read())
----> 4 1/0

ZeroDivisionError: division by zero
```

```
#error aye pani file afai open ra close garxa

with open("hello.txt","r") as f:
    print(f.read())
```

```
Hello world
```

```
with open("hello.txt","rt") as f: # default (rt)=> read in text mode r ra rt autai ho
    print(f.read())
```

```
Hello world
```

```
with open("hello.txt") as f: # default=> read in text mode kei nabane pani read garxa yesle kinani tala read banako xa
    print(f.read())
```

```
Hello world
```

```
#binary file read garna lai aile samma text file read garirako xam
# text modema binary file read garna mildaina


with open("python.png",'rb') as f: # default=> read in text mode kei nabane pani read garxa yesle kinani tala read banako xa
    print(f.read())
```

```
b'\x89PNG\r\n\x1a\n\x00\x00\x00\rIHDR\x00\x00\x00@\x00\x00\x00@\x08\x06\x00\x00\x00\xaaiq\xde\x00\x00\x00\x06bKGD\x00\xff\x00\xff\x00\xf
```

```
#modes haru note ma cha



#JSON string
#every programming language ko afnai structure hunxa farak farak
#applicatioon ra programming language bich communication garna json bata garxa
#backend le json use garera data pathauxa


import json

person = {
     "name": 'ram',
     "age": 22,
     "address": 'ktm',
     "skills":['python','machine learning'],
     "is_programmer":True,
     "phone":None
}

x=json.dumps(person)
print(x,type(x))
```

```
{"name": "ram", "age": 22, "address": "ktm", "skills": ["python", "machine learning"], "is_programmer": true, "phone": null} <class 'str
```

```
import json

person = [{
     "name": 'ram',
     "age": 22,
     "address": 'ktm',
     "skills":['python','machine learning'],
     "is_programmer":True,
     "phone":None
}]

x=json.dumps(person)
print(x,type(x))
```

```
[{"name": "ram", "age": 22, "address": "ktm", "skills": ["python", "machine learning"], "is_programmer": true, "phone": null}] <class 's
```

```
#json string  lai convert garne aru bata ako lai afule bujne format ma

 # json ma single quotion hudaina double quotion huna paryo True pani true hunxa ra None hudaina null hunxa
import json

json_string = """
{
     "name": "ram",
     "age": 22,
     "address": "ktm",
     "skills":[
      "python",
     "machine learning",
     "django"
     ],
     "is_programmer":true,
```

```
    "phone":null
}
"""

x= json.loads(json_string)
print(x,type(x))
```

```
{'name': 'ram', 'age': 22, 'address': 'ktm', 'skills': ['python', 'machine learning', 'django'], 'is_programmer': True, 'phone': None} <
```

```
import json

person = {
    "name": 'ram',
    "age": 22,
    "address": 'ktm',
    "skills":['python','machine learning'],
    "is_programmer":True,
    "phone":None
}

x=json.dumps(person)
print(x,type(x))


#convert  the dictionary to json and write in file
with open("person.json", "w") as f:
    json.dump(person,f)
```

```
{"name": "ram", "age": 22, "address": "ktm", "skills": ["python", "machine learning"], "is_programmer": true, "phone": null} <class 'str
```

```
#file lai read garera json ma convert garera python nikalne
import json
with open("person.json", "r") as f:
    data = f.read()
    x= json.loads(data)
    print(x['skills'][0],type(x))
```

```
python <class 'dict'>
```

```
import json # yesari direct garna ni milyo
with open("person.json", "r") as f:
    x= json.load(f)
    print(x['skills'][0],type(x))
```

```
python <class 'dict'>
```

Start coding or generate with AI.