Torrodjae Somerville
CTEC 298-101
Dr.Bemley
11/12/2025

jupyter Numpy Tutorial Completion Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python [conda env:base] * Anaconda Toolbox

[1]:
```
import numpy as np

a = np.zeros(3)
a
```

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

[2]:
```
z = np.zeros(10)
z
```

[2]:
```
array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

[3]:
```
z.shape
```

[3]:
```
(10,)
```

[4]:
```
z.shape = (10,1)
z
```

[4]:
```
array([[0.],
       [0.],
       [0.],
       [0.],
       [0.],
       [0.],
       [0.],
       [0.],
       [0.],
       [0.]])
```

[5]:
```
z = np.ones(10)
z
```

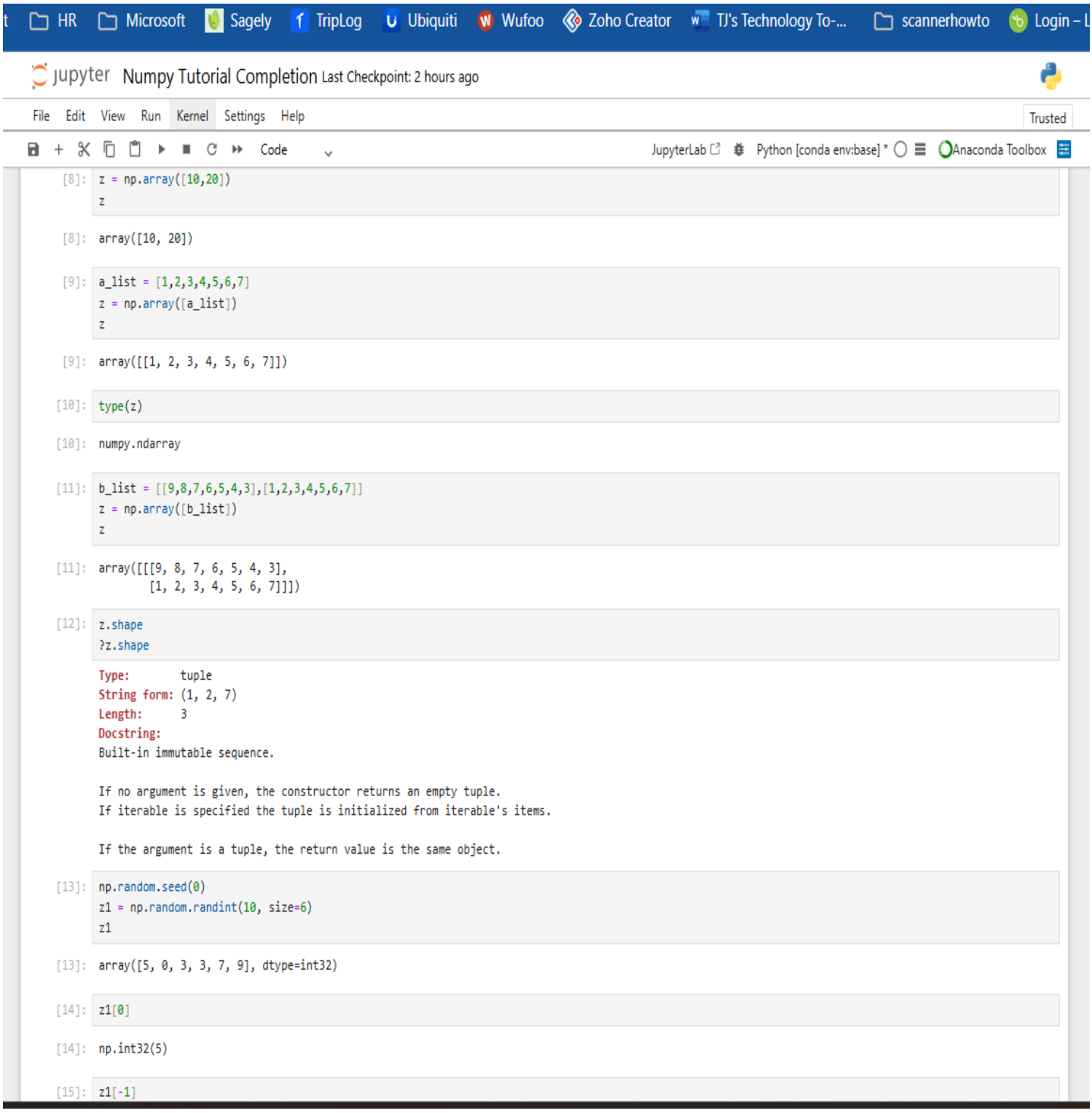
[5]:
```
array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1.])
```

[6]:
```
z = np.empty(3)
z
```

[6]:
```
array([0., 0., 0.])
```

[7]:
```
z = np.linspace(2, 10, 5)
z
```

[7]:
```
array([ 2.,  4.,  6.,  8., 10.])
```


HR
Microsoft
Sagely
TripLog
Ubiquiti
Wufoo
Zoho Creator
TJ's Technology To-...
scannerhowto
Login – L
jupyter
Numpy Tutorial Completion
Last Checkpoint: 2 hours ago
File
Edit
View
Run
Kernel
Settings
Help
Trusted
Code
JupyterLab
Python [conda env:base] *
Anaconda Toolbox
[8]: z = np.array([10,20])
z
[8]: array([10, 20])
[9]: a_list = [1,2,3,4,5,6,7]
z = np.array([a_list])
z
[9]: array([[1, 2, 3, 4, 5, 6, 7]])
[10]: type(z)
[10]: numpy.ndarray
[11]: b_list = [[9,8,7,6,5,4,3],[1,2,3,4,5,6,7]]
z = np.array([b_list])
z
[11]: array([[[9, 8, 7, 6, 5, 4, 3],
[1, 2, 3, 4, 5, 6, 7]]])
[12]: z.shape
?z.shape
Type: tuple
String form: (1, 2, 7)
Length: 3
Docstring:
Built-in immutable sequence.
If no argument is given, the constructor returns an empty tuple.
If iterable is specified the tuple is initialized from iterable's items.
If the argument is a tuple, the return value is the same object.
[13]: np.random.seed(0)
z1 = np.random.randint(10, size=6)
z1
[13]: array([5, 0, 3, 3, 7, 9], dtype=int32)
[14]: z1[0]
[14]: np.int32(5)
[15]: z1[-1]

jupyter Numpy Tutorial Completion Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python [conda env:base] * Anaconda Toolbox

[17]:
```
from skimage import io
photo = io.imread('2-2.jpg')
type(photo)
```

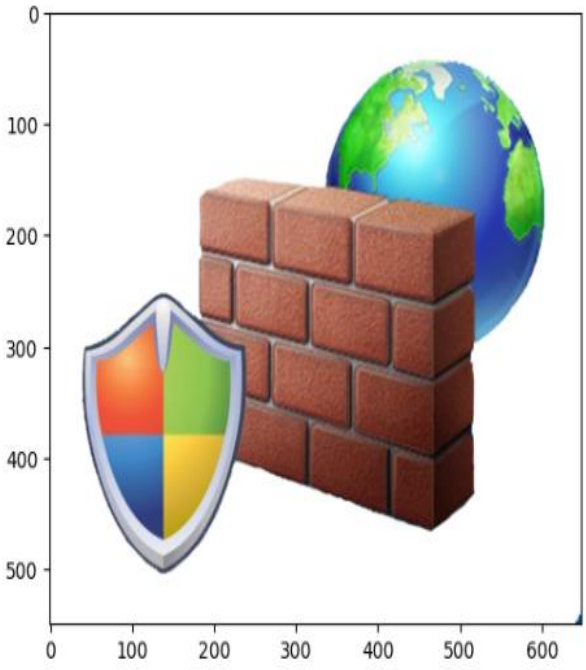
[17]:
```
numpy.ndarray
```

[18]:
```
photo.shape
```

[18]:
```
(550, 648, 3)
```

[19]:
```
import matplotlib.pyplot as plt
plt.imshow(photo)
```

[19]:
```
<matplotlib.image.AxesImage at 0x2de035ef380>
```



[20]:
```
plt.imshow(photo[::-1])
```

[20]:
```
<matplotlib.image.AxesImage at 0x2de048d1bd0>
```

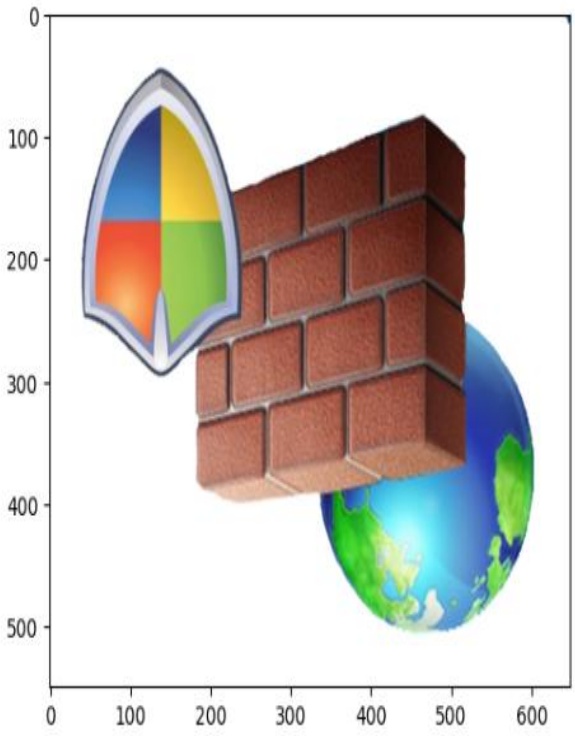
jupyter Numpy Tutorial Completion Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python [conda env:base] * Anaconda Toolbox

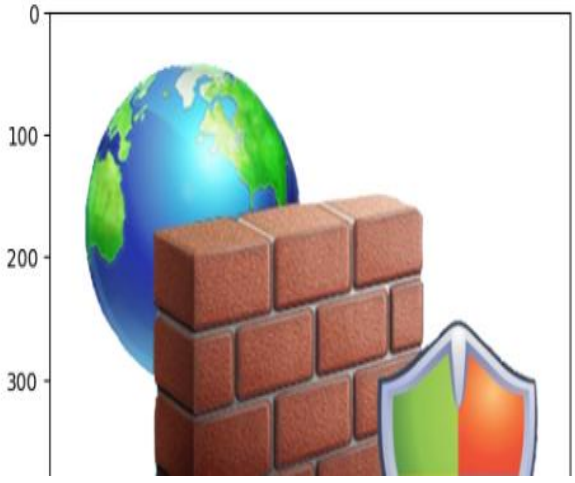
[20]:
```
plt.imshow(photo[::-1])
```

[20]:
```
<matplotlib.image.AxesImage at 0x2de048d1bd0>
```



[21]:
```
plt.imshow(photo[:,::-1])
```

[21]:
```
<matplotlib.image.AxesImage at 0x2de04964e10>
```



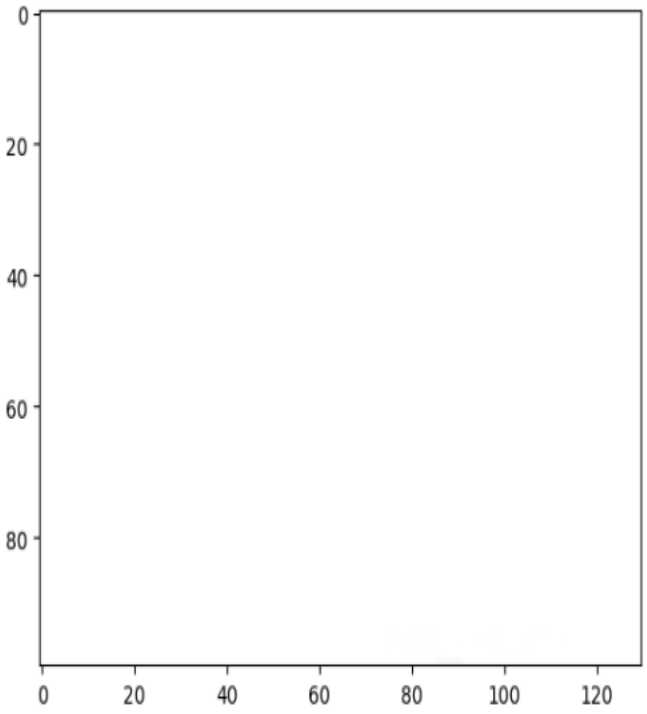
jupyter Numpy Tutorial Completion Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python [conda env:base] * Anaconda Toolbox

[22]:
```
plt.imshow(photo[50:150, 150:280])
```

[22]:
```
<matplotlib.image.AxesImage at 0x2de04d2b610>
```



[23]:
```
plt.imshow(photo[::2, ::2])
```

[23]:
```
<matplotlib.image.AxesImage at 0x2de04dc9450>
```

jupyter Numpy Tutorial Completion Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python [conda env:base] * Anaconda Toolbox

[24]:
```
photo
photo_sin = np.sin(photo)
photo_sin
```

[24]:
```
array([[[-0.5063, -0.5063, -0.5063],
        [-0.5063, -0.5063, -0.5063],
        [-0.5063, -0.5063, -0.5063],
        ...,
        [-0.5063, -0.5063, -0.5063],
        [-0.5063, -0.5063, -0.5063],
        [-0.5063, -0.5063, -0.5063]],

       [[-0.5063, -0.5063, -0.5063],
        [-0.5063, -0.5063, -0.5063],
        [-0.5063, -0.5063, -0.5063],
        ...,
        [-0.5063, -0.5063, -0.5063],
        [-0.5063, -0.5063, -0.5063],
        [-0.5063, -0.5063, -0.5063]],

       [[-0.5063, -0.5063, -0.5063],
        [-0.5063, -0.5063, -0.5063],
        [-0.5063, -0.5063, -0.5063],
        ...,
        [-0.5063, -0.5063, -0.5063],
        [-0.5063, -0.5063, -0.5063],
        [-0.5063, -0.5063, -0.5063]],

       ...,

       [[-0.5063, -0.5063, -0.5063],
        [-0.5063, -0.5063, -0.5063],
        [-0.5063, -0.5063, -0.5063],
        ...,
        [ 0.8413, -0.9937, -0.9746],
        [ 0.8413, -0.9937,  0.609 ],
        [ 0.    , -0.63  ,  0.609 ]],

       [[-0.5063, -0.5063, -0.5063],
        [-0.5063, -0.5063, -0.5063],
        [-0.5063, -0.5063, -0.5063],
        ...,
        [ 0.8413, -0.9937,  0.609 ],
        [ 0.909 , -0.9937,  0.9966],
        [ 0.8413,  0.514 , -0.3384]],

       [[-0.5063, -0.5063, -0.5063],
        [-0.5063, -0.5063, -0.5063],
```

jupyter Numpy Tutorial Completion Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python [conda env:base] * Anaconda Toolbox

[25]:
```
print(np.sum(photo))
print(np.prod(photo))
print(np.mean(photo))
print(np.std(photo))
print(np.var(photo))
print(np.min(photo))
print(np.max(photo))
print(np.argmin(photo))
print(np.argmax(photo))
```

```
209187432
0
195.6485521885522
79.41634712550251
6306.956190758311
0
255
277145
0
```

[26]:
```
z = np.array([1,2,3,4,5])
```

[27]:
```
z<3
```

[27]:
```
array([ True,  True, False, False, False])
```

[28]:
```
z>3
```

[28]:
```
array([False, False, False,  True,  True])
```

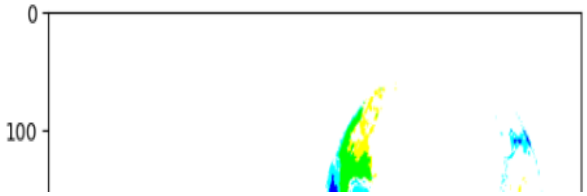
[29]:
```
z[z>3]
```

[29]:
```
array([4, 5])
```

[30]:
```
photo_masked = np.where(photo > 50, 255, 0)
```

[31]:
```
plt.imshow(photo_masked)
```

[31]:
```
<matplotlib.image.AxesImage at 0x2de04e179d0>
```



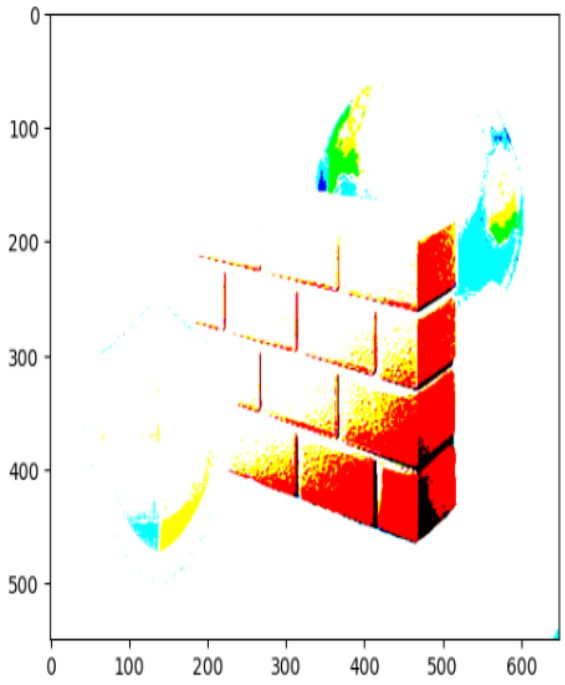
ToAssist | HR | Microsoft | Sagely | TripLog | Ubiquiti | Wufoo | Zoho Creator | TJ's Technology To-... | scannerhowto | Login – La

jupyter Numpy Tutorial Completion Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python [conda env:base] * Anaconda Toolbox

[31]:
```
<matplotlib.image.AxesImage at 0x2de04e179d0>
```



[32]:
```
a_array = np.array([1,2,3,4,5])
b_array = np.array([6,7,8,9,10])
```

[33]:
```
a_array + b_array
```

[33]:
```
array([ 7,  9, 11, 13, 15])
```

[34]:
```
a_array + 30
```

[34]:
```
array([31, 32, 33, 34, 35])
```

[35]:
```
a_array * b_array
```

[35]:
```
array([ 6, 14, 24, 36, 50])
```

[36]:
```
a_array * 10
```

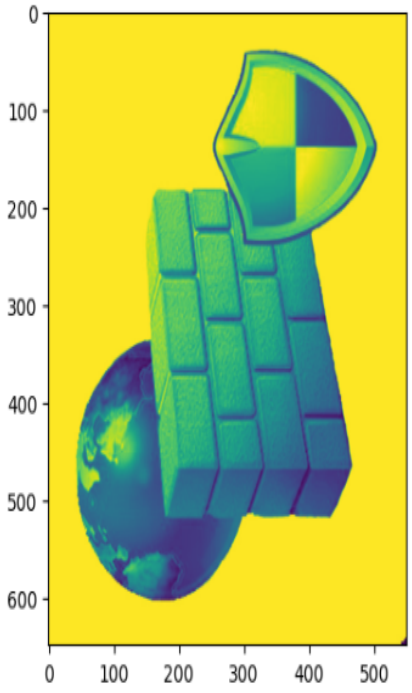
[36]:
```
array([10, 20, 30, 40, 50])
```

[37]:
```
a_array @ b_array
```

[37]:
```
np.int64(130)
```

[38]:
```
plt.imshow(photo[:,:,0].T)
```

[38]:
```
<matplotlib.image.AxesImage at 0x2de04ebe350>
```



[39]:
```
x = np.array([2, 1, 4, 3, 5])
np.sort(x)
```

[39]:
```
array([1, 2, 3, 4, 5])
```

[41]:
```
A = np.array([[1,2,3],[4,5,6],[7,8,9]])
```

[42]:
```
A[0]
```

[42]:
```
array([1, 2, 3])
```

[43]:
```
A[1]
```

[43]:
```
array([4, 5, 6])
```

[44]:
```
A[:,0]
```

jupyter Numpy Tutorial Completion Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python [conda env:base] * Anaconda Toolbox

[44]:
```
array([1, 4, 7])
```

[45]:
```
A[:,1]
```

[45]:
```
array([2, 5, 8])
```

[46]:
```
A[:,2]
```

[46]:
```
array([3, 6, 9])
```

[47]:
```
A[0,0]
```

[47]:
```
np.int64(1)
```

[48]:
```
A[2,1]
```

[48]:
```
np.int64(8)
```

[49]:
```
A[0,1]
```

[49]:
```
np.int64(2)
```

[50]:
```
A.shape
```

[50]:
```
(3, 3)
```

[51]:
```
A[1:2,0:2]
```

[51]:
```
array([[4, 5]])
```

[52]:
```
A[1:3,1:3]
```

[52]:
```
array([[5, 6],
       [8, 9]])
```

[53]:
```
A[1:,1:]
```

[53]:
```
array([[5, 6],
       [8, 9]])
```

[54]:
```
A>5
```

[54]:
```
array([[False, False, False],
       [False, False,  True],
       [ True,  True,  True]])
```

jupyter Numpy Tutorial Completion Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python [conda env:base] * Anaconda Toolbox

[55]:
```
A[A>5]
```

[55]:
```
array([6, 7, 8, 9])
```

[56]:
```
np.zeros((4,5))
```

[56]:
```
array([[0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.]])
```

[57]:
```
np.ones((3,2))
```

[57]:
```
array([[1., 1.],
       [1., 1.],
       [1., 1.]])
```

[58]:
```
np.diag([1,2,3])
```

[58]:
```
array([[1, 0, 0],
       [0, 2, 0],
       [0, 0, 3]])
```

[59]:
```
np.empty((3,3))
```

[59]:
```
array([[0.00648, 0.00648, 0.00648],
       [0.00648, 0.00648, 0.00648],
       [0.00648, 0.00648, 0.00648]])
```

[60]:
```
np.random.random((4,2))
```

[60]:
```
array([[0.6235637 , 0.38438171],
       [0.29753461, 0.05671298],
       [0.27265629, 0.47766512],
       [0.81216873, 0.47997717]])
```

[63]:
```
B = np.zeros((3,3))
```

[64]:
```
np.vstack((A,B))
```

[64]:
```
array([[1., 2., 3.],
       [4., 5., 6.],
       [7., 8., 9.],
       [0., 0., 0.],
       [0., 0., 0.],
       [0., 0., 0.]])
```

[65]:
```
np.hstack((A,B))
```

jupyter Numpy Tutorial Completion Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python [conda env:base] * Anaconda Toolbox

[67]:
```
np.append(A,B)
```

[67]:
```
array([1., 2., 3., 4., 5., 6., 7., 8., 9., 0., 0., 0., 0., 0., 0., 0.,
       0.])
```

[68]:
```
np.hsplit(A,3)
```

[68]:
```
[array([[1],
        [4],
        [7]]),
 array([[2],
        [5],
        [8]]),
 array([[3],
        [6],
        [9]])]
```

[69]:
```
np.vsplit(A,3)
```

[69]:
```
[array([[1, 2, 3]]), array([[4, 5, 6]]), array([[7, 8, 9]])]
```

[70]:
```
np.transpose(A)
```

[70]:
```
array([[1, 4, 7],
       [2, 5, 8],
       [3, 6, 9]])
```

[71]:
```
A.T
```

[71]:
```
array([[1, 4, 7],
       [2, 5, 8],
       [3, 6, 9]])
```

[72]:
```
np.dot(A,A)
```

[72]:
```
array([[ 30,  36,  42],
       [ 66,  81,  96],
       [102, 126, 150]])
```

[73]:
```
A@A
```

[73]:
```
array([[ 30,  36,  42],
       [ 66,  81,  96],
       [102, 126, 150]])
```

[74]:
```
5*A
```

[74]:
```
array([[ 5, 10, 15],
```

jupyter Numpy Tutorial Completion Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python [conda env:base] * Anaconda Toolbox

[75]:
```
A/2
```

[75]:
```
array([[0.5, 1. , 1.5],
       [2. , 2.5, 3. ],
       [3.5, 4. , 4.5]])
```

[76]:
```
A+3
```

[76]:
```
array([[ 4,  5,  6],
       [ 7,  8,  9],
       [10, 11, 12]])
```

[77]:
```
np.mean(A)
```

[77]:
```
np.float64(5.0)
```

[78]:
```
np.mean(A,axis=0)
```

[78]:
```
array([4., 5., 6.])
```

[79]:
```
np.std(A)
```

[79]:
```
np.float64(2.581988897471611)
```

[80]:
```
np.linalg.eig(A)
```

[80]:
```
EigResult(eigenvalues=array([ 1.61168440e+01, -1.11684397e+00, -3.38433605e-16]), eigenvectors=array([[-0.23197069, -0.78583024,  0.40824829],
       [-0.52532209, -0.08675134, -0.81649658],
       [-0.8186735 ,  0.61232756,  0.40824829]]))
```

[81]:
```
np.linalg.qr(A)
```

[81]:
```
QRResult(Q=array([[-0.12309149,  0.90453403,  0.40824829],
       [-0.49236596,  0.30151134, -0.81649658],
       [-0.86164044, -0.30151134,  0.40824829]]), R=array([[-8.12403840e+00, -9.60113630e+00, -1.10782342e+01],
       [ 0.00000000e+00,  9.04534034e-01,  1.80906807e+00],
       [ 0.00000000e+00,  0.00000000e+00, -1.11164740e-15]]))
```

[82]:
```
np.linalg.svd(A)
```

[82]:
```
SVDResult(U=array([[-0.21483724,  0.88723069,  0.40824829],
       [-0.52058739,  0.24964395, -0.81649658],
       [-0.82633754, -0.38794278,  0.40824829]]), S=array([1.68481034e+01, 1.06836951e+00, 3.33475287e-16]), Vh=array([[-0.47967118, -0.57236779, -0.665
06441],
       [-0.77669099, -0.07568647,  0.62531805],
       [-0.40824829,  0.81649658, -0.40824829]]))
```

[ ]: