**Data type**

**String:**

name = "irfan"  
print(name)

output: Irfan

name = "irfan"  
print(type(name))

output: str

name = "irfan"  
print("My name is "+name)

output: My name is Irfan

first\_name = "Irfan"  
middle\_name="Shah"  
last\_name ="Mayeen"  
full\_name =first\_name +" " +middle\_name +" "+last\_name  
print(full\_name)

output: Irfan Shah Mayeen

**Integer:**

year = 21  
printf(year)

output:21

year = 21  
print(type(year))

output: int

year = 21  
year=year+4  
print(year)

output:25

Now the problem is

year = 21  
printf("My age is "+year)

output: Errror.

Solution: typecasting variable

year = 21  
print("My age is "+str(year))

Output: My age is 21

**Float:**

area = 15.5  
print(area)

Output:15.5

area = 15.5  
print(type(area))

Output:float

**Boolean:**

ans = True  
print(ans)

Output:true

ans = True  
print(type(ans))

Output: bool

ans = True  
print("Are you human? "+str(ans))

Output: Are you human?t true

Multiple Variable in single line

name , age , area,ans ="Irfan",21,15.5,True

Multiple Variable Contain same value

irfan=sibbir=miraz=22

**String**

Length

Length =len(String)

Find index

name ="Irfan Shah"  
print(name.find("I"))

Output: 0

name ="Irfan Shah"  
print(name.find("r"))

Output: 1

name ="Irfan Shah"  
print(name.find("p"))

Output: -1

Capitalize( first letter will be upper case)

name ="irfan shah"  
print(name.capitalize())

Output: Irfan shah

Upper and Lowe case letter( it will not change the string just print)

name ="Irfan Shah"  
print(name.lower())  
print(name.upper())

Output: irfan shah

IRFAN SHAH

Digit or not(full string)

name ="Irfan Shah"  
print(name.isdigit())

output:False

name ="1236"  
print(name.isdigit())

output: True

name ="123 6"  
print(name.isdigit())

output:False ( because here a space after 3. It is not a digit)

Alphabet or not(full string)

name="Irfan shah"  
print(name.isalpha())

output: false ( because space after n)

name="Irfanshah"  
print(name.isalpha())

output:True

name="123"  
print(name.isalpha())

Output:False

Count frequency

name ="aaabbbbccc"

print(name.count("a"))

output:3 .(here a arrives 3 time)

name ="aaabbbbccc"

print(name.count("b"))

output:4

Replace

name ="aaaccc"

print(name.replace("a","b"))

output: bbbccc ( all a replaced by b)

String Print n times

name ="Bangladesh"

print(name\*3)

output: BangladeshBangladeshBangladesh ( bangladesh print 3 times)

name ="Irfan Shah"

print(name)

print(len(name))

print(name.find("f"))

print(name.capitalize())

print(name.lower())

print(name.upper())

print(name.isdigit())

print(name.isalpha())

print(name.count("a"))

print(name\*3)

**Typecasting**

Type casting = convert the data type of a value to another data type

x=1 #int  
y=2.0 #float  
z="3" #str  
  
print(int(x))  
print(int(y))  
print(int(z))

Output: 1

2

3

y=2.0 #float  
z="3" #str

y=int(y)  
z=int(z)  
print(y)  
print(z)

Output: 1

2

3

**User input**

Input();

name = input("what is your name:")  
age=input("How old are you: ")  
print(name)  
print(age)

output: what is your name:irfan

How old are you: 18

irfan

18

age=input("How old are you: ")  
age=age+1

print(age)

Output: Error

Solution: typecasting variable

age=int(input("How old are you: "))

**Math Functions**

import math  
pi=3.14  
print(round(pi))  
print(math.ceil(pi))  
print(math.floor(pi))  
print(abs(pi))  
print(pow(3,2))  
print(math.sqrt(64))

3

4

3

3.14

9

8.0

maximum && minimum

x=2  
y=3  
z=6  
print(max(x,y,z))

print(min(x,y,z)

Output:6 (maximum number)

2 (minimum number)

**String Slicing**

name ="Irfan Shah"  
first\_name= name[0:5]  
print(first\_name)

output: Irfan

name ="Irfan Shah"  
last\_name=name[6:10]  
print(last\_name)

output: shah

name ="Irfan Shah"  
funky\_name=name[0:8:2]  
print(funky\_name)

Output : IfnSa (print 1 skip character)

name ="Irfan Shah"

funky\_name=name[0:10:3]  
print(funky\_name)

Output: IaSh

name ="Irfan Shah"

funky\_name=name[::2]  
print(funky\_name)

output: IfnSa ( here name[ : :🡪 python assume first empty ,last empty

Reverse string

name ="Irfan Shah"

reverse\_name= name[::-1]  
print(reverse\_name)

Output: hahS nafrI

Slice() function

website = "http://google.com"  
slice = slice(7,-4)  
print(website[slice])

Output: google

website = "http://google.com"  
website2= "http://facebook.com"  
slice = slice(7,-4)  
print(website[slice])  
print(website2[slice])

Output: google

facebook

**Else if**

age =int(input("Enter your age:"))  
if age>=18 :  
 print("Your are an adult")  
else:  
 print("you are a child")

age =int(input("Enter your age:"))  
if age>=18 :  
 print("Your are an adult")  
elif age<0:  
 print("You have not born")  
else:  
 print("you are a child")

age =int(input("Enter your age:"))  
if age>=18 :  
 print("Your are an adult")  
elif age==12:  
 print("you are 12 years old")  
elif age<0:  
 print("You have not born")  
else:  
 print("you are a child"

Conditional operator

temp=float(input("Enter temperature"))  
  
if temp >= 0 and temp <= 30:  
 print("Weather is good")  
 print("Go outside!")  
elif temp<0 or temp>30:  
 print("the weather bad today")  
 print("Don't Go!")

**Loops**

While loop

name =""  
while len(name)==0:  
 name = input("Enter your name: ")  
print("Hello "+name)

or

name = None  
while len(name)==0:  
 name = input("Enter your name: ")  
print("Hello "+name)

For loop

for i in range(5):  
 print(i)

Output:

0

1

2

3

4

for i in range (2,5):  
 print(i)

output:

2

3

4

for i in range (2,10,2):  
 print(i)

output:

2

4

6

8

for i in range (1,10,2):  
 print(i)

output:

1

3

5

7

9

for i in "Irfan":  
 print(i)

outputs:

I

r

f

a

n

## Timer

for second in range(5,0,-1):  
 print(second)  
 time.sleep(1)  
print("Happy Birthday!")

Nested loop

row = int(input("Enter rows:"))  
col = int(input("Enter colums:"))  
symbol = input("Enter a symbol:")  
for i in range(row):  
 for j in range(col):  
 print(symbol,end="")  
 print()

outputs:

Enter rows:3

Enter colums:3

Enter a symbol:\*

\*\*\*

\*\*\*

\*\*\*

For control’

#break

while True:  
 name = input("Enter your name:")  
 if name!="":  
 break

#continue

phone\_number="0199-334-3853"  
for i in phone\_number:  
 if i=="-":  
 continue  
 print(i,end="")

#pass

for i in range(1,10):  
 if i== 5:  
 pass  
 else:  
 print(i,end=" ")

outputs: 1 2 3 4 6 7 8 9

**List**

# list = used to store multiple items in a asingle variable

food = ["pizza" , "burger" , "chicken" ,"hotdog"]  
print(food)

outputs: ['pizza', 'burger', 'chicken', 'hotdog']

food = ["pizza" , "burger" , "chicken" ,"hotdog"]  
print(food[0])  
print(food[1])  
print(food[2])

outputs:

pizza

burger

chicken

ood = ["pizza" , "burger" , "chicken" ,"hotdog"]  
food[0]="rool"  
print(food[0])

outputs: rool

food = ["pizza" , "burger" , "chicken" ,"hotdog"]  
for x in food:  
 print(x)

outputs:

pizza

burger

chicken

hot dog

#add element in the last

food = ["pizza" , "burger" , "chicken" ,"hotdog"]  
food.append("ice cream")  
for x in food:  
 print(x)

outputs:

pizza

burger

chicken

ice cream

#remove element from last

food = ["pizza" , "burger" , "chicken" ,"hotdog"]  
food.pop()  
for x in food:  
 print(x)

outputs:

pizza

burger

chicken

#insert given index

food = ["pizza" , "burger" , "chicken" ,"hotdog"]  
food.insert(0,"cake")  
for x in food:  
 print(x)

outputs:

cake

pizza

burger

chicken

ice cream

#sort

food = ["pizza" , "burger" , "chicken" ,"hotdog"]  
food.sort()  
for x in food:  
 print(x)

outputs:

burger

chicken

hotdog

roll

#clear list

food.clear()

**2D List**

drinks = ["coffe","tea","soda"]  
dinner= ["pizza","hotdog","kacchi"]  
desert = ["cake","ice cream"]  
food =[drinks,dinner,desert]  
print(food)

outputs: [['coffe', 'tea', 'soda'], ['pizza', 'hotdog', 'kacchi'], ['cake', 'ice cream']]

drinks = ["coffe","tea","soda"]  
dinner= ["pizza","hotdog","kacchi"]  
desert = ["cake","ice cream"]  
food =[drinks,dinner,desert]  
print(food[0])

outputs : ['coffe', 'tea', 'soda']

drinks = ["coffe","tea","soda"]  
dinner= ["pizza","hotdog","kacchi"]  
desert = ["cake","ice cream"]  
food =[drinks,dinner,desert]  
print(food[0])  
print(food[1])  
print(food[2])

outputs :

['coffe', 'tea', 'soda']

['pizza', 'hotdog', 'kacchi']

['cake', 'ice cream']

drinks = ["coffe","tea","soda"]  
dinner= ["pizza","hotdog","kacchi"]  
desert = ["cake","ice cream"]  
food =[drinks,dinner,desert]  
print(food[0][0])

outputs : coffe

drinks = ["coffe","tea","soda"]  
dinner= ["pizza","hotdog","kacchi"]  
desert = ["cake","ice cream"]  
food =[drinks,dinner,desert]  
print(food[1][2])

outputs : kacchi

**Tuple**

#student (name,age,gender  
student = ("Irfan",21,"male")  
print(student.count("Irfan"))

Outputs: 1 ( because here only one Irfan)

student = ("Irfan",21,"male")  
print(student.index("male"))

Outputs: 2 ( because male index number is 2)

student = ("Irfan",21,"male")  
for x in student:  
 print(x,end=" ")

outputs : Irfan 21 male

student = ("Irfan",21,"male")  
if "Irfan" in student:  
 print("Irfan is here")

outputs : Irfan is here

#problems of tuple

Tuple are unchanble

Student[0]= “Rahim” -> it is not allowed

student = ("Irfan",21,"male")  
print(student[0])  
print(student[1])

outputs : Irfan 21

#can be create multiple tuple

teachers = (("Ishan",26,"male"),("ashik",35,"male"))  
print(teachers[0])

outputs : ('Ishan', 26, 'male')

teachers = (("Ishan",26,"male"),("ashik",35,"male"))  
print(teachers[0])  
print(teachers[1][1])

outputs : 35

**Set**

food = {"pizza","burger","kacchi"}  
for x in food:  
 print(x,end=" ")

outputs : pizza burger kacchi

#if we add duplicate , it will print only one

subject ={"English","Bangla","computer","Bangla","Bangla"}  
for x in subject:  
 print(x,end=" ")

outputs : computer Bangla English 🡪(herew Bangla is more than one but print only one time)

food = {"pizza","burger","kacchi"}  
food.add("rool")  
food.remove("burger")  
food.clear()

#add two ser

game = {"cricket","football","hockey"}  
team = {"Bangladesh","India","Argentina"}  
game.update(team)  
for x in game:  
 print(x,end=" ")

outputs: cricket Argentina India hockey Bangladesh football 🡪 (random)

game = {"cricket","football","hockey"}  
team = {"Bangladesh","India","Argentina"}  
field = game.union(team)  
for x in field:  
 print(x,end=" ")

outputs: India football hockey Bangladesh Argentina cricket🡪 (random)

game = {"cricket","football","hockey"}  
team = {"Bangladesh","India","Argentina"}  
field = game.union(team)  
print(field)

outputs: {'Argentina', 'cricket', 'football', 'hockey', 'Bangladesh', 'India'}

#difference

animal = {"Tiger","Lion","Hen"}  
bird = {"Doel","Hen","Duck"}  
print(animal.difference(bird))

output: {'Lion', 'Tiger'} 🡪 from animal hen removed

animal = {"Tiger","Lion","Hen"}  
bird = {"Doel","Hen","Duck"}  
print(bird.difference(animal))

output: {'Duck', 'Doel'} 🡪 from bird Hen removed

animal = {"Tiger","Lion","Hen"}  
bird = {"Doel","Hen","Duck"}  
print(bird.difference(bird))

output: set() 🡪 every element remove form bird …because difference with bird to bird

#intersection (commom)

animal = {"Tiger","Lion","Hen"}  
bird = {"Doel","Hen","Duck"}  
print(animal.intersection(bird))

output: {‘Hen’} 🡪 only Hen is both in set

animal = {"Tiger","Lion","Hen"}  
bird = {"Doel","Hen","Duck"}  
print(animal.intersection(animal))

output: {'Tiger', 'Lion', 'Hen'} 🡪 because every element is available in both set ,.. cause both set are same animal

**Dictionary**

#dictionary = A changeable , unordered collection of unique key: value pairs  
# Fast beacuse they are use hashing , allow us to access a value quickly

capital={'Bangladesh':'Dhaka',  
 'India':'Dehli',  
 'China':'Beijing',  
 'Russia':'Moscow'}  
print(capital['Russia'])  
print(capital['Bangladesh'])

output: Moscow

Dhaka

capital={'Bangladesh':'Dhaka',  
 'India':'Dehli',  
 'China':'Beijing',  
print(capital['USA'])

output: Error 🡪 because USA is not in our dictionary

capital={'Bangladesh':'Dhaka',  
 'India':'Dehli',  
 'China':'Beijing',  
 'Russia':'Moscow'}  
print(capital.get('Bangladesh'))  
print(capital.get('USA'))

output: Dhaka

None

#print keys

capital={'Bangladesh':'Dhaka',  
 'India':'Dehli',  
 'China':'Beijing',  
 'Russia':'Moscow'}  
print(capital.keys())

output: dict\_keys(['Bangladesh', 'India', 'China', 'Russia'])

#print values

capital={'Bangladesh':'Dhaka',  
 'India':'Dehli',  
 'China':'Beijing',  
 'Russia':'Moscow'}

print(capital.values())

outputs: dict\_values(['Dhaka', 'Dehli', 'Beijing', 'Moscow'])

#entire dictionary print

capital={'Bangladesh':'Dhaka',  
 'India':'Dehli',  
 'China':'Beijing',  
 'Russia':'Moscow'}  
print(capital.items())

output: dict\_items([('Bangladesh', 'Dhaka'), ('India', 'Dehli'), ('China', 'Beijing'), ('Russia', 'Moscow')])

#or

capital={'Bangladesh':'Dhaka',  
 'India':'Dehli',  
 'China':'Beijing',  
 'Russia':'Moscow'}  
for key,value in capital.items():  
 print(key,value)

output:

Bangladesh Dhaka

India Dehli

China Beijing

Russia Moscow

#update (add)

capital={'Bangladesh':'Dhaka',  
 'India':'Dehli',  
 'China':'Beijing',  
 'Russia':'Moscow'}

capital.update({'Germany':'Berlin'})  
for key,value in capital.items():  
 print(key,value)

output:

Bangladesh Dhaka

India Dehli

China Beijing

Russia Moscow

Germany Berlin

#update(change)

apital={'Bangladesh':'Dhaka',  
 'India':'Dehli',  
 'China':'Beijing',  
 'Russia':'Moscow'}

capital.update({'Bangladesh': 'Munshiganj'})  
for key,value in capital.items():  
 print(key,value)

output:

Bangladesh Munshiganj

India Dehli

China Beijing

Russia Moscow

#remove

capital={'Bangladesh':'Dhaka',  
 'India':'Dehli',  
 'China':'Beijing',  
 'Russia':'Moscow'}

capital.pop('India')  
for key,value in capital.items():

output:

Bangladesh Munshiganj

China Beijing

Russia Moscow

**Indexing**

name = "irfan"  
if(name[0].islower()):  
 name=name.capitalize()  
print(name)

output: Irfan

name = "irfan"  
first\_name= name[0:5].upper()  
print(first\_name)

output: IRFAN

name = "irfan Shah"

last\_name=name[6:10].lower()  
print(last\_name)

output: shah

name = "irfan Shah!"  
last\_character = name[-1]  
second\_last =name[-2]  
print(last\_character)  
print(second\_last)

output: !

h

**Function**

def hello():  
 print("hello!")  
  
hello()  
hello()

output: hello!

Hello!

#with parameter

def hello1(name):  
 print("hello "+name)  
 print("Nice to meet you")  
  
hello1("Irfan")

output: hello Irfan

Nice to meet you

def printName(first\_name,last\_name):  
 print("Hello "+first\_name+" "+last\_name)  
  
printName("Irfan","Shah")

output: Hello Irfan Shah

#return

def multiply(num1,num2):  
 result = num1\*num2  
 return result  
  
x=multiply(5,2)  
print(x)

print(multiply(5,3))

output: 10

15

**Keyword Argument**

#keyword arguments = argument preceded by an indentifier when we pass them to a function  
# The order of the arguments doesn't matter,unlike positional arguments  
# python knows the names of the arguments that our function receieves

def hello(first,middle ,last):  
 print("Hello "+first+" "+middle+" "+last)  
hello("Irfan","shah","Mayeen")

output: Hello Irfan Shah Mayeen

#but

def hello(first,middle ,last):  
 print("Hello "+first+" "+middle+" "+last)  
hello("Shah","Irfan","Mayeen")

output: Hello shah Irfan Mayeen 🡪 caller order matter.

##now solve it using keyword argument

def hello(first,middle ,last):  
 print("Hello "+first+" "+middle+" "+last)  
hello(middle="shah",first="Irfan",last="Mayeen")

output: Hello Irfan Shah Mayeen 🡪 pass argument with the same key as parameter

**Nested Function**

num = input("Enter a number :")  
num = float(num)  
num=abs(num)  
num=round(num)  
print(num)

#or

print(round(abs(float(input("Enter a number: ")))))

**Scope**

name= "Irfan"  
def display\_name():  
 name ="code" # local scope (available only inside this function  
 print(name)  
  
print(name)  
display\_name()

output: Irfan 🡪 from Global

code 🡪 from local 🡪ffrom function

**Args\***

#agrs = parameter that will pack all argument into a tuple  
# useful so that a function can accept a varying amount of arguments

def add(\*args):  
 sum=0  
 for i in args:  
 sum+=i  
 return sum  
print(add(1,2,3,4,5,6))

output: 21

def add(\*stuff):  
 sum=0;  
 for i in stuff:  
 sum+=i  
 return sum  
  
print(add(1,2,3,4,5,6))

output: 21

def add(\*stuff):  
 sum=0  
 stuff = list(stuff)  
 stuff[0]=0  
 for i in stuff:  
 sum+=i  
 return sum  
  
print(add(1,2,3,4,5,6))

output: 20 🡪 stuff[0] =1 changed stuff[0]=0

stuff = list(stuff)  
 stuff[0]=0  
 stuff[1]=0  
 for i in stuff:  
 sum+=i  
 return sum  
  
print(add(1,2,3,4,5,6))

output: 18 🡪 stuf[0] =1 changed to 0 , stuff[1] =2 changed to 0 . so total 3 will decrease

**Kwargs\***

def hello(first,last):  
 print("Hello! "+first+" "+last)  
  
hello(first="Irfan",middle="Shah",last="Mayeen" )

output: error 🡪 argument 3 , but function parameter 2

#solve using \*\*kwargs

def hello(\*\*kwargs):  
 print("Hello! "+kwargs['first']+" "+kwargs['last'])  
  
hello(first="Irfan",middle="Shah",last="Mayeen" )

output: Hello! Irfan Mayeen

#full list print

def hello(\*\*kwargs):  
 print("Hello",end=" ")  
 for key,value in kwargs.items():  
 print(value,end=" ")  
  
hello(title="Mr.",first="Irfan",middle="Shah",last="Mayeen")

output: Hello mr. Irfan Shah Mayeen

#if we do For loop like this

def hello(\*\*kwargs):  
 print("Hello",end=" ")  
 for value in kwargs.items():  
 print(value,end=" ")  
  
hello(title="Mr.",first="Irfan",middle="Shah",last="Mayeen")

output: Hello ('title', 'Mr.') ('first', 'Irfan') ('middle', 'Shah') ('last', 'Mayeen')

**format function**

animal = "cow"  
item = "moon"  
print("The "+animal+" jumped over the "+item)

output: The cow jumped over the moon

#uisng format method

print("The {} jumped over the {}".format("cow","moon"))

output: The cow jumped over the moon

#or

animal = "cow"  
item = "moon"  
print("The {} jumped over the {}".format(animal,item))

output: The cow jumped over the moon

#using index positional argument

animal = "cow"  
item = "moon"  
print("The {0} jumped over the {1}".format(animal,item))

output: The cow jumped over the moon

animal = "cow"  
item = "moon"  
print("The {1} jumped over the {0}".format(animal,item))

output: The moon jumped over the cow

print("The {3} jumped over the {2}".format("tiger","Lion","Moon","Cat"))

output: The Cat jumped over the Moon

#keyword argument

print("The {animal} jumped over the {item}".format(animal="lion",item="moon"))

output: The lion jumped over the moon

#using string

text = "The {} jupmed over the {}"  
print(text.format("Tiger","moon"))

output: The Tiger jumped over the moon

#space

name="Irfan"  
print("hello,my name is {}.Nice to meet you".format(name))  
print("hello,my name is {:20}. Nice to meet you".format(name))

output: hello,my name is Irfan.Nice to meet you

hello,my name is Irfan . Nice to meet you

name="Irfan"  
print("hello,my name is {}.Nice to meet you".format(name))  
print("hello,my name is {:20}. Nice to meet you".format(name))  
print("hello,my name is {:<20}. Nice to meet you".format(name)) #right  
print("hello,my name is {:>20}. Nice to meet you".format(name)) #left  
print("hello,my name is {:^20}. Nice to meet you".format(name)) #central

output:

hello,my name is Irfan.Nice to meet you

hello,my name is Irfan . Nice to meet you

hello,my name is Irfan . Nice to meet you

hello,my name is Irfan. Nice to meet you

hello,my name is Irfan . Nice to meet you

#number

number = 3.14159  
print("The number pi is {}".format(number))  
print("The number pi is {:.2f}".format(number))  
print("The number pi is {:.3f}".format(number))

output:

The number pi is 3.14159

The number pi is 3.14

The number pi is 3.142

number = 1000  
print("The number is {}".format(number))  
print("The number is {:,}".format(number))  
print("The number is {:b}".format(number)) #binary  
print("The number is {:o}".format(number))#ocatal  
print("The number is {:x}".format(number)) #hexa-decimal lowercase  
print("The number is {:X}".format(number)) #hexa-decimal uppercase  
print("The number is {:E}".format(number)) #sciencific notation

output:

The number is 1000

The number is 1,000

The number is 1111101000

The number is 1750

The number is 3e8

The number is 3E8

The number is 1.000000E+03

**Random Number**

import random  
x = random.randint(1,6) #integer number  
print(x)  
y = random.random() #floating number  
print(y)

#ramdom choice from a list

myList =['rock','paper','scissors']  
z= random.choice(myList)  
print(z)

#random suffle

cards=[1,2,3,4,5,6,7,8,9,"J","Q","K","A"]  
random.shuffle(cards)  
print(cards)

**Exception Handle**

print(5/0)

it is math maticaly impossible … so it is an exception

numerator = int(input("Enter a number to divide: "))  
denominator = int (input("Enter a numbr to divide by:"))  
result = numerator / denominator  
print(result)

if user input denominator =0 .. it will exception and stops the programso

solution:

try:  
 numerator = int(input("Enter a number to divide: "))  
 denominator = int (input("Enter a numbr to divide by:"))  
 result = numerator / denominator  
 print(result)  
except Exception:  
 print("You enter denominator as 0")

try:  
 numerator = int(input("Enter a number to divide: "))  
 denominator = int (input("Enter a numbr to divide by:"))  
 result = numerator / denominator  
 print(result)  
except ZeroDivisionError as e:  
 print(e)  
 print("you can't divided by zero! idiot!")  
except ValueError as e:  
 print(e)  
 print("Enter only number plz")  
except Exception as e:  
 print(e)  
 print("Something went wronng:")  
else:  
 print(result)  
finally:  
 print("this is alyaws execute")

**File**

**Import os**

import os  
path ="C:\\Users\\Admin\\PycharmProjects\\File\\test.txt"  
if os.path.exists(path):  
 print("that location exists")  
 if os.path.isfile(path):  
 print("That is a file")  
else:  
 print("That location doesn't exists")  
  
location ="C:\\Users\\Admin\\PycharmProjects\\File\\Folder"  
if os.path.exists(location):  
 print("that location exists")  
 if os.path.isdir(location):  
 print("That is a Directory/folder")  
else:  
 print("That location doesn't exists")

output:

that location exists

That is a file

that location exists

That is a Directory/folder

#read a file

#if the file inside the project folder

with open('test.txt') as file:  
 print(file.read())

#if the file is not in the project folder

with open('C:\\Users\\Admin\\PycharmProjects\\File\\test.txt') as file:  
 print(file.read())

#finally

try:  
 with open('test.txt') as file:  
 print(file.read())  
except FileNotFoundError  
 print("File was not found")

#write a file

text = "Irfan"  
with open('test.txt','w') as file:  
 file.write(text)

#but the problem is , the text it overwritten.. / previous text delete

Solve : use append

with open('test.txt','a') as file:

**copy file**

#copyfile()= copies contents of a file  
#copy() = copyfile() + permission mode+destination ca be a directory  
#copy2()= copy() + copies metadata (fie's creation and modification times)  
  
import shutil  
shutil.copfile('test.txt','copy.txt') #(source , destination)

n('test.txt','a') as file:

**move file**

import os  
source ="text.txt"  
destination="C:\\Users\\Admin\\Desktop\\text.txt"  
try:  
 if os.path.exists(destination):  
 print("There is already a file there")  
 else:  
 os.replace(source,destination)  
 print(source+" was moved")  
  
except FileNotFoundError:  
 print(source+" "+"was not found")

**Delete file**

import os  
os.remove('test.txt')

#or

import os  
os.remove('test.txt')  
  
#or  
  
path = "test.txt"  
try:  
 os.remove(path)  
 print("File removed")  
except FileNotFoundError:  
 print("File not found")

**Modules**