***SQLite download link***

[***https://sqlitebrowser.org/dl/***](https://sqlitebrowser.org/dl/)

***Core Python***

Print

Math

Comparison (Bolean logic)

External Input (Int, Float, String)

**List**

a = [10, 20, 30, 40, 40, 60]

print (a)

OUTPUT

[10, 20, 30, 40, 40, 60]

**INSERT**

thislist = ["apple", "banana", "cherry"]

thislist.insert(3, "watermelon")

print(thislist)

OUTPUT

['apple', 'banana', 'cherry', 'watermelon']

**.APPEND**

thislist = ["apple", "banana", "cherry"]

thislist.append("orange")

print(thislist)

OUTPUT

['apple', 'banana', 'cherry', 'orange']

**EXTEND**

thislist = ["apple", "banana", "cherry"]

tropical = ["mango", "pineapple", "papaya"]

thislist.extend(tropical)

print(thislist)

OUTPUT

['apple', 'banana', 'cherry', 'mango', 'pineapple', 'papaya']

**POP**

thislist = ["apple", "banana", "cherry"]

thislist.pop(1)

print(thislist)

OUTPUT

['apple', 'cherry']

**SORT**

1. thislist = [9,10,11,89,0,2,3,4,]

thislist.sort()

print(thislist)

OUTPUT

[0, 2, 3, 4, 9, 10, 11, 89]

**2)** thislist = ["orange", "mango", "pineapple","kiwi", "pineapple", "banana"]

thislist.sort()

print(thislist)

**LENGTH**

company =["kolte-patil develope","apocotex industeries ltd","genius power infra","raills India","mcx","mahindra auto"]

print(len(company))

print(company)

OUTPUT

6

['kolte-patil develope', 'apocotex industeries ltd', 'genius power infra', 'raills India', 'mcx', 'mahindra auto']

**REMOVE**

prevclose=[262.50,432.20,84.75,241.50,1509.35,356.70]

prevclose.remove(1509.35)

print(prevclose)

OUTPUT

[262.5, 432.2, 84.75, 241.5, 356.7]

**Reverse**

1)

thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]

print(thislist[2:5])

OUTPUT

['cherry', 'orange', 'kiwi']

2)

thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]

print(thislist[:4])

OUTPUT

['apple', 'banana', 'cherry', 'orange']

3)

thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]

print(thislist[2:])

OUTPUT

4)

fruits = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]

f1 = fruits [-1]

f2 = fruits [-2]

f3 = fruits [-3]

f4 = fruits [-4]

f5 = fruits [-5]

f6 = fruits [-6]

f7 = fruits [-7]

print ("\n Reverse first fruit = ",f1)

print ("\n Reverse second fruit = ",f2)

print ("\n Reverse third fruit = ",f3)

print ("\n Reverse fourth fruit = ",f4)

print ("\n Reverse fifth fruit = ",f5)

print ("\n Reverse sixth fruit = ",f6)

print ("\n Reverse seventh fruit = ",f7)

**Numpy**

**VERSION**

import numpy as np

print(np.\_\_version\_\_)

OUTPUT

1.26.4

**PRINT**

1) import numpy as np

a = np.array([1, 2, 3, 4, 5])

print(a)

OUTPUT

[1 2 3 4 5]

2) import numpy as np

a = np.array([1, 2, 3, 4, 5])

print(a)

import numpy as np

arr = np.array(["a", "b", "c", "d", "e"])

print(arr)

OUTPUT

[1 2 3 4 5]

['a' 'b' 'c' 'd' 'e']

**ARRAY FORMATION**

import numpy as np

a = np.array([1,2,3,4,5,6])

b = np.array([2])

c = a+b;

print(a)

print(b)

print(c)

**FORMING ARRAY FROM AN ARRAY**

import numpy as np

arr = np.array([0,1, 2, 3, 4])

a=(arr[0])

b=(arr[1])

c=(arr[2])

d=(arr[3])

e=(arr[4])

x=np.array([e,a, c, b, b])

y=np.array([a,a, a, a, a])

z=np.array([b,a, b, d, a])

print(x)

print(y)

print(z)

OUTPUT

[4 0 2 1 1]

[0 0 0 0 0]

[1 0 1 3 0]

**#NUMPY TO LIST**

import numpy as np

# NumPy array

arr = np.array([1, 1, 1, 2, 3, 4, 5, 5, 6, 7])

# Convert to list

arr\_list = arr.tolist()

print(arr\_list)

OUTPUT

[1, 1, 1, 2, 3, 4, 5, 5, 6, 7]

**ARRAY SEARCH**

#array Search

import numpy as np

arr = np.array([1, 2, 3, 4, 5, 4, 4])

x = np.where(arr == 4)

print(x)

OUTPUT

(array([3, 5, 6], dtype=int64),)

1. #array Search

import numpy as np

arr = np.array([1, 2, 3, 4, 5, 4, 4])

x = np.where(arr == 0)

print(x)

OUTPUT

(array([], dtype=int64),)

**Array Formation using external Input**

import numpy as np

x = int(input("Enter x? "));

y = int(input("Enter y? "));

z = int(input("Enter z? "));

a = np.array([x,y,z])

b = np.array([z,y,x])

c = a\*b;

print(a)

print(b)

print(c)

OUTPUT

Enter x? 30

Enter y? 56

Enter z? 74

[30 56 74]

[74 56 30]

[2220 3136 2220]

**Math**

1. **+,-,\*(multiply),\*\*(power)**

import numpy as np

a = np.array([1, 2, 3, 4, 5])

b = np.array([1, 2,7, 9, 10])

v=a+b

x=a-b

y=a\*b

z=a\*\*b

print(v)

print(x)

print(y)

print(z)

OUTPUT

[ 2 4 10 13 15]

[ 0 0 -4 -5 -5]

[ 1 4 21 36 50]

[ 1 4 2187 262144 9765625]

**1)#%**

import numpy as np

a = np.array([1,2,3,4,5,6])

x=a\*10/100

print(x)

x1=a+x

print(x1)

OUTPUT

[0.1 0.2 0.3 0.4 0.5 0.6]

[1.1 2.2 3.3 4.4 5.5 6.6]

**LESSER GREATER**

import numpy as np

a = np.array([45,46,47,48])

b = np.array([56])

d = a > b

e = a < b

f = a == b

g = a != b

h = a >= b

i = b <= a

print (d)

print(e)

print(f)

print(g)

print(h)

print(i)

OUTPUT

[False False False False]

[ True True True True]

[False False False False]

[ True True True True]

[False False False False]

[False False False False]

**POPULARITY (SYMBLOS LIKE -,+ WILL BE REMOVED)**

import numpy as np

arr = np.array([-1, -2, 1, 2, +3, 4])

newarr = np.absolute(arr)

print(newarr)

OUTPUT

[1 2 1 2 3 4]

**SIZE**

1. import numpy as np

b = np.array(["affra","jasmine"])

x=b.size

print(x)

OUTPUT

2

1. import numpy as np

b = np.array([1, 2, 3, 4, 5])

x=b.size

print(x)

OUTPUT

5

**NEGATIVE**

import numpy as np

arr = np.array([ 20, 30, 40,50,60,70,80,90,])

c=arr[0]

d=arr[1]

e=arr[2]

a=arr[-1]

b=arr[-4]

f=arr[-6]

print(c)

print(d)

print(e)

print(a)

print(b)

print(f)

OUTPUT

20

30

40

90

60

40

**REVERSE**

1) import numpy as np

arr = np.array([ 20, 30, 40,50,60,70,80,90,])

x=arr[2:4]

print(x)

y=arr[1:]

print(y)

z=arr[:4]

print(z)

OUTPUT

[40 50]

[30 40 50 60 70 80 90]

[20 30 40 50]

2) import numpy as np

x = np.array([12, 38,65,87,67])

print("Original array:")

print(x)

print("Reverse array:")

y = x[::-1]# -1 inside colon

print(y)

OUTPUT

Original array:

[12 38 65 87 67]

Reverse array:

[67 87 65 38 12]

1. **import numpy as np**

#Reverse an array

x = np.array([12, 38,65,87,67])

ra= np.flip(x)

print(ra)

OUTPUT

[67 87 65 38 12]

**Replace**

import pandas as pd

df = pd.read\_excel("mark.xlsx")

#df = pd.read\_csv("data.csv")

c=df.columns

print(c)

#replace

replace=df.replace(30, 5)

print(replace)

OUTPUT

Index(['Name', 'Math', 'Phy', 'Che', 'Eng', 'Bio', 'Tamil'], dtype='object')

Name Math Phy Che Eng Bio Tamil

0 Manu 72 72 72 72 5 72

1 Arun 5 5 60 5 5 60

2 Arun 52 52 60 80 52 60

3 Arun 52 52 72 52 5 80

4 Arun 60 60 80 52 91 60

5 Arun 80 80 52 52 72 5

6 Arun 80 80 60 80 72 52

7 Arun 52 52 80 72 72 52

8 Sanjay 72 72 52 72 60 72

9 Sanjay 72 72 72 72 52 52

10 Sanjay 52 52 80 80 52 60

11 Arun 60 60 72 72 60 80

12 Manu 60 60 5 52 52 60

13 Arun 91 91 52 80 91 72

14 Manu 91 91 91 80 60 52

**Comparison**

**Slice (Forward and reverse etc)**

**SORTING**

# sorting a array

import numpy as np

arr = np.array([0, 12, 1, 3, 4, 5, 6])

arr1 = np.sort(arr)

print("After Sorting",arr1)

## Sort NumPy Array in decreasing order

arr2=np.sort(arr)[::-1]

print("After Sorting in decending order",arr2)

OUTPUT

After Sorting [ 0 1 3 4 5 6 12]

After Sorting in decending order [12 6 5 4 3 1 0]

**MIN AND MAX**

#min and max

import numpy as np

arr = np.array([0, 12, 1, 3, 4, 5, 6])

min=np.min([arr])

max=np.max([arr])

print(min)

print(max)

OUTPUT

0

12

**SUM**

import numpy as np

arr = np.array([1, 2, 3, 5, 9])

sum=np.sum([arr])

mean=np.mean([arr])

print(sum)

print(mean)

OUTPUT

20

4.0

**CONCANTENATE**

#Concatenate

import numpy as np

arr1 = np.array([1, 2, 3])

arr2 = np.array([4, 5, 6])

arr = np.concatenate((arr1, arr2))

print(arr)

OUTPUT

[1 2 3 4 5 6]

Average

Ratio

**ROUND OFF**

#Round off

import numpy as np

arr = np.array([41.25, 42.32, 43.25, 44.9])

arr1 = np.around(arr, 1)

print(arr1)

#Round off

import numpy as np

arr = np.array([41.25, 42.32, 43.65, 44.9999])

arr1 = np.around(arr, 0)

print(arr1)

OUTPUT

[41.2 42.3 43.2 44.9]

[41. 42. 44. 45.]

**COVERT LIST TO NP ARRAY**

import numpy as np

# list to numpy

a = [10, 20, 30, 40, 40, 60]

# Convert to NumPy array

a\_numpy = np.array(a)

print("List:", a)

print("NumPy Array:", a\_numpy)

OUTPUT

List: [10, 20, 30, 40, 40, 60]

NumPy Array: [10 20 30 40 40 60]

2)

#Numpy to list

import numpy as np

# NumPy array

arr = np.array([1, 1, 1, 2, 3, 4, 5, 5, 6, 7])

# Convert to list

arr\_list = arr.tolist()

print(arr\_list)

OUTPUT

[1, 1, 1, 2, 3, 4, 5, 5, 6, 7]

**CONVERTION**

import numpy as np

arr = np.array([1, 2, 3, 4, 5, 5, 6, 7])

# Convert to float

arr\_float = arr.astype(np.float32)

print("Array as float:", arr\_float)

# Convert to string

arr\_str = arr.astype(str)

print("Array as string:", arr\_str)

OUTPUT

Array as float: [1. 2. 3. 4. 5. 5. 6. 7.]

Array as string: ['1' '2' '3' '4' '5' '5' '6' '7']

2**)# float to int**

import numpy as np

# Create a float array

float\_arr = np.array([1.2, 3.4, 5.6, 7.8])

# Convert to int

int\_arr = float\_arr.astype(np.int32)

print("Array as int:", int\_arr)

OUTPUT

Array as int: [1 3 5 7]

3)# Create a float array

import numpy as np

a= np.array([1.2, 3.4, 5.6, 7.8])

# Convert to string

string\_arr = a.astype(str)

print(string\_arr)

OUTPUT

['1.2' '3.4' '5.6' '7.8']

**HEADING**

import numpy as np

arr = np.array([41.25, 42.9, 43.25, 44])

arr1 = np.ceil([arr])

arr2=np.floor([arr])

print("Ceil=",arr1)

print("Floor=",arr2)

OUTPUT

Ceil= [[42. 43. 44. 44.]]

Floor= [[41. 42. 43. 44.]]

**UNIQUE**

import numpy as np

arr = np.array([1, 1, 1, 2, 3, 4, 5, 5, 6, 7])

x = np.unique(arr)

print(x)

OUTPUT

[1 2 3 4 5 6 7]

***Panda to numpy***

import pandas as pd

import numpy as np

a=np.array([11,99,303,44,55,6])

b=np.array([1,2,3,4,5,6])

c=a+b

d=np.array(["Abi","Bala","Coulins","Dinesh","Elger","Ganesh"])

frame={"Name":d, "A":a, "B":b, "C":c}

g=pd.DataFrame(frame)

print(g)

Read CSV

**Read XL**

**1)**

#read XL

import pandas as pd

df = pd.read\_excel("M\_C\_D.xlsx")

print(df)

OUTPUT

Time Number of Customers

0 09:00:00 10.000000

1 10:00:00 47.692308

2 11:00:00 85.384615

3 12:00:00 123.076923

4 13:00:00 160.769231

5 14:00:00 198.461538

6 15:00:00 236.153846

7 16:00:00 273.846154

8 17:00:00 311.538462

9 18:00:00 349.230769

10 19:00:00 386.923077

11 20:00:00 424.615385

12 21:00:00 462.307692

13 22:00:00 500.000000

(We should save the csv or excel file in the spyder and then we that we can read or write our file.)

1. import pandas as pd

import numpy as np

a=np.array([11,99,303,44,55,6])

b=np.array([1,2,3,4,5,6])

c=a+b

d=np.array(["Abi","Bala","Coulins","Dinesh","Elger","Ganesh"])

frame={"Name":d, "A":a, "B":b, "C":c}

g=pd.DataFrame(frame)

print(g)

1. **ASSIGNMENT IN PANDA**

import pandas as pd

import numpy as np

x=np.array([120,140,160,180,190])

y=np.array([20,40,60,80,90])

z=np.array([125,145,165,185,195])

A=x+y

B=x-y

C=x+z

D=y+z

frame={"X":x, "Y":y, "Z":z, "x+y":A ,"x-y":B,"x+z":C,"y+z":D}

g=pd.DataFrame(frame)

print(g)

g.to\_excel('output1.xlsx', index=False)

g.to\_csv('file1.csv', index=False)

**OUTPUT**

X Y Z x+y x-y x+z y+z

0 120 20 125 140 100 245 145

1 140 40 145 180 100 285 185

2 160 60 165 220 100 325 225

3 180 80 185 260 100 365 265

4 190 90 195 280 100 385 285

Write CSV or XLSX

Form a frame

**Info,size,shape**

import pandas as pd

#df = pd.read\_excel("data.csv")

df = pd.read\_csv("data.csv")

print(df)

c=df.columns

print(c)

print("")

information=df.info()

print(information)

print("")

size=df.size

print(size)

print("")

shape=df.shape

print(shape)

OUTPUT

wdir='C:/Users/affraabu/Pd')

Age Experience Rank Nationality Go

0 36 10 9 UK NO

1 42 12 4 USA NO

2 23 4 6 N NO

3 52 4 4 USA NO

4 43 21 8 USA YES

5 44 14 5 UK NO

6 66 3 7 N YES

7 35 14 9 UK YES

8 52 13 7 N YES

9 35 5 9 N YES

10 24 3 5 USA NO

11 18 3 7 UK YES

Index(['Age', 'Experience', 'Rank', 'Nationality', 'Go'], dtype='object')

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 12 entries, 0 to 11

Data columns (total 5 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Age 12 non-null int64

1 Experience 12 non-null int64

2 Rank 12 non-null int64

3 Nationality 12 non-null object

4 Go 12 non-null object

dtypes: int64(3), object(2)

memory usage: 612.0+ bytes

None

60

(12, 5)

**Head**

import pandas as pd

df =pd.read\_excel('mark.xlsx')

df.rename(columns={'Math': 'M'}, inplace=True)

h=df.head(1)

print(h)

print("\n")

df.rename(columns={'Math': 'M','Che':'P'}, inplace=True)

h=df.head(1)

print(h)

OUTPUT

Name M Phy Che Eng Bio Tamil

0 Manu 72 72 72 72 30 72

Name M Phy P Eng Bio Tamil

0 Manu 72 72 72 72 30 72

**Tail**

#tail & head

import pandas as pd

df = pd.read\_excel("mark.xlsx")

#df = pd.read\_csv("data.csv")

c=df.columns

print(c)

h=df.head(2)

t=df.tail(2)

print(h)

print(t)

OUTPUT

Index(['Name', 'Math', 'Phy', 'Che', 'Eng', 'Bio', 'Tamil'], dtype='object')

Name Math Phy Che Eng Bio Tamil

0 Manu 72 72 72 72 30 72

1 Arun 30 30 60 30 30 60

Name Math Phy Che Eng Bio Tamil

13 Arun 91 91 52 80 91 72

14 Manu 91 91 91 80 60 5

**Location**

#loc

import pandas as pd

df = pd.read\_excel("mark.xlsx")

#df = pd.read\_csv("data.csv")

c=df.columns

print(c)

location =df.loc[[0, 1]]

print(location)

print("")

location1 =df.loc[[5,0, 1]]

print(location1)

OUTPUT

Index(['Name', 'Math', 'Phy', 'Che', 'Eng', 'Bio', 'Tamil'], dtype='object')

Name Math Phy Che Eng Bio Tamil

0 Manu 72 72 72 72 30 72

1 Arun 30 30 60 30 30 60

Name Math Phy Che Eng Bio Tamil

5 Arun 80 80 52 52 72 30

0 Manu 72 72 72 72 30 72

1 Arun 30 30 60 30 30 60

**I Location**

(#i loc)

import pandas as pd

df = pd.read\_excel("mark.xlsx")

#df = pd.read\_csv("data.csv")

c=df.columns

print(c)

selected\_rows1 = df.iloc[:4]

print(selected\_rows1)

print("")

selected\_rows2 = df.iloc[10:]

print(selected\_rows2)

OUTPUT

Index(['Name', 'Math', 'Phy', 'Che', 'Eng', 'Bio', 'Tamil'], dtype='object')

Name Math Phy Che Eng Bio Tamil

0 Manu 72 72 72 72 30 72

1 Arun 30 30 60 30 30 60

2 Arun 52 52 60 80 52 60

3 Arun 52 52 72 52 30 80

Name Math Phy Che Eng Bio Tamil

10 Sanjay 52 52 80 80 52 60

11 Arun 60 60 72 72 60 80

12 Manu 60 60 30 52 52 60

13 Arun 91 91 52 80 91 72

14 Manu 91 91 91 80 60 52

**I Location**

#i location

import pandas as pd

df = pd.read\_excel("mark.xlsx")

#df = pd.read\_csv("data.csv")

c=df.columns

print(c)

selected\_rows = df.iloc[1:4]

print(selected\_rows)

OUTPUT

Index(['Name', 'Math', 'Phy', 'Che', 'Eng', 'Bio', 'Tamil'], dtype='object')

Name Math Phy Che Eng Bio Tamil

1 Arun 30 30 60 30 30 60

2 Arun 52 52 60 80 52 60

3 Arun 52 52 72 52 30 80

**COLUMN**

**(Seprate column)**

import pandas as pd

#df = pd.read\_excel("data.csv")

df = pd.read\_csv("data.csv")

c=df.columns

print(c)

col =df[["Age","Experience"]]

print(col)

col =df[["Age","Rank"]]

print(col)

col =df[["Age","Nationality"]]

print(col)

col =df[["Age","Go"]]

OUTPUT

Index(['Age', 'Experience', 'Rank', 'Nationality', 'Go'], dtype='object')

Age Experience

0 36 10

1 42 12

2 23 4

3 52 4

4 43 21

5 44 14

6 66 3

7 35 14

8 52 13

9 35 5

10 24 3

11 18 3

Age Rank

0 36 9

1 42 4

2 23 6

3 52 4

4 43 8

5 44 5

6 66 7

7 35 9

8 52 7

9 35 9

10 24 5

11 18 7

Age Nationality

0 36 UK

1 42 USA

2 23 N

3 52 USA

4 43 USA

5 44 UK

6 66 N

7 35 UK

8 52 N

9 35 N

10 24 USA

11 18 UK

Age Go

0 36 NO

1 42 NO

2 23 NO

3 52 NO

4 43 YES

5 44 NO

6 66 YES

7 35 YES

8 52 YES

9 35 YES

10 24 NO

11 18 YES

Sorting

**Query**

#query

import pandas as pd

df =pd.read\_excel('mark.xlsx')

q1=df.query('Math > 90')

print(" Math > 90\n",q1)

print("/n")

q2=df.query('Math > 90 and Phy >90')

print(q2)

print("/n")

q3=df.query('Math > 90 and Phy >90 and Eng >90')

print(q3)

q5=df.query('Math == 90 or Phy ==90 or Eng ==90 or Che==90')

print(q5)

print("/n")

q5=df.query('Math == 91')

print(q5)

OUTPUT

Math > 90

Name Math Phy Che Eng Bio Tamil

13 Arun 91 91 52 80 91 72

14 Manu 91 91 91 80 60 52

/n

Name Math Phy Che Eng Bio Tamil

13 Arun 91 91 52 80 91 72

14 Manu 91 91 91 80 60 52

/n

Empty DataFrame

Columns: [Name, Math, Phy, Che, Eng, Bio, Tamil]

Index: []

Empty DataFrame

Columns: [Name, Math, Phy, Che, Eng, Bio, Tamil]

Index: []

/n

Name Math Phy Che Eng Bio Tamil

13 Arun 91 91 52 80 91 72

14 Manu 91 91 91 80 60 52

Min

Max

**Average**

import pandas as pd

df = pd.read\_excel("mark.xlsx")

#df = pd.read\_csv("data.csv")

c=df.columns

print(c)

#mean(averge)

import pandas as pd

df =pd.read\_excel('mark.xlsx')

df1 = df[["Bio","Phy","Math","Tamil","Eng"]]

m=df1.mean()

print(m)

OUTPUT

Bio 58.400000

Phy 65.066667

Math 65.066667

Tamil 60.933333

Eng 66.533333

dtype: float64

**Sum**

import pandas as pd

df = pd.read\_excel("mark.xlsx")

#df = pd.read\_csv("data.csv")

c=df.columns

print(c)

#sum

import pandas as pd

df =pd.read\_excel('mark.xlsx')

df1 = df[["Bio","Phy","Math","Tamil","Eng"]]

s=df1.sum()

print(s)

OUTPUT

Index(['Name', 'Math', 'Phy', 'Che', 'Eng', 'Bio', 'Tamil'], dtype='object')

Bio 876

Phy 976

Math 976

Tamil 914

Eng 998

dtype: int64

Frame to numpy

Numpy to frame

DB Syntax:

CSV TO DB-

XL To DB-

**READ DB**

# read DB  
import pandas as pd  
import sqlite3 as db  
con = db.connect('mark.sqlite')  
df = pd.read\_sql\_query("SELECT \* FROM Users", con)  
print(df)  
con.commit()  
con.close()

**DB To SQLITE**

import pandas as pd

import sqlite3 as db

df= pd.read\_excel('sales11.xlsx')

database = "sales11.sqlite"

conn = sqlite3.connect(database)

df.to\_sql(name='Users',con=conn,if\_exists='replace',index=False)

conn.close()

**SQLITE Query :**

SELECT \* FROM Users;

SELECT Name FROM Users;

SELECT Name,Math From Users;

SELECT Name,Math From Users order by Math ASC;

SELECT Name,Math From Users order by Math DESC;

SELECT \* FROM Users WHERE Math> 90;

SELECT \* FROM Users WHERE Math> 90 and Phy>90;

SELECT \* FROM Users WHERE Math> 90 or Phy>90;

SELECT \* FROM Users WHERE Math IN (71, 91);

SELECT \* FROM Users WHERE Math BETWEEN 40 AND 80;

SELECT \* FROM Users WHERE Math NOT BETWEEN 40 AND 80;

SELECT Math, Phy, (Math + Phy) FROM Users;

SELECT Math, Phy, (Math - Phy) FROM Users;

SELECT Math, Phy, Tamil,(Math - Phy + Tamil) FROM Users;

SELECT min(Math) FROM Users;

SELECT max(Math) FROM Users;

SELECT avg(Math) FROM Users;

SELECT count(Math) FROM Users;

SELECT sum(Math) FROM Users;

SELECT \*, min(Math) FROM Users;

SELECT \*, min(Math) FROM Users;

SELECT \*, min(Math) FROM Users;

SELECT \* FROM users GROUP BY Name;

**DB to Panda –**

import pandas as pd

import sqlite3 as db

con = db.connect('m.sqlite')

df = pd.read\_sql\_query("SELECT\* FROM Users WHERE Math NOT BETWEEN 40 and 50", con)

print(df)

con.commit()

con.close()

**Plot**

**Line, scatter, Area and bar plot from data within program**

# read by default 1st sheet of an excel file  
df = pd.read\_excel('plot.xlsx')  
df.plot(kind = 'line', x= 'A', y= 'B', color = 'red')  
# set the title  
plot.title('Line')  
plot.savefig('line.pdf')  
# show the plot  
plot.show()

#change kind =Line as Scatter….

**Colour changing**

Matplotlib provides a variety of predefined colors that you can use in your plots.  
 Here is a list of some of the common color names that you can use in your code:  
'b' - blue  
'g' - green  
'r' - red  
'c' - cyan  
'm' - magenta  
'y' - yellow  
'k' - black  
'w' - white  
You can also use hexadecimal color codes to specify a custom color. For example,  
'#FF0000' represents pure red, '#00FF00' represents pure green, and '#0000FF'  
represents pure blue.  
You can pass the color name or hexadecimal color code as the value of the  
color parameter in the plt.plot() function to change the color of the line.

Others

Graph without giving data

scatter  
from matplotlib import pyplot as plot  
x = [10, 20, 30,40,50,60]  
y = [100, 200, 300,400,500,600]  
plot.scatter(x, y)  
plot.title('s graph')  
plot.ylabel('Y axis')  
plot.xlabel('X axis')  
plot.show()

ALGORITHIM

**If**

a = 3000  
b = 4000  
if b > a:  
  print("b is greater than a")

#space must

OUTPUT

b is greater than a

Elif

#ELSE IF  
a = 433  
b = 433  
if b > a:  
  print("b is greater than a")  
elif a == b:  
  print("a and b are equal")

OUTPUT

a and b are equal

#ELSE IF  
a = 4331  
b = 4332  
if b > a:  
  print("b is greater than a")  
elif a == b:  
  print("a and b are equal")

OUTPUT

b is greater than a

#ELSE IF  
a = 4336  
b = 4332  
if b > a:  
  print("b is greater than a")  
elif a == b:  
  print("a and b are equal")

OUTPUT

#no output wll appear(no condition available)

**Else**

a = 200  
b = 33  
if b > a:  
  print("b is greater than a")  
elif a == b:  
  print("a and b are equal")  
else:  
  print("else ")

OUTPUT

Else(#if there is false function else will appear)

a = 2

b = 2

c = 2

if a > b and a > c:

print("a is greater")

elif a == b and a == c:

print("all are equal")

elif b > a and b >c:

print("b is greater")

else:

print("else")

OUTPUT

all are equal

(#it will read each and every condition and give the output of the true material)

2)

# Define user ID and password

user\_id = "bala"

password = "1234"

# Get user input for user ID and password

input\_user\_id = input("Enter user ID: ")

input\_password = input("Enter password: ")

# Check if the user ID and password are correct

if input\_user\_id == user\_id and input\_password == password:

# Give access to the module

print("Access granted. Welcome to the module!")

else:

# Deny access

print("Access denied. Incorrect user ID or password.")

OUTPUT

Enter user ID: 23

Enter password: affra

Access denied. Incorrect user ID or password.

3)

score1 = int(input("Enter score1: "))  
if score1 > 90:  
    print("Student grade is A")  
elif score1 >= 80 and score1 <= 89:  
    print("Student grade is B")  
elif score1 >= 70 and score1 <= 79:  
    print("Student grade is C")  
elif score1 >= 60 and score1 <= 69:  
    print("Student grade is D")  
else:  
    print("Student grade is F")

OUTPUT

Enter score1: 2

Student grade is F

TASK

1)

#Below 18.5: "Underweight"

#18.5 to 24.9: "Normal weight"

#25 to 29.9: "Overweight"

#30 or above: "Obese"

weight = int(input("Enter weight: "))

if weight < 18.5:

print("under weight")

elif weight >= 18.5 and weight <= 24.9:

print("normal weight")

elif weight>= 25 and weight <= 29.9:

print("over weight")

elif weight >= 30 and weight <= 59:

print("obese")

else:

print("human")

OUTPUT  
Enter weight: 24

normal weight

2)

#0 to 2 years: "Infant"

#3 to 12 years: "Child"

#13 to 19 years: "Teenager"

#20 to 59 years: "Adult"

#60 years and above: "Senior Citizen"

category = int(input("Enter category: "))

if category < 2:

print("infant")

elif category >= 3 and category <= 12:

print("child")

elif category >= 13 and category <= 19:

print("teenager")

elif category >= 20 and category <= 59:

print("adult")

elif category >= 60 and category <=79 :

print("senior citizen")

else:

print("human")

OUTPUT

Enter category: 45

Adult

Task 3) select any 5 from the task in list and use them in eles if concept as password way while giving the password the task should open on the excel

**Base code**

import pandas as pd

c= input("Enter employee for file Employee,feedback for file Feedback.xlsx.....")

if c=='employee' or c=='Employee' or c=='EMPLOYEE' or c=='emp' or c=='employ'or c=='worker':

df1=pd.read\_excel("Employee\_Attendance(1).xlsx")

print(df1)

elif c=='feedback' or c=='Feedback' or c=='FEEDBACK' or c=='feed' or c=='Feedb':

df2=pd.read\_excel("Feedback.xlsx")

print(df2)

elif c=='incometax' or c=='Incometax' or c=='INCOMETAX' or c=='income' or c=='tax' or c=='Tax':

df3=pd.read\_excel("Incometax.xlsx")

print(df3)

elif c=='salary' or c== 'SALARY' or c=='Salary' or c=='sal' :

df4=pd.read\_excel("salary.xlsx")

print(df4)

elif c=='bank' or c=='Bank' or c=='BANK' :

df4=pd.read\_excel("bank\_data.xlsx")

print(df4)

elif c=='output' or c == 'OUTPUT' or c=='Output' or c=='Out' or c=='out' :

df4=pd.read\_excel("output.xlsx")

print(df4)

else:

print("File not found ")

**Function**

def add():

x=2

y=4

c=x+y

print("add=",c)

def sub():

x=2

y=4

c1=x-y

print("sub=",c1)

z=add()

z1=sub()

print(z)

OUTPUT

add= 6

sub= -2

None

2)

**#creat and call**

def add():

x=2

y=4

c=x+y

print("add=",c)

def sub():

x=2

y=4

c1=x-y

print("sub=",c1)

def div():

x=2499

y=22.9

c2=x/y

print("div=",c2)

def multiply():

x=143

y=12

c3=3\*y

print("multiply=",c3)

def power():

x=3

c4=3\*\*x

print("power=",c4)

z=add()

z1=sub()

z2=div()

z3=multiply()

z4=power()

print(z)

OUTPUT

add= 6

sub= -2

div= 109.12663755458516

multiply= 36

power= 27

None

**3)return**

def add():

x=2

y=4

c=x+y

print("add=",c)

return c

def sub():

x=2

y=4

c1=x-y

print("sub=",c1)

return c1

def div():

x=2499

y=22.9

c2=x/y

print("div=",c2)

return c2

def multiply():

x=1437

y=23

c3=x\*y

print("multiply=",c3)

return c3

def power():

x=5

c4=5\*\*x

print("power=",c4)

return c4

z=add()

print(z)

z1=sub()

print(z1)

z2=div()

print(z2)

z3=multiply()

print(z3)

z4=power()

print(z4)

OUTPUT

add= 6

6

sub= -2

-2

div= 109.12663755458516

109.12663755458516

multiply= 33051

33051

power= 3125

3125

**4)#return function**

import pandas as pd  
def file1():  
    df1=pd.read\_excel("xlrandom1.xlsx")  
    print(df1)  
    return df1  
def file2():  
    df2=pd.read\_excel("xlrandom2.xlsx")  
    print(df2)  
    return df2  
def file3():  
    df3=pd.read\_excel("xlrandom3.xlsx")  
    print(df3)  
    return df3  
def file4():  
    df4=pd.read\_excel("xlrandom4.xlsx")  
    print(df4)  
    return df4  
c= input("Enter the file you want to read")  
if c=='1' or c=='file1' or c=='f1':  
   a=file1()  
elif c=='2' or c=='file2'or c=='f2':  
   b=file2()  
elif c=='3' or c=='file3'or c=='f3':  
   c=file3()  
elif c=='4' or c=='file4'or c=='f4':  
   d=file4()  
else:  
    print("File not found ")

OUTPUT

**for**

for x in range(5): #start at 0 end before 5 increment by 1

print("hi")

**5)#function using userid**

import pandas as pd

user\_id = "abcd"

password = "1234"

def add():

x=2

y=4

c=x+y

print("add=",c)

return c

def sub():

x=2

y=4

c1=x-y

print("sub=",c1)

return c1

def file1():

df1=pd.read\_excel("Feedback.xlsx")

print(df1)

return df1

# Get user input for user ID and password

input\_user\_id = input("Enter user ID: ")

input\_password = input("Enter password: ")

# Check if the user ID and password are correct

if input\_user\_id == user\_id and input\_password == password:

z=add()

print(z)

z1=sub()

print(z1)

c= input("Enter the file you want to read")

if c=='1' or c=='file1'or c=='f1':

b=file1()

else:

# Deny access

print("Access denied. Incorrect user ID or password.")

hi

hi

hi

hi

hi

2)

for x in range(5): #start at 0 end before 5 increment by 1

a=int(input("Enter salary"))

print("salary=",a)

print(x)

OUTPUT

Enter salary3000

salary= 3000

0

Enter salary3000

salary= 3000

1

Enter salary4000

salary= 4000

2

Enter salary6000

salary= 6000

3

Enter salary7000

salary= 7000

4

3)

# Prints out 3,4,5  
for x in range(3, 6):  #start at 3 end before 6 increment by 1  
    print(x)

OUTPUT

3

4

5

4)

# Prints out 3,5  
for x in range(3, 100, 2):   #start at 3 end before 7 increment by 2  
    print(x)

OUTPUT

#Appear on 3 till 99

1. *# to show number of file at a time*

*import pandas as pd*

for i in range(10):

c= input("Enter the 1 for file 1 ,2 for file 2...")

if c=='1' or c=='file1' or c=='f1':

df1=pd.read\_excel("xlrandom1.xlsx")

print(df1)

elif c=='2' or c=='file2'or c=='f2':

df2=pd.read\_excel("xlrandom2.xlsx")

print(df2)

elif c=='3' or c=='file3'or c=='f3':

df3=pd.read\_excel("xlrandom3.xlsx")

print(df3)

elif c=='4' or c=='file4'or c=='f4':

df4=pd.read\_excel("xlrandom4.xlsx")

print(df4)

else:

print("File not found ")

**try except used in last**

import pandas as pd  
df=pd.read\_csv("file.csv")  
print(df)

import pandas as pd  
try:  
    df=pd.read\_csv("file.csv")  
  
except:  
    print ("no file in location ")

**while**

# Initialize a variable

number = 1

while number <= 10:

print(number)

number = number+1

#error sim

number = 1

while number <= 10:

print(number)

**while break statement**

age = 21

# the test condition is always True

while age > 18:

print('You can vote')

break

age = 120

# the test condition is always True

while age > 18:

print('You can vote')

break

else:

print("you are not eligible to vote")

**while true**

The statement while True: in Python creates an infinite loop. This means the block  
of code inside the loop will keep running indefinitely unless it is interrupted by a  
 break statement or an external event  
like the program being manually stopped

while True:  
      a1 =input("user id =")  
      a2=input("password =")  
  
  
      while a1=="abcd" and a2=="1234":  
        print("Welcome")  
        break  
      else:  
        print("Authorization denied")

**filter**

a = ["what", "how", "why","is","in","your"]#only lower case  
  
  
ip= input("Enter your sentence: ")  
l=ip.lower()  
print("\ninput in lower case =",l)  
# Split the input string into individual words  
b = l.split()  
print("\ninput in list Format=",b)  
  
# Create a new list without the words to remove  
filter = [word for word in b if word.lower() not in a]  
print("\nfiltered words=",filter)  
# Join the filtered words back together  
result = " ".join(filter)  
print("\nKey word=", result)

**JOIN**

b = ['Hi', 'how', 'are', 'you', 'James']  
c = " ".join(b)  
print(c)

OUTPUT

Hi how are you james

**Spilt**

ip= input("Enter your sentence: ")  
  
# Split the input string into individual words  
b = ip.split()  
print(b)

**filter**

a = ['apple', 'banana', 'cherry']   # list of words to exclude (case insensitive)  
b = ['Apple', 'orange', 'Banana', 'grape', 'Cherry']  
filtered\_words = [word for word in b if word.lower() not in a]  
print(filtered\_words)

OUTPUT

['orange', 'grape']

**Low , upper , title**

a=" HI HOW ARE YOU"  
b=a.lower()  
print(b)

a=" good morning"  
b=a.upper()  
print(b)

a=" HI HOW ARE YOU"  
b=a.title()  
print(b)

ip=input("Enter =")  
a=ip.lower()  
b=ip.upper()  
c=ip.title()  
print("Lower Format =",a)  
print("Upper Format=",b)  
print("Title Format=",c)