# DATE:

**PROGRAM NO:** 08

**AIM:** creating 1d array and performing arithmetic operators , slicing , joining and all aggregate function on it. Create 2–d array and perform slicing and joining.

# ALGORITHM:

Step 1 - start the process. Step 2 – import numpy.

Step 3 – create two 1-d arrays b, b1.

Step 4 – perform arithmetic operations on 1– d array. Step 5 – slice the 1 – d array a[start , stop].

Step 6 – join 1–d array np.hstack(( b,b1)), np.vstack((b,b1)). Step 7 – perform aggregate functions on 1–d array.

Step 8 – create two 2–d array.

Step 9 –perform slicing on t[:3,3:] and joining on np.vstack((t,v)) then print them.

Step 10 – stop the process.

# CODE:

**import** numpy **as** np a=[10,20,30,40,50,60]

b=np.array(a) print(b)

print(**"arithmetic operations on 1-d array(scalar)"**) c=np.add(b,10)

d=np.subtract(b,12) e=np.multiply(b,2) f=np.divide(b,2) g=np.remainder(b,4) print(c)

print(d) print(e) print(f) print(g)

a1=[1,2,3,4,5,6]

b1=np.array(a1)

print(**"arithmetic operations on 1-d array(another ndarray)"**) h=np.add(b,b1)

i=np.subtract(b,b1) j=np.multiply(b,b1) k=np.divide(b,b1) l=np.remainder(b,b1) print(b1)

print(h) print(i) print(j) print(k) print(l)

print(**"slicing on 1-d array"**) m=b[2:5]

print(m)

print(**"joining on 1-d array"**) n=np.hstack((b,b1))

o=np.vstack((b,b1)) print(n)

print(o)

print(**"aggreate function on 1-d array"**) p=np.mean(b)

q=np.median(b) r=np.max(b) s=np.min(b) print(p)

print(q) print(r) print(s)

list=[[2,4,6,8,10],[12,14,16,18,20],[22,24,26,28,30],[32,34,36,38,40]]

t=np.array(list) print(**"first 2-d array"**) print(t)

print(**"\n"**)

u=t[:3,3:]

print(**"slicing of 2-d array"**) print(u)

print(**"\n"**) list1=[[42,44,46,48,50],[52,54,56,58,60],[62,64,66,68,70],[72,74,76,78,80]]

v=np.array(list1) print(**"second 2-d array"**) print(v)

print(**"\n"**) w=np.vstack((t,v)) print(**"joining 2-d array"**) print(w)

# OUTPUT:

[10 20 30 40 50 60]

arithmetic operations on 1-d array(scalar) [20 30 40 50 60 70]

[-2 8 18 28 38 48]

[ 20 40 60 80 100 120]

[ 5. 10. 15. 20. 25. 30.]

[2 0 2 0 2 0]

arithematic operations on 1-d array(another ndarray) [1 2 3 4 5 6]

[11 22 33 44 55 66]

[ 9 18 27 36 45 54]

[ 10 40 90 160 250 360]

[10. 10. 10. 10. 10. 10.]

[0 0 0 0 0 0]

slicing on 1-d array [30 40 50]

joining on 1-d array

[10 20 30 40 50 60 1 2 3 4 5 6]

[[10 20 30 40 50 60]

[ 1 2 3 4 5 6]]

aggreate functiion on 1-d array 35.0

35.0

60

10

first 2-d array [[ 2 4 6 8 10]

[12 14 16 18 20]

[22 24 26 28 30]

[32 34 36 38 40]]

slicing of 2-d array [[ 8 10]

[18 20]

[28 30]]

second 2-d array [[42 44 46 48 50]

[52 54 56 58 60]

[62 64 66 68 70]

[72 74 76 78 80]]

joining 2-d array

[[ 2 4 6 8 10]

[12 14 16 18 20]

[22 24 26 28 30]

[32 34 36 38 40]

[42 44 46 48 50]

[52 54 56 58 60]

[62 64 66 68 70]

[72 74 76 78 80]]

# RESULT:

The above program has been executed successfully and the output is verified.

# DATE:

**PROGRAM NO:** 18

**AIM:** creating mysql connecting with python and use insert statement.

# ALGORITHM:

Step 1 - start the process.

Step 2 – import mysql.connector and error.

Step 3 – establish a link between mysql and python.

Step 4 – give the query **insert into speaker values(2,'class','2003-03-08');.**

Step 5 – commit the database. Step 6 – stop the process.

# CODE:

**import** mysql.connector

**from** mysql.connector **import** Error

**try**:

con=mysql.connector.connect(host=**'localhost'**,database=**'tom'**,user=**'root'**,pass word=**'12345'**)

s=**"insert into speaker values(2,'class','2003-03-08');"**

cur=con.cursor() result=cur.execute(s) con.commit() print(**"done"**)

**except** mysql.connector.Error **as** Error: print(**"failed inserting"**)

**finally**:

**if** (con.is\_connected()): cur.close() con.close()

print(**"mysql connector is closed"**)

# OUTPUT:

done

mysql connector is closed

+------+-------+------------+

| sno | name | dob |

+------+-------+------------+

| 2 | rhea | 2003-03-08 |

| 2 | class | 2003-03-08 |

| 2 | class | 2003-03-08 |

+------+-------+------------+

3 rows in set (0.00 sec)

# RESULT:

The above program has been executed successfully and the output is verified.

# DATE:

**PROGRAM NO:** 19

**AIM:** creating mysql connecting with python and use select statement

# ALGORITHM:

Step 1 - start the process.

Step 2 – import mysql.connector and error.

Step 3 – establish a link between mysql and python. Step 4 – give the query **select \* from speaker;.**

Step 5 – display the database. Step 6 – stop the process.

# CODE:

**import** mysql.connector

**from** mysql.connector **import** Error

**try**:

con=mysql.connector.connect(host=**'localhost'**,database=**'tom'**,user=**'root'**,pass word=**'12345'**)

s=**"select \* from speaker;"** rs=con.cursor() rs.execute(s) rec=rs.fetchall()

print(**"no of rows"**,rs.rowcount) **for** i **in** rec:

print(**"name="**,i[1])

print(**"dob="**,i[1])

**except** Error **as** e:

print(**"error message/exception type"**,e) **finally**:

**if** (con.is\_connected()): con.close()

# OUTPUT:

no of rows 1 name= rhea dob= rhea **RESULT:**

The above program has been executed successfully and the output is verified.

# DATE:

**PROGRAM NO:** 20

**AIM:** creating mysql connecting with python and use update statement.

# ALGORITHM:

Step 1 - start the process.

Step 2 – import mysql.connector and error.

Step 3 – establish a link between mysql and python.

Step 4 – give the query **update speaker set name='rhea' where sno=2;.**

Step 5 – commit the database. Step 6 – stop the process.

# CODE:

**import** mysql.connector

**from** mysql.connector **import** Error

**try**:

con=mysql.connector.connect(host=**'localhost'**,database=**'tom'**,user=**'root'**,pass word=**'12345'**)

s=**"update speaker set name='rhea' where sno=2;"**

cur=con.cursor() result=cur.execute(s) con.commit() print(**"done"**)

**except** mysql.connector.Error **as** Error: print(**"failed inserting"**)

**finally**:

**if** (con.is\_connected()): cur.close() con.close()

print(**"mysql connector is closed"**)

# OUTPUT:

done

mysql connector is closed

+------+------+------------+

| sno | name | dob |

+------+------+------------+

| 2 | rhea | 2003-03-08 |

| 2 | rhea | 2003-03-08 |

| 2 | rhea | 2003-03-08 |

+------+------+------------+

3 rows in set (0.00 sec)

# RESULT:

The above program has been executed successfully and the output is verified.

# DATE:

**PROGRAM NO:** 21

**AIM:** creating mysql connecting with python and use delete statement.

# ALGORITHM:

Step 1 - start the process.

Step 2 – import mysql.connector and error.

Step 3 – establish a link between mysql and python.

Step 4 – give the query **delete from speaker where sno=1;.**

Step 5 – commit the database. Step 6 – stop the process.

# CODE:

**import** mysql.connector

**from** mysql.connector **import** Error

**try**:

con=mysql.connector.connect(host=**'localhost'**,database=**'tom'**,user=**'root'**,pass word=**'12345'**)

s=**"delete from speaker where sno=1;"**

cur=con.cursor() result=cur.execute(s) con.commit() print(**"done"**)

**except** mysql.connector.Error **as** Error: print(**"failed inserting"**)

**finally**:

**if** (con.is\_connected()): cur.close() con.close()

print(**"mysql connector is closed"**)

# OUTPUT:

done

mysql connector is closed

+------+-------+------------+

| sno | name | dob |

+------+-------+------------+

| 2 | rhea | 2003-03-08 |

| 2 | class | 2003-03-08 |

+------+-------+------------+

2 rows in set (0.00 sec)

# RESULT:

The above program has been executed successfully and the output is verified.