**Nested try blocks**
try block within try block is called nested try block.

**Syntax**

```
try: → Outer try block
   statement-1
   statement-2
   try:  → Inner try block
       statement-3
       statement-4
   except <exception-name>:
      statement-5
except <exception-name>:
      statement-6
```

if there is an exception inside inner try block, it is handled by inner except block. If inner except block not able to handle, it is able to handle by outer except block.
If there is an exception inside outer try block, it is handled by outer except block.

**Example:**

```
try:
    a=int(input("enter first number "))
    b=int(input("enter second number "))
    try:
        c=a/b
        print(f'{a}/{b}={c}')
    except ZeroDivisionError:
        print("cannot divide number with zero")
except ValueError:
    print("invalid input or input must be integers")
```

**Output:**

```
enter first number 5
enter second number 0
cannot divide number with zero
```

**Files or Working with files**

Data or objects created inside RAM or Main Memory is temporary. To save the data or objects permanently python uses file system.
A file is collection data or information.
A file is named memory location in secondary storage device (Disk).

**Types of files**

There are two types of files

1. Text file
2. Binary file

Text file allows only text type or string type data.
Binary file allows bytes data (image, audio, video,…)

**Basic steps to work with files**

1. **Open file**
2. **Read/Write**
3. **Close file**

**open()**

**open()** is a predefined function in python. This function opens the file in a given mode and type and return file object. This file object is used to perform operation on file.

**Syntax:**
variable=open("filename","mode")

filename is a string or path.
File is open in different modes

| **Mode** | **Description** |
|---|---|
| w | Write mode, this mode create new file for writing data.<br>If filename exists it overwrites. |
| r | Read mode, this opens the for reading. If filename not exists it raises FileNotFoundError |
| a | Append mode, This mode allows to add more data to an existing file. If file is not exists, it create new file for adding or writing data. |
| x | open for exclusive creation, failing if the file already exists |
| r+ or | This mode is for reading and writing/update |

| | |
|---|---|
| w+ | |
| t | Text type (default) “w” → “wt” “r” → “rt” |
| b | Binary type  “wb”,”rb”,”ab”,”xb” |

**Working with text files**

In text file data is stored in text format or it allows only string.

The following functions are used for writing and reading data from text file

1. Write(s)
2. Writelines()
3. Print()
4. Read()
5. Readline()

**write(s)**

This method writes string inside file. After writing it does not insert new line.

```
# Creating text file
import sys
try:
    f=open("file1.txt","w") # current working directory
    f.write("Python is a high level programming language")
    f.write(str(10))
    f.write(str(1.5))
    print("data is saved...")
except:
    t=sys.exc_info()
    print(t[1])
finally:
    f.close()
```

**Output:**

```
Data is saved…
```

**writelines()**

This method is used to write more than one value inside file.
The values has to be represented as list.

**Example:**

```
# Creating text file

try:
    f=open("file2.txt","w")
    data=["101\n","naresh\n","python\n","4000.0\n"]
    f.writelines(data)
    print("data saved...")
except:
    print("error")
finally:
    f.close()
```

**Output:**

```
 Data saved…
```

**Example:**

```
# Create text file

try:
    f=open("file3.txt","w")
    while True:
        rollno=int(input("Enter Rollno "))
        name=input("Enter Name ")
        print(rollno,name,file=f)
        ch=input("Add another Student ?")
        if ch=="no":
            break
except:
    print("error")
finally:
    f.close()
```

**Output:**

```
Enter Rollno 1
Enter Name naresh
Add another Student ?yes
Enter Rollno 2
Enter Name suresh
Add another Student ?yes
```

```
Enter Rollno 3
Enter Name kishore
Add another Student ?no
```

**read(size)**
read() method, read given size of bytes or characters from file. If size not defined, it read complete file and return as one string.

**Example:**

```
# reading text file

try:
    f=open("file1.txt","r")
    s=f.read()
    print(s)
except:
    print("error")
finally:
    f.close()

try:
    f=open("file1.txt","r")
    s=f.read(5)
    print(s)
except:
    print("error")
finally:
    f.close()
```

**Output:**

```
Python is a high level programming language101.5
Pytho
```

**Example:**

```
# Write a program to count alphabets inside file

try:
    f=open("file1.txt","r")
    c=0
    while True:
        s=f.read(1)
        if s=='':
            break
        if (s>='A' and s<='Z') or (s>='a' and s<='z'):
            c=c+1
except:
    print("error")
finally:
    f.close()

print(f'Count of Alphabets {c}')
```

**Output:**

```
Count of Alphabets 37
```

**readline (size)**
if size is not defined readlines  read one line.
If size is defined, it read those many number of characters.

```
# Reading data from file

try:
    f=open("file3.txt","r")
    stud1=f.readline()
    stud2=f.readline()
    print(stud1,end='')
    print(stud2,end='')
    stud3=f.readline()
    print(stud3,end='')
except:
    print("error")
finally:
    f.close()
```

**Output:**

```
1 naresh
2 suresh
3 kishore
```

CSV file
JSON file