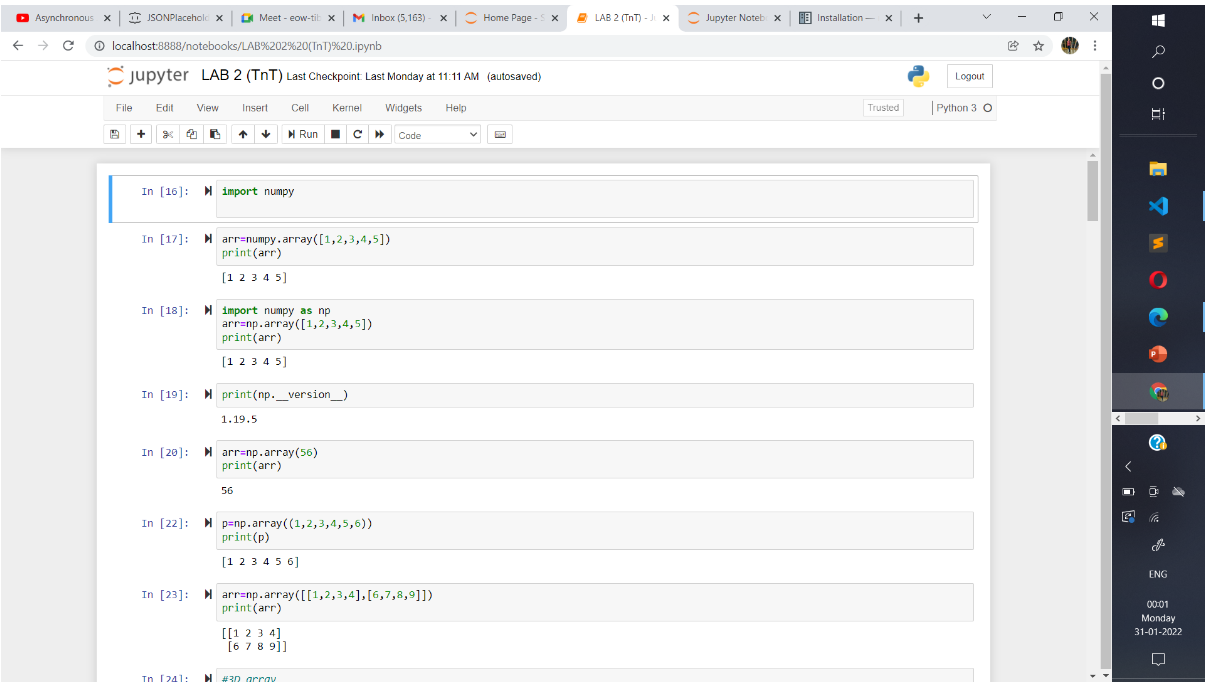
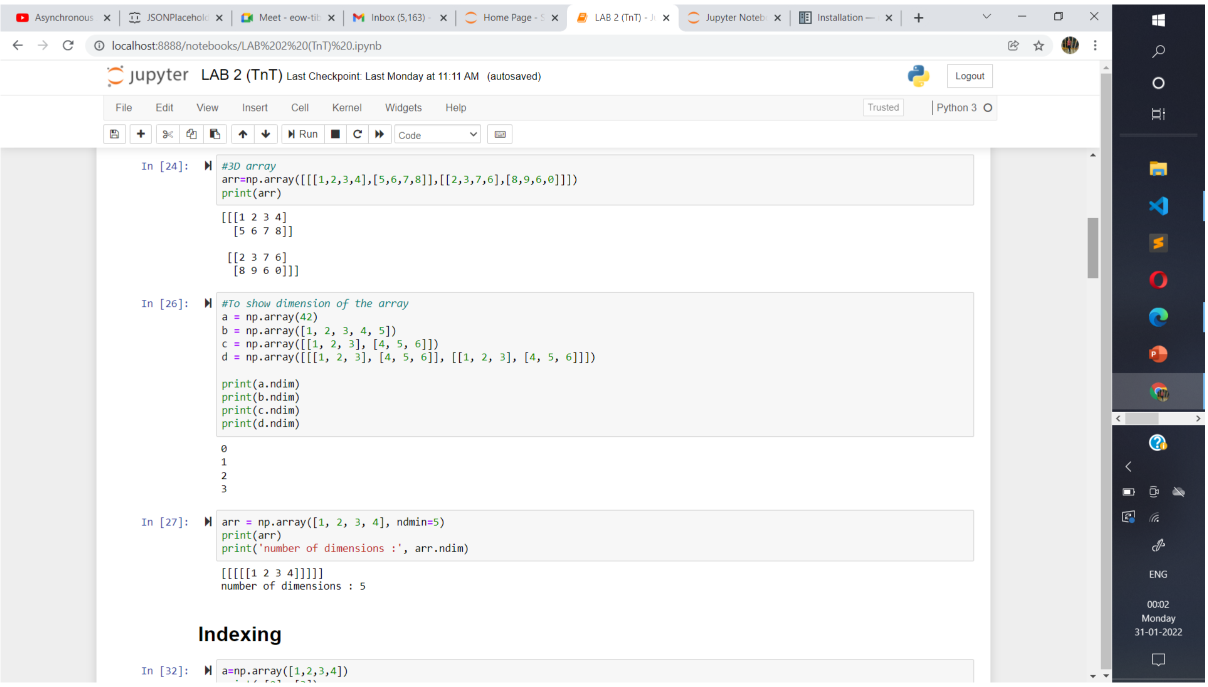
MEGHA SINGH

1928300

CSSE


Asynchronous
JSONPlacehol
Meet - eow-tib
Inbox (5,163) -
Home Page - S
LAB 2 (TnT) - J
Jupyter Noteb
Installation —
localhost:8888/notebooks/LAB%202%20(TnT)%20.ipynb
jupyter LAB 2 (TnT) Last Checkpoint: Last Monday at 11:11 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
ENG
00:01
Monday
31-01-2022

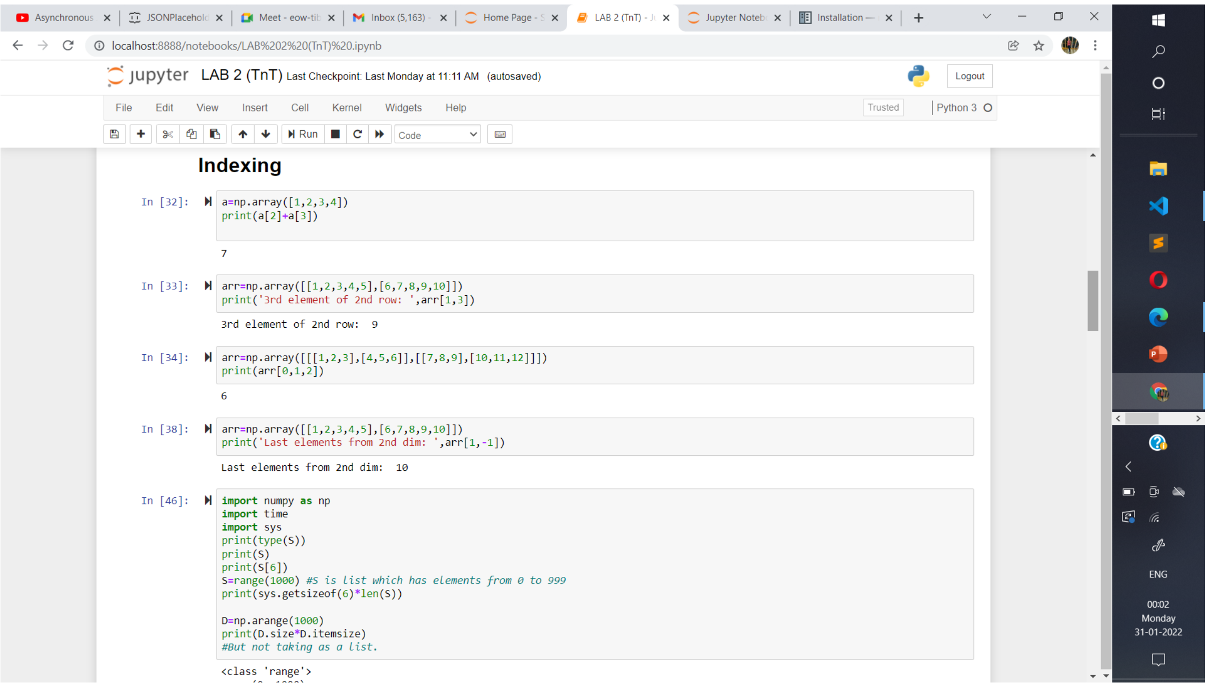
```
In [16]: import numpy

In [17]: arr=numpy.array([1,2,3,4,5])
         print(arr)

         [1 2 3 4 5]

In [18]: import numpy as np
         arr=np.array([1,2,3,4,5])
         print(arr)

         [1 2 3 4 5]

In [19]: print(np.__version__)

         1.19.5

In [20]: arr=np.array(56)
         print(arr)

         56

In [22]: p=np.array((1,2,3,4,5,6))
         print(p)

         [1 2 3 4 5 6]

In [23]: arr=np.array([[1,2,3,4],[6,7,8,9]])
         print(arr)

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

In [24]: #3D array
```


Asynchronous
JSONPlacehol
Meet - eow-tib
Inbox (5,163) -
Home Page - S
LAB 2 (TnT) - J
Jupyter Noteb
Installation —
localhost:8888/notebooks/LAB%202%20(TnT)%20.ipynb
jupyter LAB 2 (TnT) Last Checkpoint: Last Monday at 11:11 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
ENG
00:02
Monday
31-01-2022

```
In [24]: #3D array
         arr=np.array([[[1,2,3,4],[5,6,7,8]],[[2,3,7,6],[8,9,6,0]]])
         print(arr)

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

          [[2 3 7 6]
           [8 9 6 0]]]

In [26]: #To show dimension of the array
         a = np.array(42)
         b = np.array([1, 2, 3, 4, 5])
         c = np.array([[1, 2, 3], [4, 5, 6]])
         d = np.array([[[1, 2, 3], [4, 5, 6]], [[1, 2, 3], [4, 5, 6]]])

         print(a.ndim)
         print(b.ndim)
         print(c.ndim)
         print(d.ndim)

         0
         1
         2
         3

In [27]: arr = np.array([1, 2, 3, 4], ndmin=5)
         print(arr)
         print('number of dimensions :', arr.ndim)

         [[[[[1 2 3 4]]]]]
         number of dimensions : 5
```

Indexing

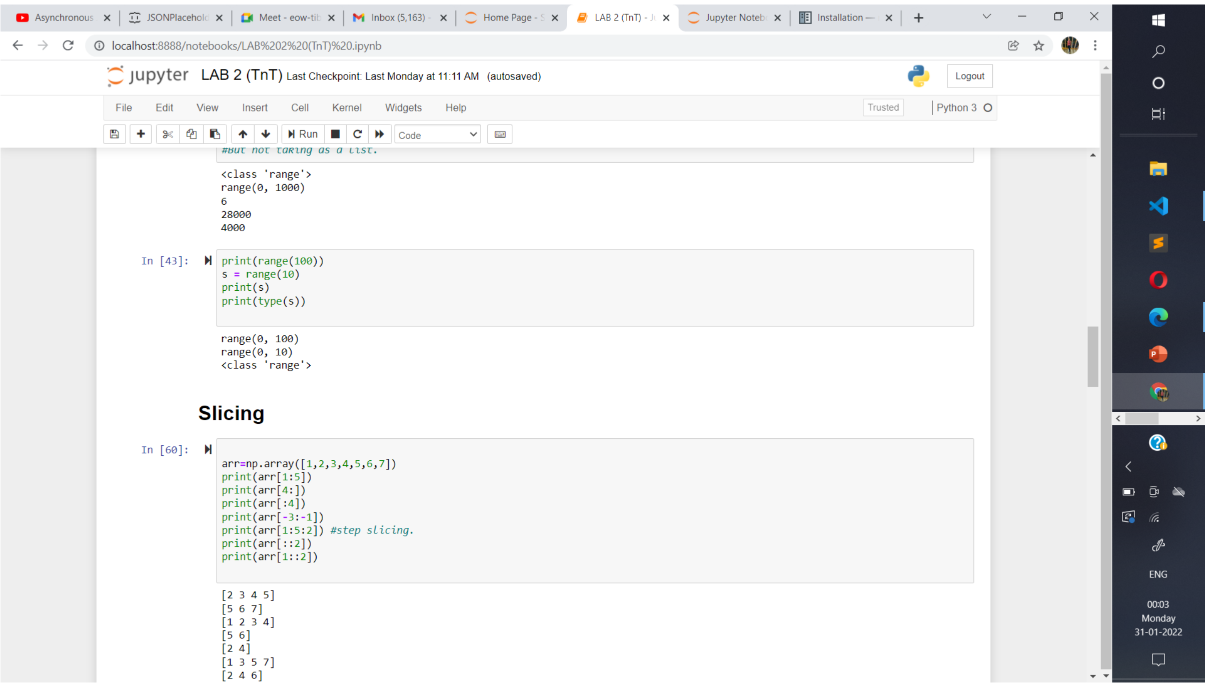
```
In [32]: a=np.array([1,2,3,4])
```



LAB 2 (TnT) Last Checkpoint: Last Monday at 11:11 AM (autosaved)

Indexing

```
In [32]:  a=np.array([1,2,3,4])
          print(a[2]+a[3])

          7

In [33]:  arr=np.array([[1,2,3,4,5],[6,7,8,9,10]])
          print('3rd element of 2nd row: ',arr[1,3])

          3rd element of 2nd row:  9

In [34]:  arr=np.array([[[1,2,3],[4,5,6]],[[7,8,9],[10,11,12]]])
          print(arr[0,1,2])

          6

In [38]:  arr=np.array([[1,2,3,4,5],[6,7,8,9,10]])
          print('Last elements from 2nd dim: ',arr[1,-1])

          Last elements from 2nd dim:  10

In [46]:  import numpy as np
          import time
          import sys
          print(type(S))
          print(S)
          print(S[6])
          S=range(1000) #S is list which has elements from 0 to 999
          print(sys.getsizeof(6)*len(S))

          D=np.arange(1000)
          print(D.size*D.itemsize)
          #But not taking as a list.

          <class 'range'>
```

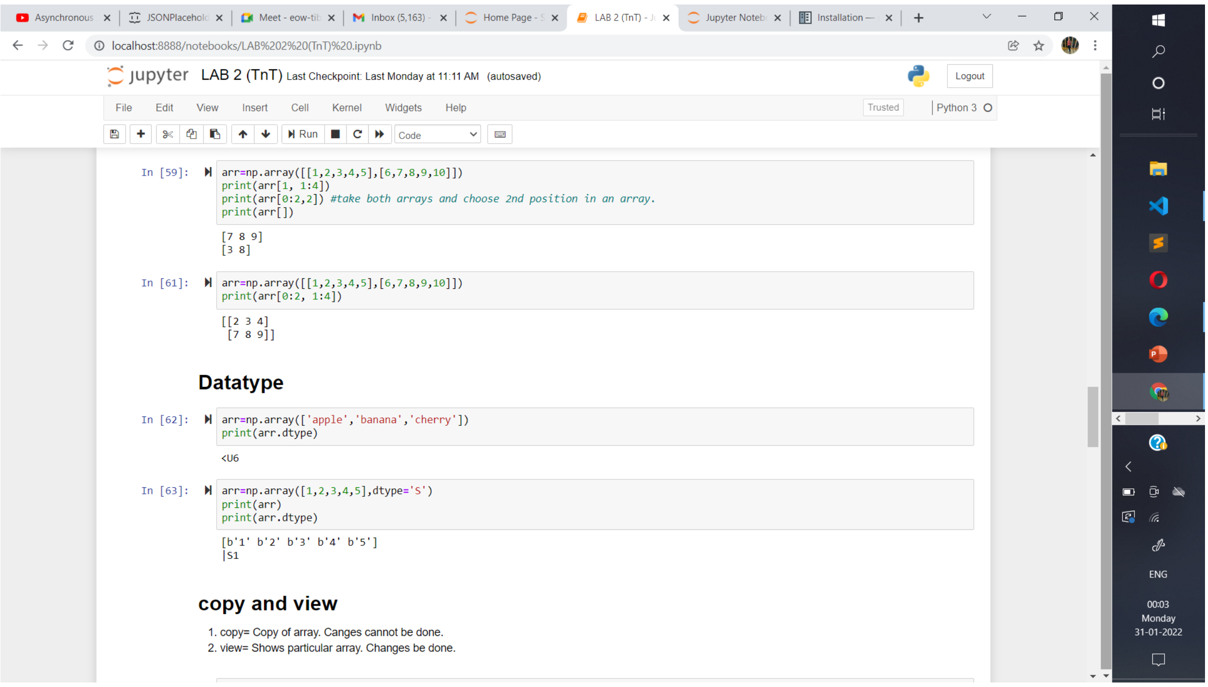


LAB 2 (TnT) Last Checkpoint: Last Monday at 11:11 AM (autosaved)

```
          #But not taking as a list.

          <class 'range'>
          range(0, 1000)
          6
          28000
          4000

In [43]:  print(range(100))
          s = range(10)
          print(s)
          print(type(s))

          range(0, 100)
          range(0, 10)
          <class 'range'>
```

Slicing

```
In [60]:  
          arr=np.array([1,2,3,4,5,6,7])
          print(arr[1:5])
          print(arr[4:])
          print(arr[:4])
          print(arr[-3:-1])
          print(arr[1:5:2]) #step slicing.
          print(arr[::2])
          print(arr[1::2])

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


localhost:8888/notebooks/LAB%202%20(TnT)%20.ipynb
jupyter LAB 2 (TnT) Last Checkpoint: Last Monday at 11:11 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
ENG
00:03
Monday
31-01-2022

```
In [59]: arr=np.array([[1,2,3,4,5],[6,7,8,9,10]])
         print(arr[1, 1:4])
         print(arr[0:2,2]) #take both arrays and choose 2nd position in an array.
         print(arr[])

         [7 8 9]
         [3 8]
```

```
In [61]: arr=np.array([[1,2,3,4,5],[6,7,8,9,10]])
         print(arr[0:2, 1:4])

         [[2 3 4]
          [7 8 9]]
```

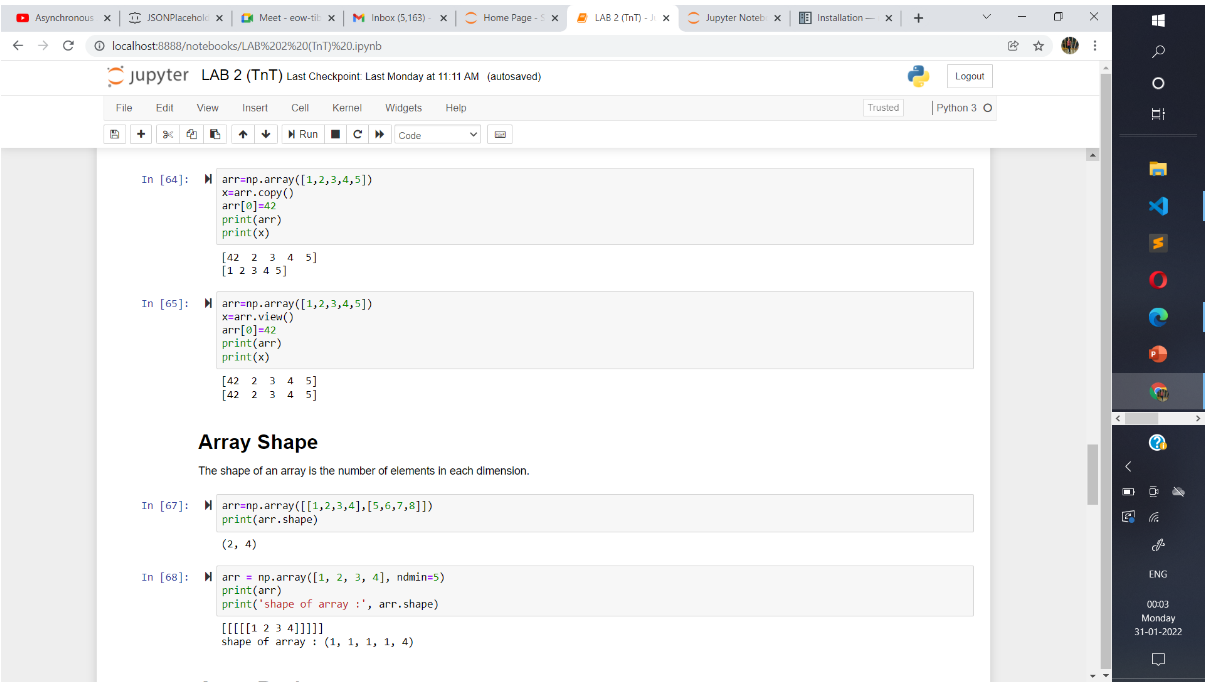
Datatype

```
In [62]: arr=np.array(['apple','banana','cherry'])
         print(arr.dtype)

         <U6
```

```
In [63]: arr=np.array([1,2,3,4,5],dtype='S')
         print(arr)
         print(arr.dtype)

         [b'1' b'2' b'3' b'4' b'5']
         |S1
```

copy and view

1. copy= Copy of array. Canges cannot be done.
2. view= Shows particular array. Changes be done.


localhost:8888/notebooks/LAB%202%20(TnT)%20.ipynb
jupyter LAB 2 (TnT) Last Checkpoint: Last Monday at 11:11 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
ENG
00:03
Monday
31-01-2022

```
In [64]: arr=np.array([1,2,3,4,5])
         x=arr.copy()
         arr[0]=42
         print(arr)
         print(x)

         [42  2  3  4  5]
         [1 2 3 4 5]
```

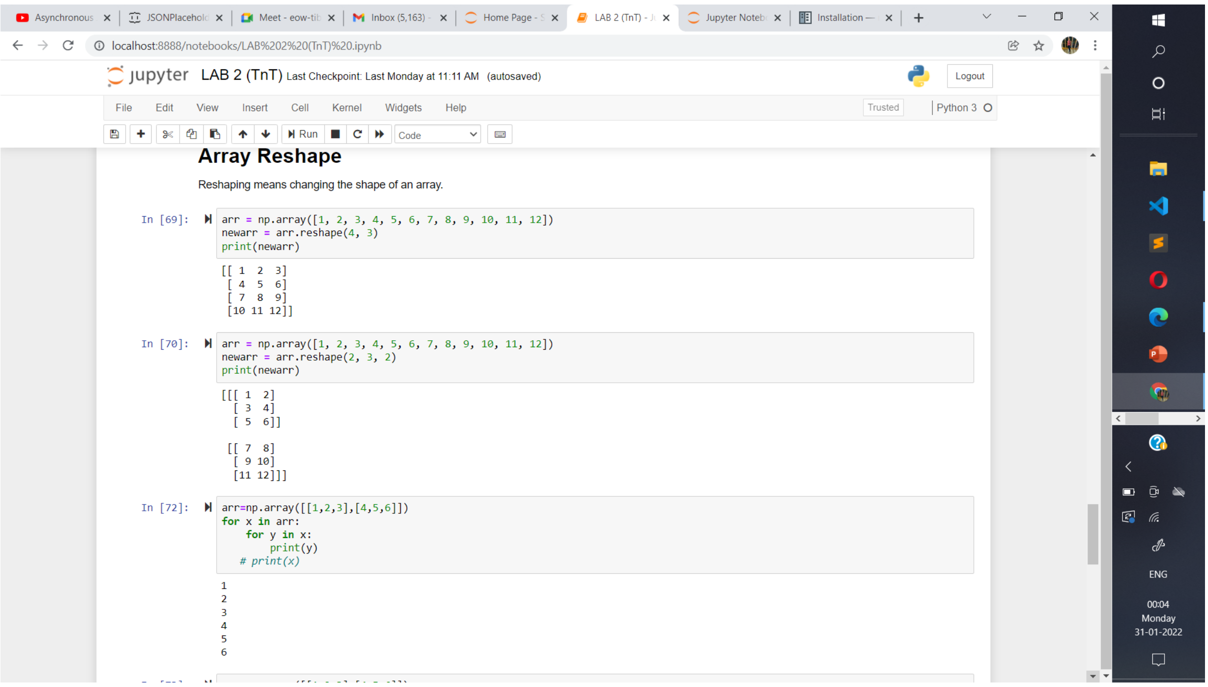
```
In [65]: arr=np.array([1,2,3,4,5])
         x=arr.view()
         arr[0]=42
         print(arr)
         print(x)

         [42  2  3  4  5]
         [42  2  3  4  5]
```

Array Shape

The shape of an array is the number of elements in each dimension.

```
In [67]: arr=np.array([[1,2,3,4],[5,6,7,8]])
         print(arr.shape)

         (2, 4)
```

```
In [68]: arr = np.array([1, 2, 3, 4], ndmin=5)
         print(arr)
         print('shape of array :', arr.shape)

         [[[[[1 2 3 4]]]]]
         shape of array : (1, 1, 1, 1, 4)
```


Asynchronous
JSONPlaceholc
Meet - eow-tib
Inbox (5,163)
Home Page
LAB 2 (TnT)
Jupyter Noteb
Installation —
localhost:8888/notebooks/LAB%202%20(TnT)%20.ipynb
jupyter LAB 2 (TnT) Last Checkpoint: Last Monday at 11:11 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
ENG
00:04
Monday
31-01-2022

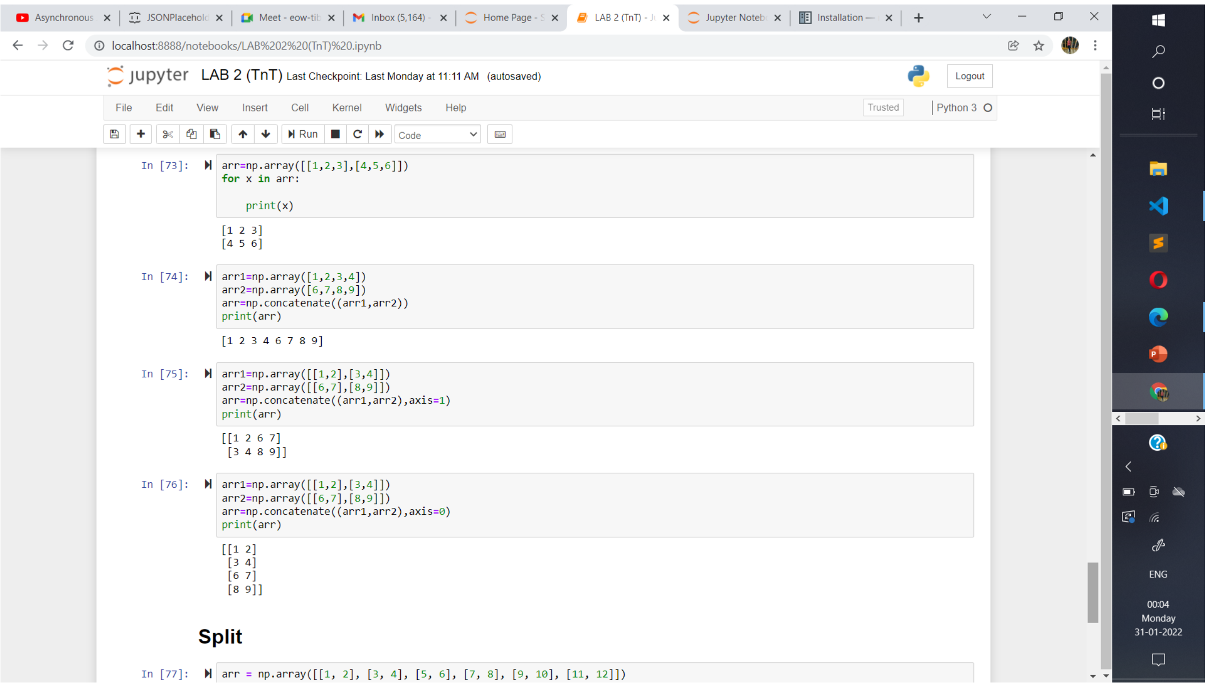
Array Reshape

Reshaping means changing the shape of an array.

```
In [69]:  arr = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])
          newarr = arr.reshape(4, 3)
          print(newarr)

          [[ 1  2  3]
           [ 4  5  6]
           [ 7  8  9]
           [10 11 12]]
```

```
In [70]:  arr = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])
          newarr = arr.reshape(2, 3, 2)
          print(newarr)

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

           [[ 7  8]
            [ 9 10]
            [11 12]]]
```

```
In [72]:  arr=np.array([[1,2,3],[4,5,6]])
          for x in arr:
              for y in x:
                  print(y)
             # print(x)

          1
          2
          3
          4
          5
          6
```


Asynchronous
JSONPlaceholc
Meet - eow-tib
Inbox (5,164)
Home Page
LAB 2 (TnT)
Jupyter Noteb
Installation —
localhost:8888/notebooks/LAB%202%20(TnT)%20.ipynb
jupyter LAB 2 (TnT) Last Checkpoint: Last Monday at 11:11 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
ENG
00:04
Monday
31-01-2022

```
In [73]:  arr=np.array([[1,2,3],[4,5,6]])
          for x in arr:

              print(x)

          [1 2 3]
          [4 5 6]
```

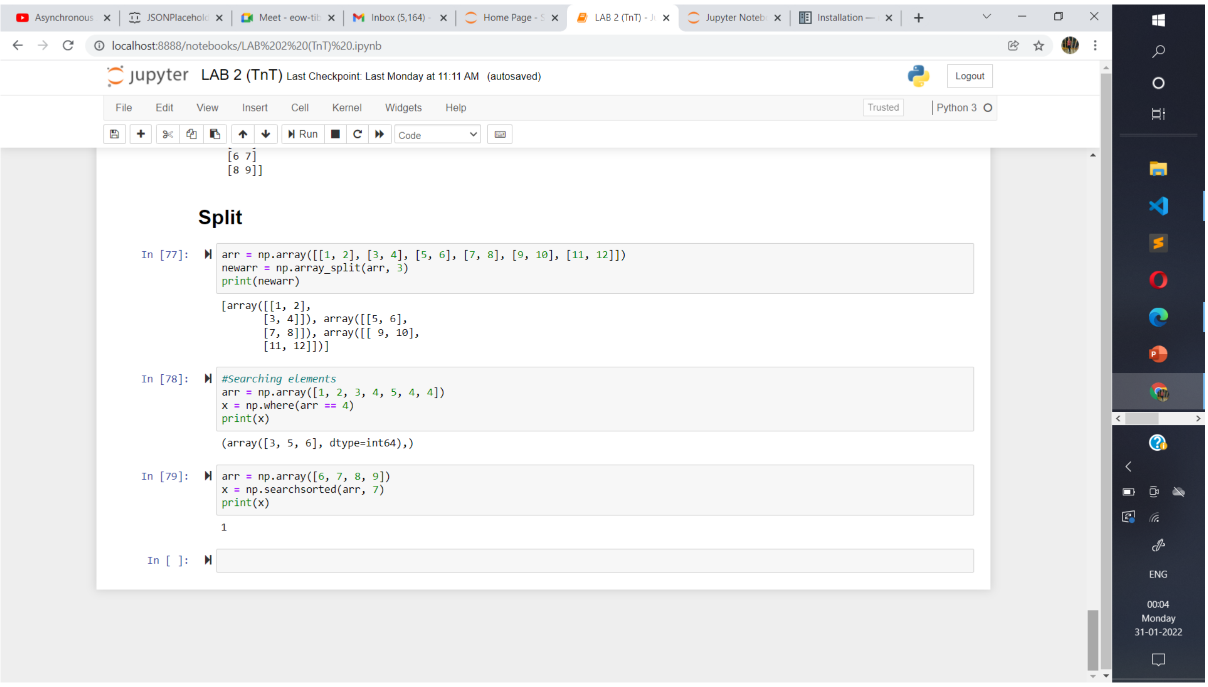
```
In [74]:  arr1=np.array([1,2,3,4])
          arr2=np.array([6,7,8,9])
          arr=np.concatenate((arr1,arr2))
          print(arr)

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

```
In [75]:  arr1=np.array([[1,2],[3,4]])
          arr2=np.array([[6,7],[8,9]])
          arr=np.concatenate((arr1,arr2),axis=1)
          print(arr)

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

```
In [76]:  arr1=np.array([[1,2],[3,4]])
          arr2=np.array([[6,7],[8,9]])
          arr=np.concatenate((arr1,arr2),axis=0)
          print(arr)

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

Split

```
In [77]:  arr = np.array([[1, 2], [3, 4], [5, 6], [7, 8], [9, 10], [11, 12]])
```



LAB 2 (TnT) Last Checkpoint: Last Monday at 11:11 AM (autosaved)

```
 [6 7]
 [8 9]]
```

Split

In [77]:

```
arr = np.array([[1, 2], [3, 4], [5, 6], [7, 8], [9, 10], [11, 12]])
newarr = np.array_split(arr, 3)
print(newarr)
```

```
[array([[1, 2],
       [3, 4]]), array([[5, 6],
       [7, 8]]), array([[ 9, 10],
       [11, 12]])]
```

In [78]:

```
#Searching elements
arr = np.array([1, 2, 3, 4, 5, 4, 4])
x = np.where(arr == 4)
print(x)
```

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

In [79]:

```
arr = np.array([6, 7, 8, 9])
x = np.searchsorted(arr, 7)
print(x)
```

```
1
```

In [ ]: