771_Tanvi Jadhav.ipynb - Colabo

colab.research.google.com/drive/1od48xWD5ztTZ0ZmxPtZm2fHuHTs2xOP#scrollTo=eUSYjuQKfnAl

771_Tanvi Jadhav.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

Essential of Data Science Lab Assignment No: 3

```
[7] import numpy as np
```

```
[8] array1= np.array([[1,2,3],[4,5,6],[7,8,9]])

    array1
```

```
array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9]])
```

```
array2=np.array([[11,12,13], [14,15,16],[17,18,19]])

array2
```

```
array([[11, 12, 13],
       [14, 15, 16],
       [17, 18, 19]])
```

1 Matrix Addition

1.1 Addition

```
[14] resultarray = array1+array2
     print("\nsing Operator:\n",resultarray)
     resultarray=np.add(array1, array2)
     print("\nusing Numpy Function:\n",resultarray)
```

0s completed at 3:30 PM

28°C Sunny

Search

ENG IN

15:39 10-06-2023

771_Tanvi Jadhav.ipynb - Colabo

colab.research.google.com/drive/1od48xWD5ztTZ0ZmxPtZm2fHuHTs2xOP#scrollTo=eUSYjuQKfnAl

771_Tanvi Jadhav.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

1 Matrix Addition

1.1 Addition

```
resultarray = array1+array2
print("\nsing Operator:\n",resultarray)
resultarray=np.add(array1, array2)
print("nusing Numpy Function: \n",resultarray)
```

```
sing Operator:
 [[12 14 16]
 [18 20 22]
 [24 26 28]]
nusing Numpy Function:
 [[12 14 16]
 [18 20 22]
 [24 26 28]]
```

1.2 Substraction

```
[21] resultarray = array1-array2

     print("\nUsing Operator:\n", resultarray)

     resultarray=np.subtract(array1, array2)

     print("\nUsing Numpy Function: \n", resultarray)
```

0s completed at 3:30 PM

28°C Sunny

Search

ENG IN

15:30 10-06-2023

771_Tanvi Jadhav.ipynb - Colabo

colab.research.google.com/drive/1od48xWD5ztTZ0ZmxPtZm2fHuHTs2xOP#scrollTo=eUSYjuQKfnAl

771_Tanvi Jadhav.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

```
[21] resultarray = array1-array2

     print("\nUsing Operator:\n", resultarray)

     resultarray=np.subtract(array1, array2)

     print("\nUsing Numpy Function: \n", resultarray)
```

```
Using Operator:
 [[-10 -10 -10]
 [-10 -10 -10]
 [-10 -10 -10]]

Using Numpy Function:
 [[-10 -10 -10]
 [-10 -10 -10]
 [-10 -10 -10]]
```

1.3 Multiplication

```
resultarray=array1*array2

print("\nUsing Operator: \n", resultarray)
resultarray=np.multiply(array1, array2)
print("\nUsing Numpy Function: \n",resultarray)
```

```
Using Operator:
 [[ 11  24  39]
 [ 56  75  96]
 [119 144 171]]

Using Numpy Function:
 [[ 11  24  39]
 [ 56  75  96]
```

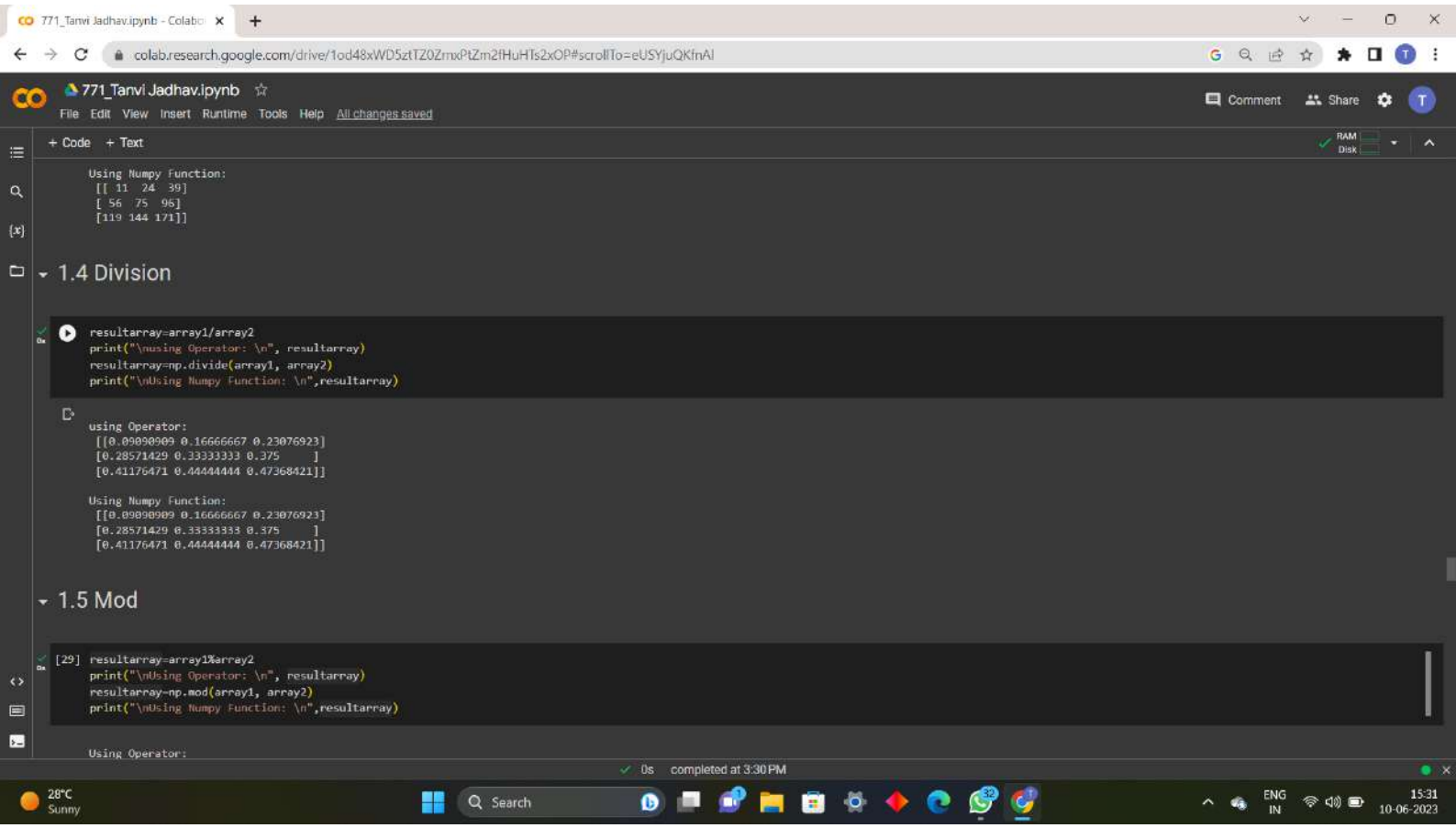
0s completed at 3:30 PM

28°C Sunny

Search

ENG IN

15:30 10-06-2023


771_Tanvi Jadhav.ipynb - Colabo
colab.research.google.com/drive/1od48xWD5ztTZ0ZmxPtZm2fHuHTs2xOP#scrollTo=eUSYjuQKfnAl
771_Tanvi Jadhav.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Using Numpy Function:
[[11 24 39]
[56 75 96]
[119 144 171]]
1.4 Division
resultarray=array1/array2
print("\nusing Operator: \n", resultarray)
resultarray=np.divide(array1, array2)
print("\nUsing Numpy Function: \n",resultarray)
using Operator:
[[0.09090909 0.16666667 0.23076923]
[0.28571429 0.33333333 0.375]
[0.41176471 0.44444444 0.47368421]]
Using Numpy Function:
[[0.09090909 0.16666667 0.23076923]
[0.28571429 0.33333333 0.375]
[0.41176471 0.44444444 0.47368421]]
1.5 Mod
[29] resultarray=array1%array2
print("\nUsing Operator: \n", resultarray)
resultarray=np.mod(array1, array2)
print("\nUsing Numpy Function: \n",resultarray)
Using Operator:
0s completed at 3:30 PM
28°C
Sunny
Search
ENG
IN
15:31
10-06-2023

771_Tanvi Jadhav.ipynb - Colabo

colab.research.google.com/drive/1od48xWD5ztTZ0ZmxPtZm2fHuHTs2xOP#scrollTo=eUSYjuQKfnAl

771_Tanvi Jadhav.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

```
Using Operator:
 [[1 2 3]
 [4 5 6]
 [7 8 9]]

Using Numpy Function:
 [[1 2 3]
 [4 5 6]
 [7 8 9]]
```

1.6 Dot Product

```
[30] resultarray=np.dot(array1, array2)
     print("",resultarray)
```

```
 [[ 90  96 102]
 [216 231 246]
 [342 366 390]]
```

1.7 Transpose

```
[31] resultarray=np.transpose(array1)
     print(resultarray)
     #or
     resultarray=array1.transpose()
     print(resultarray)
```

```
[[1 4 7]
 [2 5 8]
 [3 6 9]]
[[1 4 7]
 [2 5 8]
```

0s completed at 3:30 PM

28°C Sunny

Search

ENG IN

15:31 10-06-2023

771_Tanvi Jadhav.ipynb - Colabo

colab.research.google.com/drive/1od48xWD5ztTZ0ZmxPtZm2fHuHTs2xOP#scrollTo=eUSYjuQKfnAl

771_Tanvi Jadhav.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

2. Horizontal and vertical stacking of Numpy Arrays

2.1. Horizontal Stacking

```
[32] resultarray=np.hstack((array1, array2))
     resultarray
```

```
array([[ 1,  2,  3, 11, 12, 13],
       [ 4,  5,  6, 14, 15, 16],
       [ 7,  8,  9, 17, 18, 19]])
```

2.2. Vertical Stacking

```
resultarray=np.vstack((array1, array2))

resultarray
```

```
array([[ 1,  2,  3],
       [ 4,  5,  6],
       [ 7,  8,  9],
       [11, 12, 13],
       [14, 15, 16],
       [17, 18, 19]])
```

3.Custom sequence generation

0s completed at 3:30 PM

28°C Sunny

Search

ENG IN

15:31 10-06-2023

771_Tanvi Jadhav.ipynb - Colabo

colab.research.google.com/drive/1od48xWD5ztTZ0ZmxPtZm2fHuHTs2xOP#scrollTo=eUSYjuQKfnAl

771_Tanvi Jadhav.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

3.Custom sequence generation

3.1. Range

```
[34] nparray=np.arange(0,12,1).reshape(3,4)

     nparray
```

```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11]])
```

3.2. Linearly Separable

```
nparray=np.linspace(start=0, stop=24,num=12).reshape(3,4)
nparray
```

```
array([[ 0.        ,  2.18181818,  4.36363636,  6.54545455],
       [ 8.72727273, 10.90909091, 13.09090909, 15.27272727],
       [17.45454545, 19.63636364, 21.81818182, 24.        ]])
```

3.3. Empty Array

```
[36]
```

0s completed at 3:30 PM

28°C Sunny

Search

ENG IN

15:31 10-06-2023

771_Tanvi Jadhav.ipynb - Colabo

colab.research.google.com/drive/1od48xWD5ztTZ0ZmxPtZm2fHuHTs2xOP#scrollTo=eUSYjuQKfnAl

771_Tanvi Jadhav.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

3.3. Empty Array

```
[36]
```

```
[36]
```

```
[38] nparray=np.empty((3,3),int)

     nparray
```

```
array([[ 90,  96, 102],
       [216, 231, 246],
       [342, 366, 390]])
```

3.4. Emply Like Some other array

```
nparray=np.empty_like(array1)
nparray
```

```
array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9]])
```

3.5. Identity Matrix

```
[40] nparray= np.identity(3)
     nparray
```

0s completed at 3:30 PM

28°C Sunny

Search

ENG IN

15:31 10-06-2023

771_Tanvi Jadhav.ipynb - Colabo

colab.research.google.com/drive/1od48xWD5ztTZ0ZmxPtZm2fHuHTs2xOP#scrollTo=eUSYjuQKfnAl

771_Tanvi Jadhav.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

3.5. Identity Matrix

```
[40] nparray= np.identity(3)
     nparray
```

```
array([[1., 0., 0.],
       [0., 1., 0.],
       [0., 0., 1.]])
```

4. Arithmetic and Statistical Operations, Mathematical Operations, Bitwise Operators

4.1. Arithmetic Operation

```
array1=np.array([1,2,3,4,3])
array2=np.array([11,12,13,14,15])
print (array1)
print(array2)
```

```
[1 2 3 4 3]
[11 12 13 14 15]
```

```
[43] # Addition
     print(np.add(array1,array2))
     # Subtraction
     print(np.subtract(array1,array2))
     # Multiplication
     print(np.multiply(array1,array2))
     # Division
```

0s completed at 3:30 PM

28°C Sunny

Search

ENG IN

15:31 10-06-2023

771_Tanvi Jadhav.ipynb - Colabo

colab.research.google.com/drive/1od48xWD5ztTZ0ZmxPtZm2fHuHTs2xOP#scrollTo=eUSYjuQKfnAl

771_Tanvi Jadhav.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

4.2. Statistical and Mathematical Operations

```
array1=np.array([1,2,3,4,5,9,6,7,8,9,9])
# Standard Deviation
print(np.std(array1))
#Minimum
print(np.min(array1))
#Summation
print(np.sum(array1))
#Median
print(np.median(array1))
#Mean
print(np.mean(array1))
#Mode
from scipy import stats
print("Most Frequent element-",stats.mode(array1)[0])
print("Number of occarances=",stats.mode(array1)[1])
# Variance
print(np.var(array1))
```

```
2.7990553306073913
1
63
6.0
5.7272727272727275
Most Frequent element- [9]
Number of occarances= [3]
7.834710743801653
<ipython-input-45-5704265416fd>:14: FutureWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, thi
  print("Most Frequent element-",stats.mode(array1)[0])
<ipython-input-45-5704265416fd>:15: FutureWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, thi
  print("Number of occarances=",stats.mode(array1)[1])
```

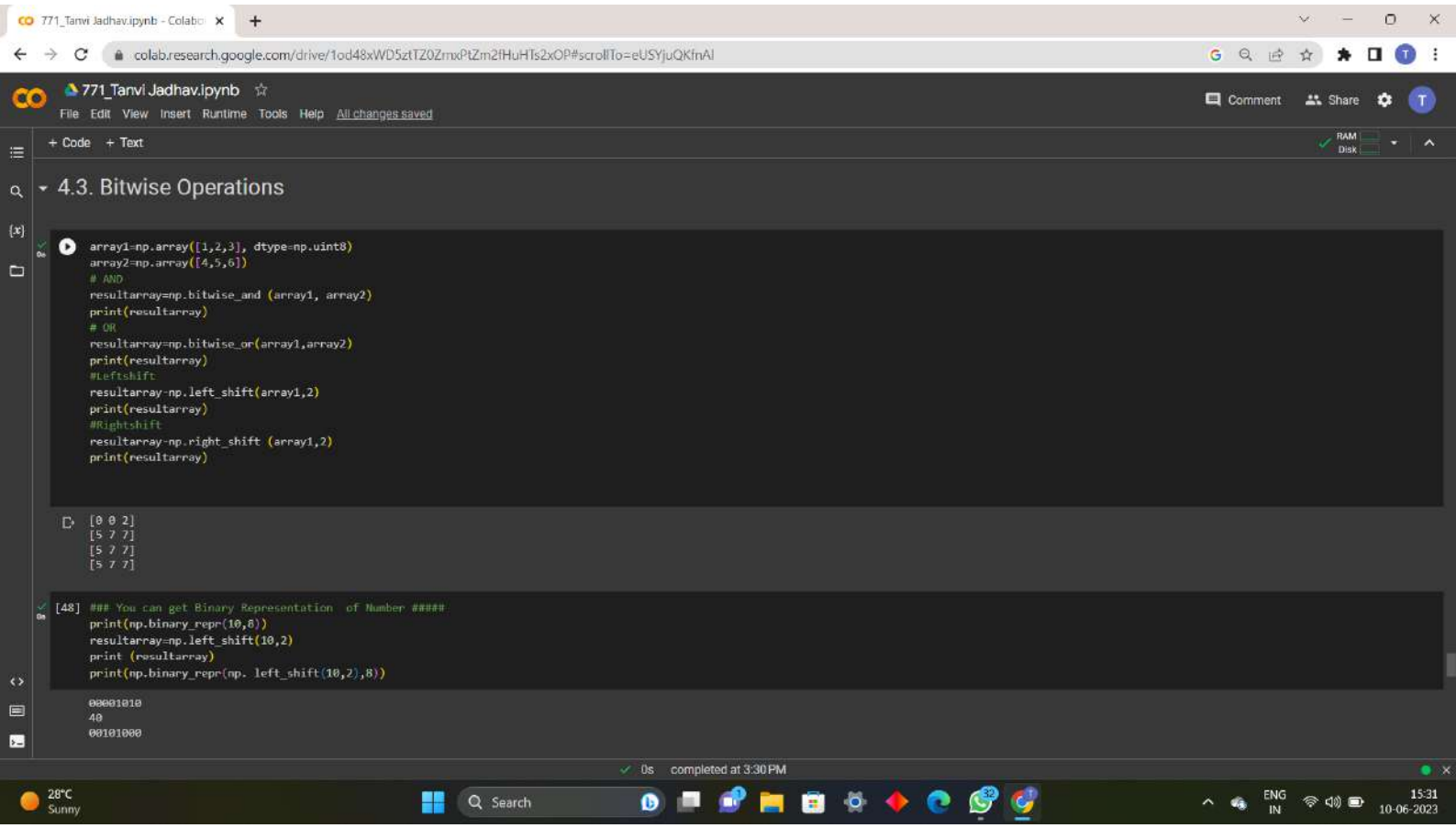
0s completed at 3:30 PM

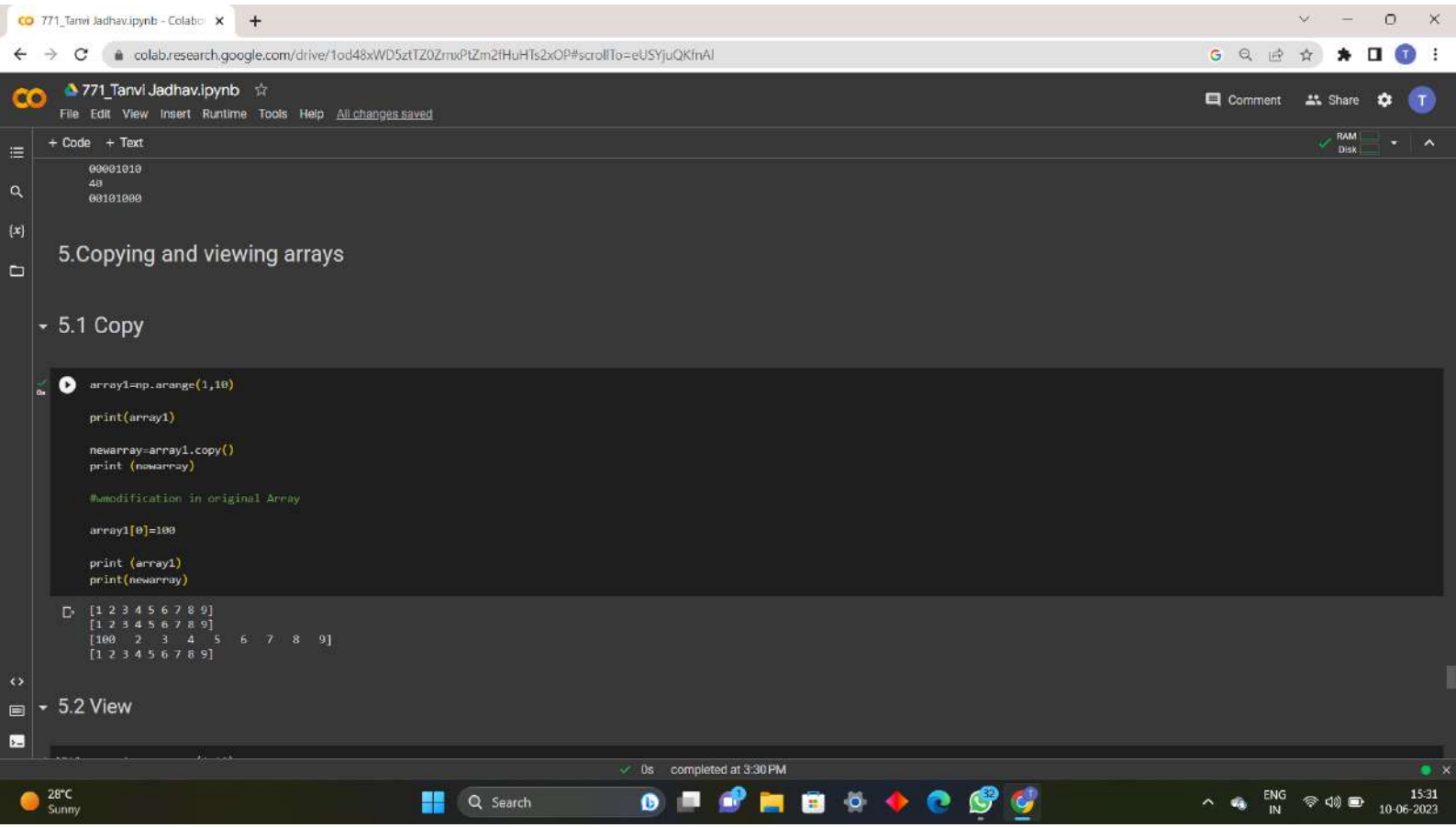
28°C Sunny

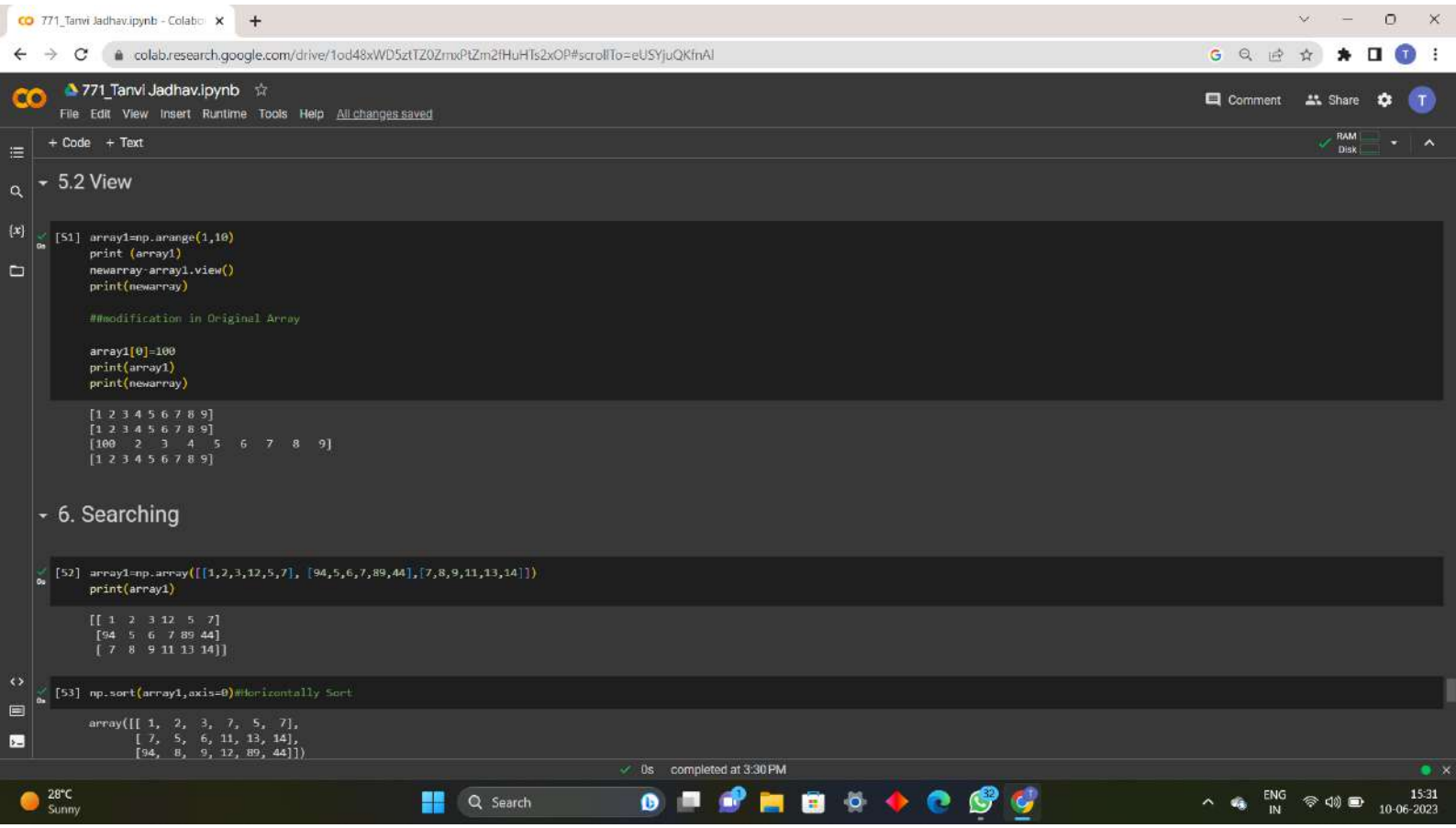
Search

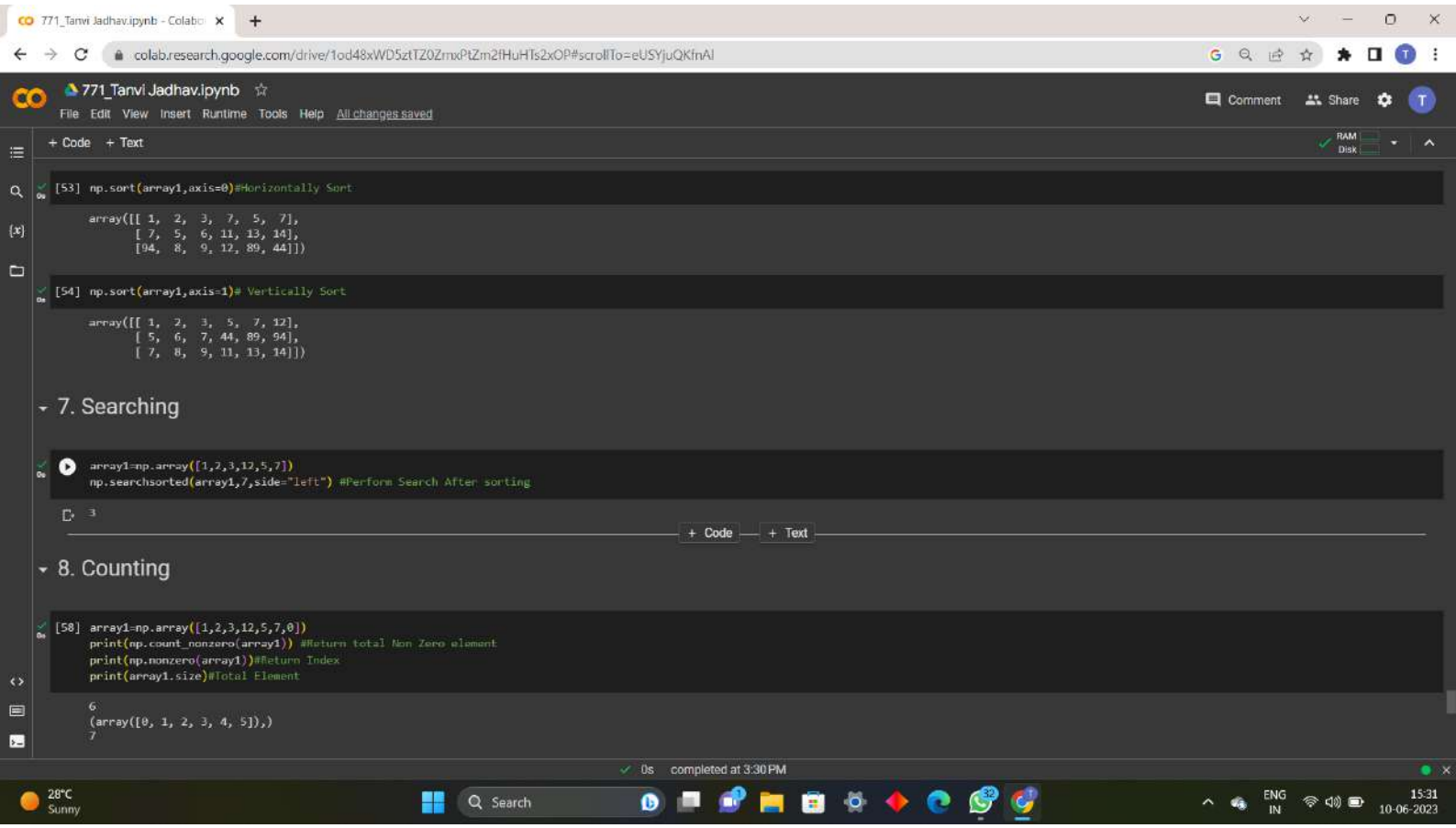
ENG IN

15:31 10-06-2023


771_Tanvi Jadhav.ipynb - Colabo
colab.research.google.com/drive/1od48xWD5ztTZ0ZmxPtZm2fHuHTs2xOP#scrollTo=eUSYjuQKfnAl
771_Tanvi Jadhav.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
4.3. Bitwise Operations
array1=np.array([1,2,3], dtype=np.uint8)
array2=np.array([4,5,6])
# AND
resultarray=np.bitwise_and (array1, array2)
print(resultarray)
# OR
resultarray=np.bitwise_or(array1,array2)
print(resultarray)
#Leftshift
resultarray=np.left_shift(array1,2)
print(resultarray)
#Rightshift
resultarray=np.right_shift (array1,2)
print(resultarray)
[0 0 2]
[5 7 7]
[5 7 7]
[5 7 7]
[48] ### You can get Binary Representation of Number #####
print(np.binary_repr(10,8))
resultarray=np.left_shift(10,2)
print (resultarray)
print(np.binary_repr(np. left_shift(10,2),8))
00001010
40
00101000
0s completed at 3:30 PM
28°C
Sunny
Search
ENG
IN
15:31
10-06-2023


771_Tanvi Jadhav.ipynb - Colabo
colab.research.google.com/drive/1od48xWD5ztTZ0ZmxPtZm2fHuHTs2xOP#scrollTo=eUSYjuQKfnAl
771_Tanvi Jadhav.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
00001010
40
00101000
5.Copying and viewing arrays
5.1 Copy
array1=np.arange(1,10)
print(array1)
newarray=array1.copy()
print (newarray)
#wmodification in original Array
array1[0]=100
print (array1)
print(newarray)
[1 2 3 4 5 6 7 8 9]
[1 2 3 4 5 6 7 8 9]
[100 2 3 4 5 6 7 8 9]
[1 2 3 4 5 6 7 8 9]
5.2 View
0s completed at 3:30 PM
28°C
Sunny
Search
ENG
IN
15:31
10-06-2023


771_Tanvi Jadhav.ipynb - Colabo
colab.research.google.com/drive/1od48xWD5ztTZ0ZmxPtZm2fHuHTs2xOP#scrollTo=eUSYjuQKfnAl
771_Tanvi Jadhav.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
5.2 View
[51] array1=np.arange(1,10)
print (array1)
newarray=array1.view()
print(newarray)
##modification in Original Array
array1[0]=100
print(array1)
print(newarray)
[1 2 3 4 5 6 7 8 9]
[1 2 3 4 5 6 7 8 9]
[100 2 3 4 5 6 7 8 9]
[1 2 3 4 5 6 7 8 9]
6. Searching
[52] array1=np.array([[1,2,3,12,5,7], [94,5,6,7,89,44],[7,8,9,11,13,14]])
print(array1)
[[1 2 3 12 5 7]
[94 5 6 7 89 44]
[7 8 9 11 13 14]]
[53] np.sort(array1,axis=0)#Horizontally Sort
array([[1, 2, 3, 7, 5, 7],
[7, 5, 6, 11, 13, 14],
[94, 8, 9, 12, 89, 44]])
0s completed at 3:30 PM
28°C
Sunny
Search
ENG
IN
15:31
10-06-2023


771_Tanvi Jadhav.ipynb - Colabo
colab.research.google.com/drive/1od48xWD5ztTZ0ZmxPtZm2fHuHTs2xOP#scrollTo=eUSYjuQKfnAl
771_Tanvi Jadhav.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
[53] np.sort(array1,axis=0)#Horizontally Sort
array([[1, 2, 3, 7, 5, 7],
[7, 5, 6, 11, 13, 14],
[94, 8, 9, 12, 89, 44]])
[54] np.sort(array1,axis=1)# Vertically Sort
array([[1, 2, 3, 5, 7, 12],
[5, 6, 7, 44, 89, 94],
[7, 8, 9, 11, 13, 14]])
7. Searching
array1=np.array([1,2,3,12,5,7])
np.searchsorted(array1,7,side="left") #Perform Search After sorting
3
+ Code + Text
8. Counting
[58] array1=np.array([1,2,3,12,5,7,0])
print(np.count_nonzero(array1)) #Return total Non Zero element
print(np.nonzero(array1))#Return Index
print(array1.size)#Total Element
6
(array([0, 1, 2, 3, 4, 5]),)
7
0s completed at 3:30 PM
28°C
Sunny
Search
ENG
IN
15:31
10-06-2023

771_Tanvi Jadhav.ipynb - Colabo

colab.research.google.com/drive/1od48xWD5ztTZ0ZmxPtZm2fHuHTs2xOP#scrollTo=eUSYjuQKfnAl

771_Tanvi Jadhav.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

8. Counting

```
[58] array1=np.array([1,2,3,12,5,7,0])
     print(np.count_nonzero(array1)) #Return total Non Zero element
     print(np.nonzero(array1))#Return Index
     print(array1.size)#Total Element
```

```
6
(array([0, 1, 2, 3, 4, 5]),)
7
```

9. Data Stacking

```
array1=np.array(np.arange(1,5).reshape(2,2))
print(array1)
array2=np.array(np.arange(11,15).reshape(2,2))
print(array2)
```

```
[[1 2]
 [3 4]]
[[11 12]
 [13 14]]
```

```
[62] newarray=np.stack([array1, array2], axis=0)
     print(newarray)
```

```
[[[ 1  2]
  [ 3  4]]

 [[11 12]
  [13 14]]]
```

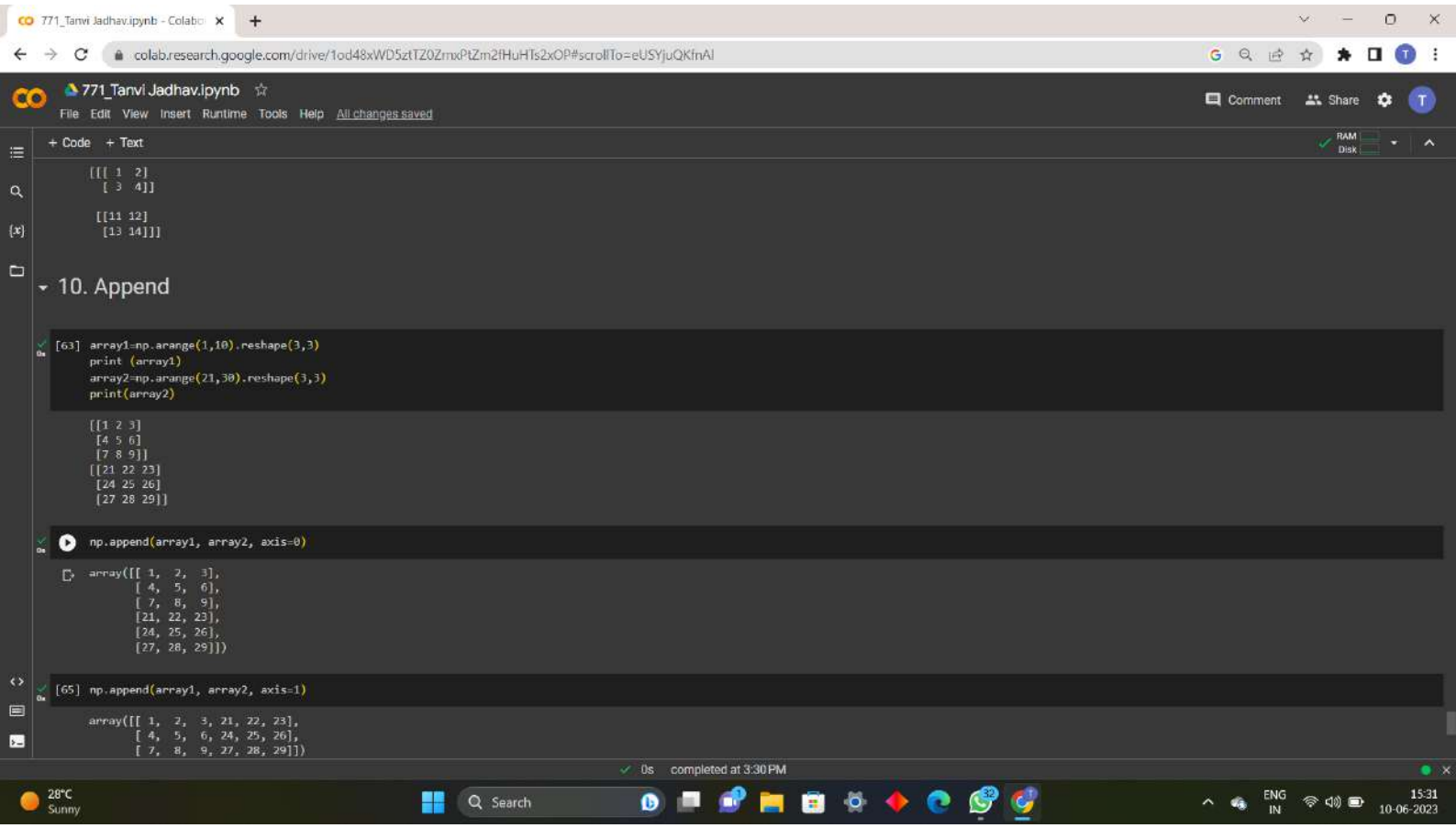
0s completed at 3:30 PM

28°C Sunny

Search

ENG IN

15:31 10-06-2023


771_Tanvi Jadhav.ipynb - Colabo
colab.research.google.com/drive/1od48xWD5ztTZ0ZmxPtZm2fHuHTs2xOP#scrollTo=eUSYjuQKfnAl
771_Tanvi Jadhav.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
[[[1 2]
[3 4]]
[[11 12]
[13 14]]]
10. Append
[63] array1=np.arange(1,10).reshape(3,3)
print (array1)
array2=np.arange(21,30).reshape(3,3)
print(array2)
[[1 2 3]
[4 5 6]
[7 8 9]]
[[21 22 23]
[24 25 26]
[27 28 29]]
np.append(array1, array2, axis=0)
array([[1, 2, 3],
[4, 5, 6],
[7, 8, 9],
[21, 22, 23],
[24, 25, 26],
[27, 28, 29]])
[65] np.append(array1, array2, axis=1)
array([[1, 2, 3, 21, 22, 23],
[4, 5, 6, 24, 25, 26],
[7, 8, 9, 27, 28, 29]])
0s completed at 3:30 PM
28°C
Sunny
Search
ENG
IN
15:31
10-06-2023

771_Tanvi Jadhav.ipynb - Colabo

colab.research.google.com/drive/1od48xWD5ztTZ0ZmxPtZm2fHuHTs2xOP#scrollTo=eUSYjuQKfnAl

771_Tanvi Jadhav.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

```
[65] np.append(array1, array2, axis=1)
```

```
array([[ 1,  2,  3, 21, 22, 23],
       [ 4,  5,  6, 24, 25, 26],
       [ 7,  8,  9, 27, 28, 29]])
```

11. Concat

```
array1=np.arange(1,10).reshape(3,3)

print(array1)

array2=np.arange(21,30).reshape(3,3)

print(array2)
```

```
[[1 2 3]
 [4 5 6]
 [7 8 9]]
[[21 22 23]
 [24 25 26]
 [27 28 29]]
```

```
[67] np.concatenate((array1, array2),axis=0)
```

```
array([[ 1,  2,  3],
       [ 4,  5,  6],
       [ 7,  8,  9],
       [21, 22, 23],
       [24, 25, 26],
       [27, 28, 29]])
```

```
[68] np.concatenate((array1, array2),axis=1)
```

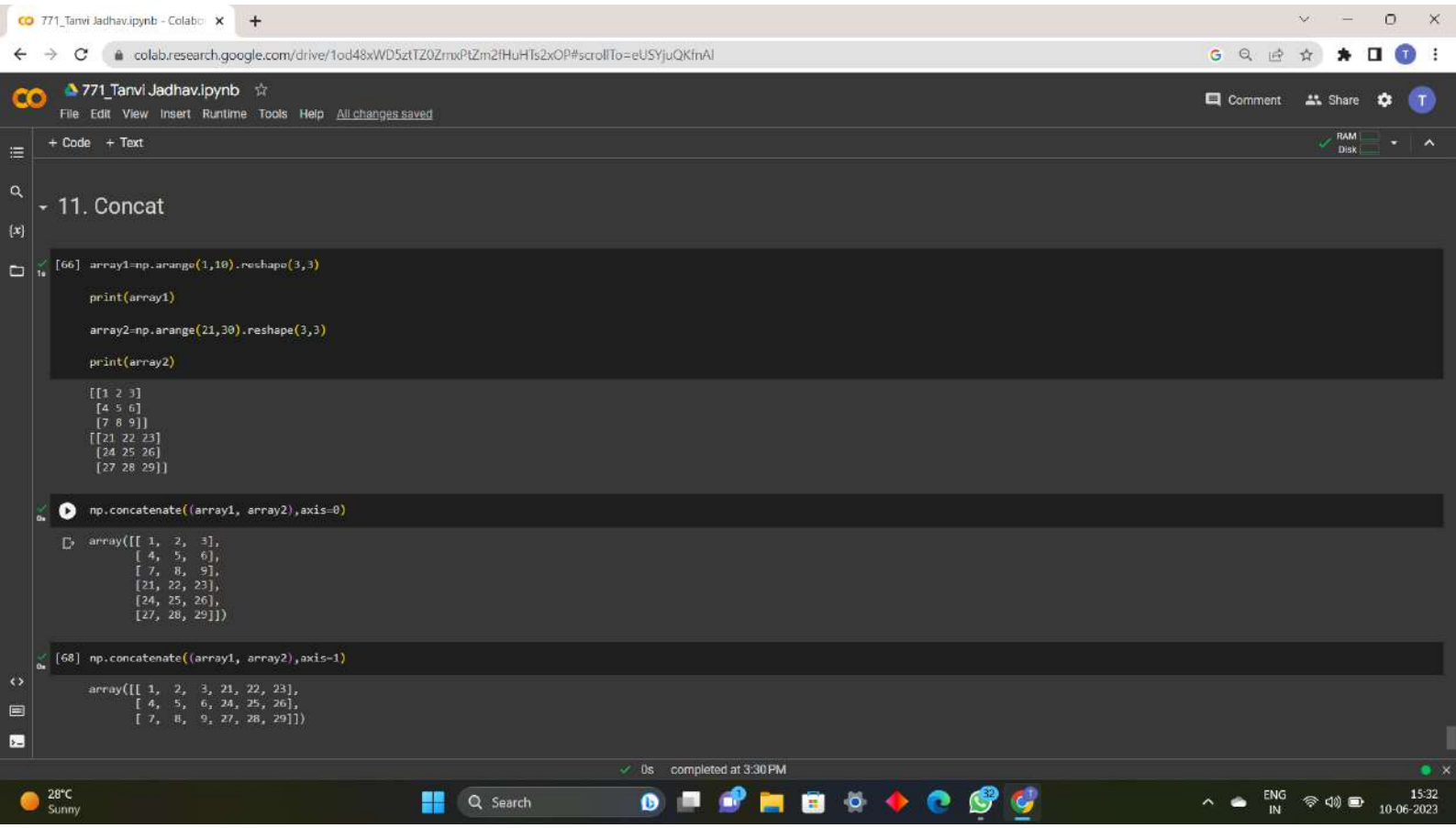
0s completed at 3:30 PM

28°C Sunny

Search

ENG IN

15:32 10-06-2023


771_Tanvi Jadhav.ipynb - Colabo
colab.research.google.com/drive/1od48xWD5ztTZ0ZmxPtZm2fHuHTs2xOP#scrollTo=eUSYjuQKfnAl
771_Tanvi Jadhav.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
11. Concat
[66] array1=np.arange(1,10).reshape(3,3)
print(array1)
array2=np.arange(21,30).reshape(3,3)
print(array2)
[[1 2 3]
[4 5 6]
[7 8 9]]
[[21 22 23]
[24 25 26]
[27 28 29]]
np.concatenate((array1, array2),axis=0)
array([[1, 2, 3],
[4, 5, 6],
[7, 8, 9],
[21, 22, 23],
[24, 25, 26],
[27, 28, 29]])
[68] np.concatenate((array1, array2),axis=1)
array([[1, 2, 3, 21, 22, 23],
[4, 5, 6, 24, 25, 26],
[7, 8, 9, 27, 28, 29]])
0s completed at 3:30 PM
28°C
Sunny
Search
ENG
IN
15:32
10-06-2023