

FILE HANDLING AND ERROR HANDLING IN PYTHON

Operating a File

Reading a file

```
#creating a file  
file = open("/content/pooja.txt", "r")
```

Writing to a file

```
file = open("/content/pooja.txt", "w")  
file.write("Hello, World!\n")  
file.close
```

```
<function TextIOWrapper.close()>
```

Appending to a file

```
file = open("example.txt", "a")  
file.write("This is an appended line. \n")  
file.close()
```

Using `with` statement

```
with open("/content/pooja.txt", "r") as file:  
    content = file.read()  
    print(content)
```

```
maria is a bad girl
```

File handling modes

```
with open("/content/image.jpg", "rb") as file:  
    data = file.read()
```

Error Handling

Try except block

```
try:  
    num = int(input("enter a number:"))  
    print(10/num)  
except ZeroDivisionError:  
    print("you cannot divide by zero")  
except ValueError:  
    print("Invalid input! Please enter a number.")
```

```
enter a number:0
you cannot divide by zero
```

Finally Block

```
try:
    file = open("/content/pooja.txt", "r")
except FileNotFoundError:
    print("File not found.")
finally:
    print("Execution complete.")
```

Execution complete.

Raising exceptions

```
def check_age(age):
    if age < 18:
        raise ValueError("Age must be 18 or older.")
    return True

try:
    check_age(16)
except ValueError as e:
    print(e)
```

Age must be 18 or older.

Reading and Writing to a file:

```
with open("/content/pooja.txt", "a") as file:
    file.write("Python is awesome!\n")

# Opening the file in read mode to print its contents
with open("/content/pooja.txt", "r") as file:
    print(file.read())
```

Python is awesome!

Appending data to a file

```
with open("/content/pooja.txt", "a") as file:
    file.write("Let's learn file handling.\n")

with open("/content/pooja.txt", "r") as file:
    print(file.read())
```

Python is awesome!
Let's learn file handling.

Handling Division by zero error

```
try:
    num1 = int(input("enter numerator:"))
    num2 = int(input("enter denominator:"))
    result = num1/num2
    print("Result:", result)
except ZeroDivisionError:
    print("Cannot divide by zero.")
except ValueError:
    print("Invalid input! Enter numeric values.")

enter numerator:50
enter denominator:25
Result: 2.0
```

Creating a custom Exception

```
class NegativeNumberError(Exception):
    pass

def check_positive(number):
    if number < 0:
        raise NegativeNumberError("Negative number entered.")
    return True

try:
    num = int(input("Enter a positive number:"))
    check_positive(num)
    print("Number is positive number")
except NegativeNumberError as e:
    print(e)

Enter a positive number: 22
Number is positive number
```