FUNDAMENTALS OF MACHINE LEARNING IN DATA SCIENCE

CSIS 3290

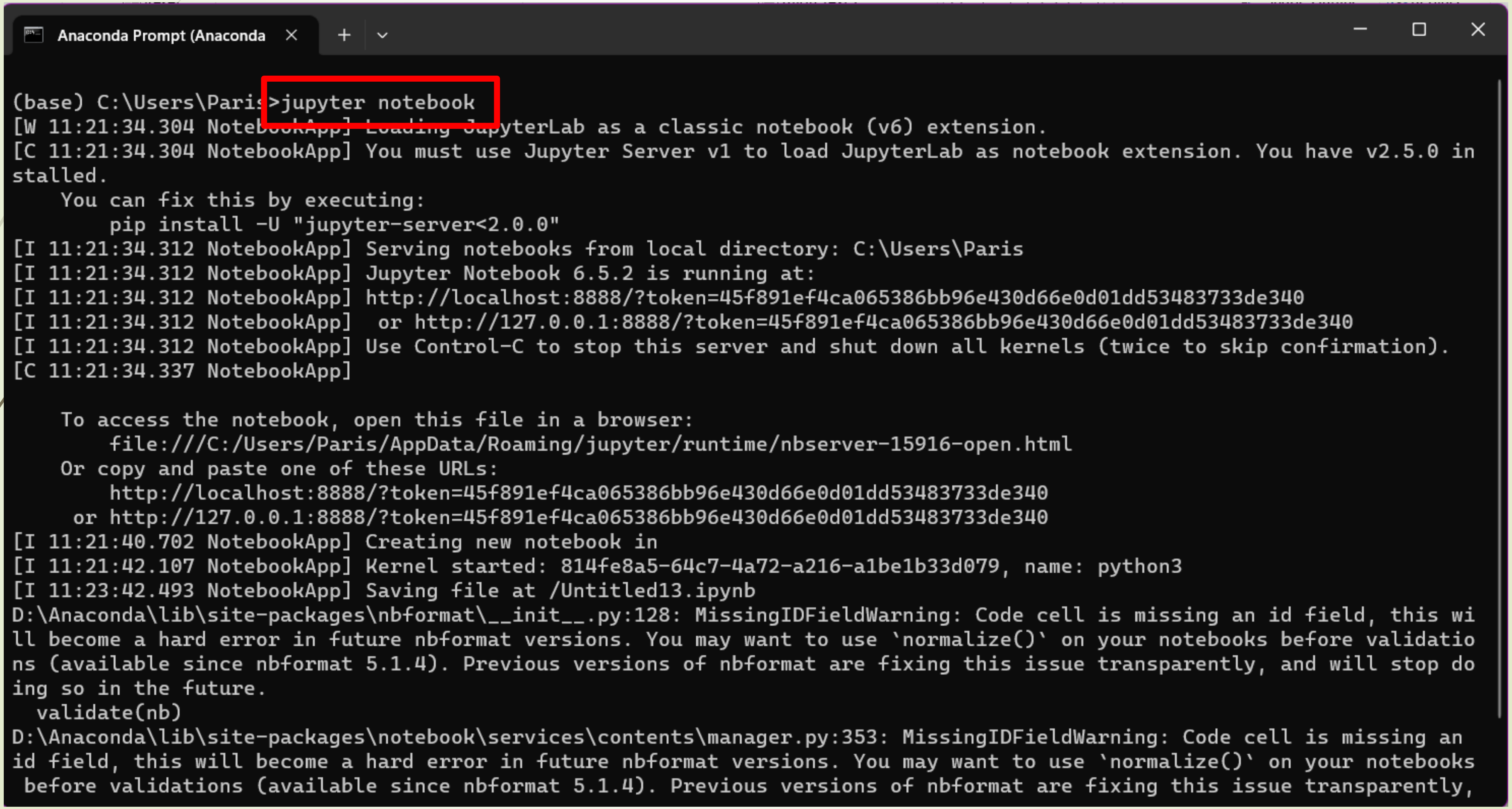
NUMPY LIBRARY

FATEMEH AHMADI (PH.D.)

Libraries in Python

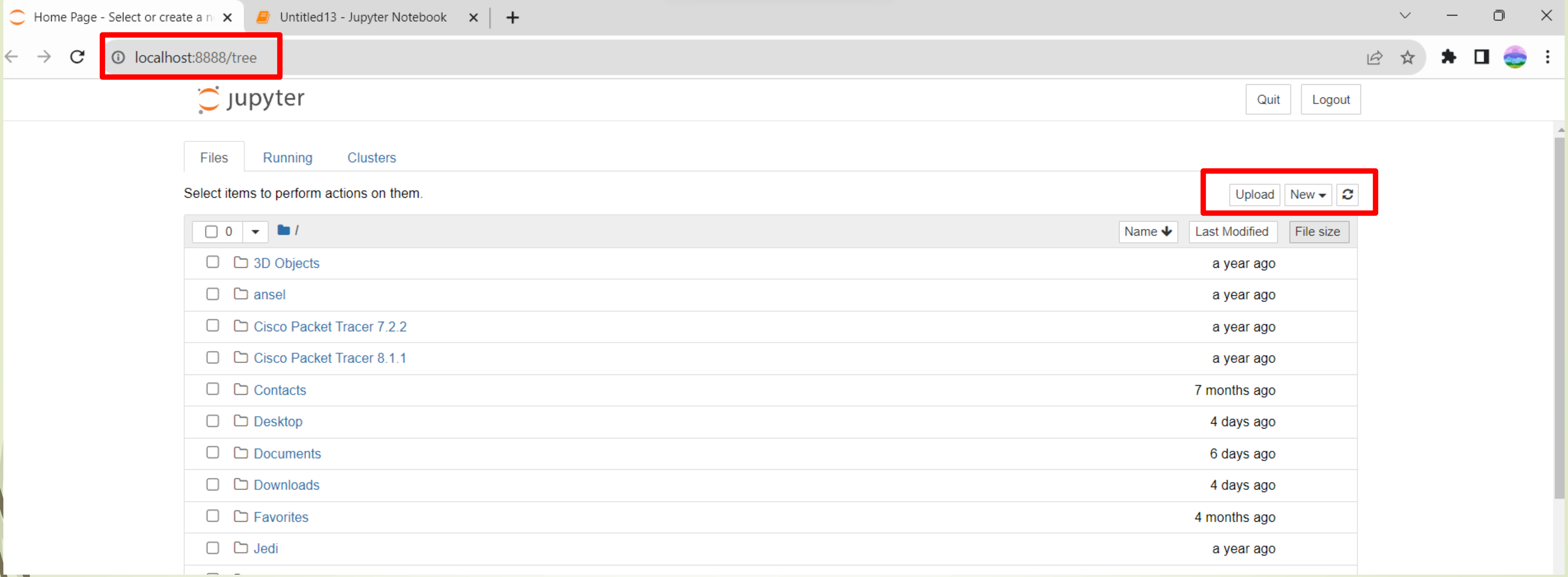
- **NumPy**
- **Pandas**
- **SciPy**
- **Sci-Kit Learn**
- **Keras**
- **TensorFlow**
- **….**

Starting with Jupyter



```
(base) C:\Users\Paris>jupyter notebook
[W 11:21:34.304 NotebookApp] Loading JupyterLab as a classic notebook (v6) extension.
[C 11:21:34.304 NotebookApp] You must use Jupyter Server v1 to load JupyterLab as notebook extension. You have v2.5.0 in
stalled.
    You can fix this by executing:
        pip install -U "jupyter-server<2.0.0"
[I 11:21:34.312 NotebookApp] Serving notebooks from local directory: C:\Users\Paris
[I 11:21:34.312 NotebookApp] Jupyter Notebook 6.5.2 is running at:
[I 11:21:34.312 NotebookApp] http://localhost:8888/?token=45f891ef4ca065386bb96e430d66e0d01dd53483733de340
[I 11:21:34.312 NotebookApp]  or http://127.0.0.1:8888/?token=45f891ef4ca065386bb96e430d66e0d01dd53483733de340
[I 11:21:34.312 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 11:21:34.337 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///C:/Users/Paris/AppData/Roaming/jupyter/runtime/nbserver-15916-open.html
    Or copy and paste one of these URLs:
        http://localhost:8888/?token=45f891ef4ca065386bb96e430d66e0d01dd53483733de340
     or http://127.0.0.1:8888/?token=45f891ef4ca065386bb96e430d66e0d01dd53483733de340
[I 11:21:40.702 NotebookApp] Creating new notebook in
[I 11:21:42.107 NotebookApp] Kernel started: 814fe8a5-64c7-4a72-a216-a1be1b33d079, name: python3
[I 11:23:42.493 NotebookApp] Saving file at /Untitled13.ipynb
D:\Anaconda\lib\site-packages\nbformat\__init__.py:128: MissingIDFieldWarning: Code cell is missing an id field, this wi
ll become a hard error in future nbformat versions. You may want to use `normalize()` on your notebooks before validatio
ns (available since nbformat 5.1.4). Previous versions of nbformat are fixing this issue transparently, and will stop do
ing so in the future.
  validate(nb)
D:\Anaconda\lib\site-packages\notebook\services\contents\manager.py:353: MissingIDFieldWarning: Code cell is missing an
id field, this will become a hard error in future nbformat versions. You may want to use `normalize()` on your notebooks
 before validations (available since nbformat 5.1.4). Previous versions of nbformat are fixing this issue transparently,
```

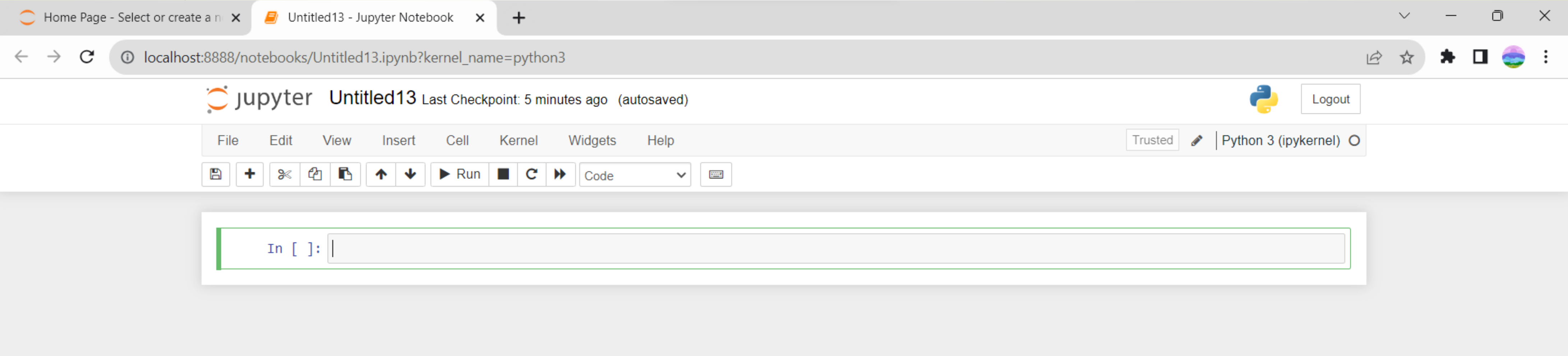
Starting with Jupyter



Select items to perform actions on them.

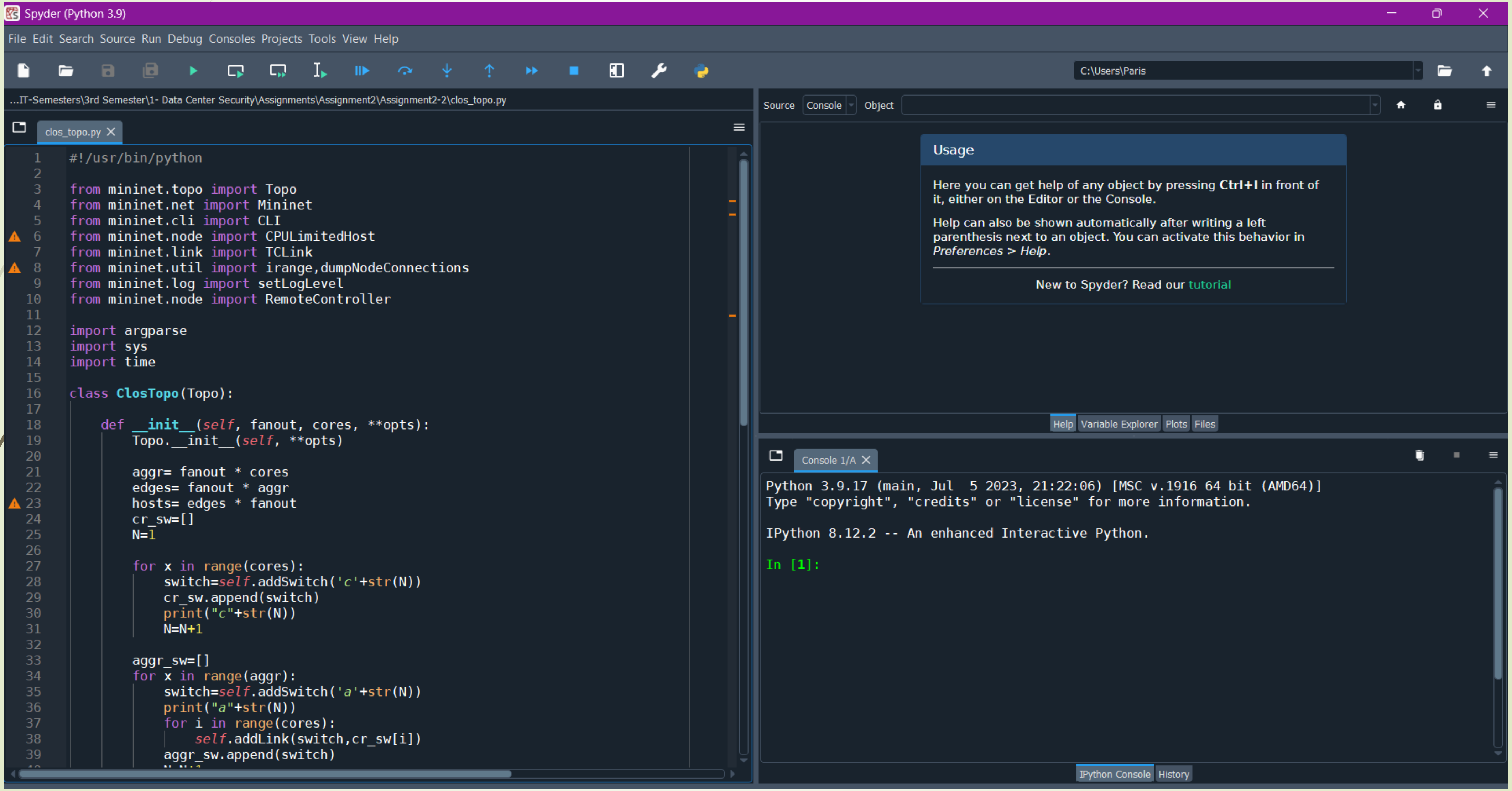
Name	Last Modified	File size
3D Objects	a year ago	
ansel	a year ago	
Cisco Packet Tracer 7.2.2	a year ago	
Cisco Packet Tracer 8.1.1	a year ago	
Contacts	7 months ago	
Desktop	4 days ago	
Documents	6 days ago	
Downloads	4 days ago	
Favorites	4 months ago	
Jedi	a year ago	

Starting with Jupyter


Home Page - Select or create a n
Untitled13 - Jupyter Notebook
localhost:8888/notebooks/Untitled13.ipynb?kernel_name=python3
jupyter
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3 (ipykernel)
Run
Code
In [ ]:

Untitled13 Last Checkpoint: 5 minutes ago (autosaved)

Starting with Spyder


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Paris
...IT-Semesters\3rd Semester\1- Data Center Security\Assignments\Assignment2\Assignment2-2\clos_topo.py
clos_topo.py
Source
Console
Object
Help
Variable Explorer
Plots
Files
Console 1/A
IPython Console
History

```python
#!/usr/bin/python

from mininet.topo import Topo
from mininet.net import Mininet
from mininet.cli import CLI
from mininet.node import CPULimitedHost
from mininet.link import TCLink
from mininet.util import irange,dumpNodeConnections
from mininet.log import setLogLevel
from mininet.node import RemoteController

import argparse
import sys
import time

class ClosTopo(Topo):

    def __init__(self, fanout, cores, **opts):
        Topo.__init__(self, **opts)

        aggr= fanout * cores
        edges= fanout * aggr
        hosts= edges * fanout
        cr_sw=[]
        N=1

        for x in range(cores):
            switch=self.addSwitch('c'+str(N))
            cr_sw.append(switch)
            print("c"+str(N))
            N=N+1

        aggr_sw=[]
        for x in range(aggr):
            switch=self.addSwitch('a'+str(N))
            print("a"+str(N))
            for i in range(cores):
                self.addLink(switch,cr_sw[i])
            aggr_sw.append(switch)
```

Usage

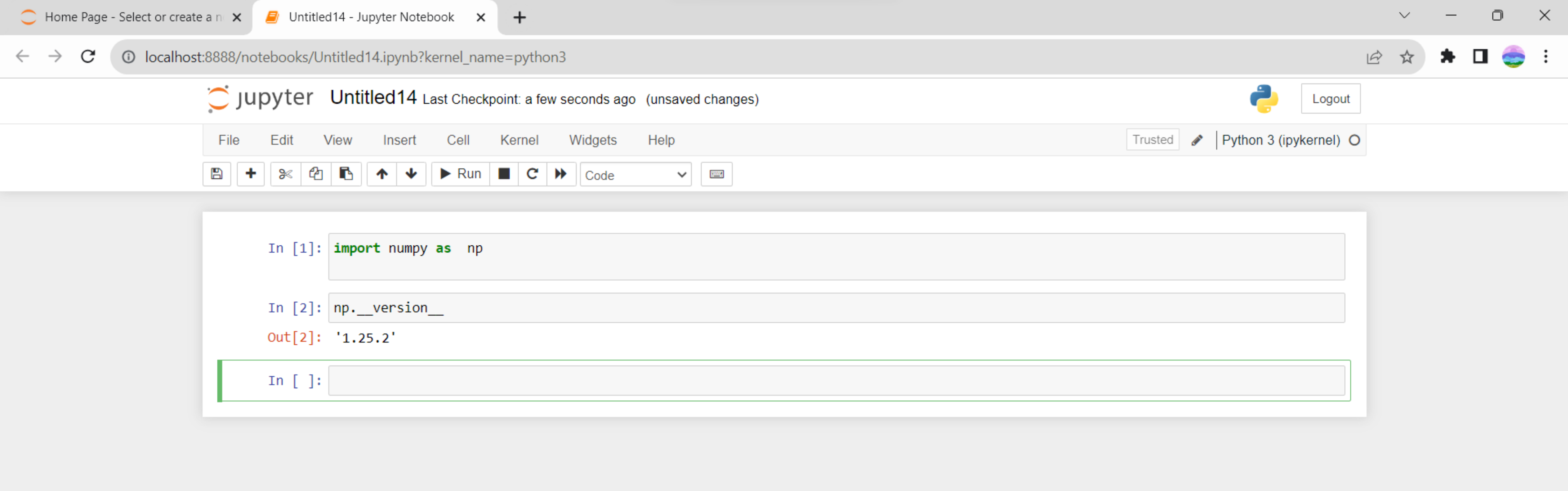
Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

```
Python 3.9.17 (main, Jul  5 2023, 21:22:06) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.12.2 -- An enhanced Interactive Python.

In [1]:
```

NumPy



Home Page - Select or create a n
Untitled14 - Jupyter Notebook
localhost:8888/notebooks/Untitled14.ipynb?kernel_name=python3
jupyter
Untitled14 Last Checkpoint: a few seconds ago (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3 (ipykernel) ○
Run
Code

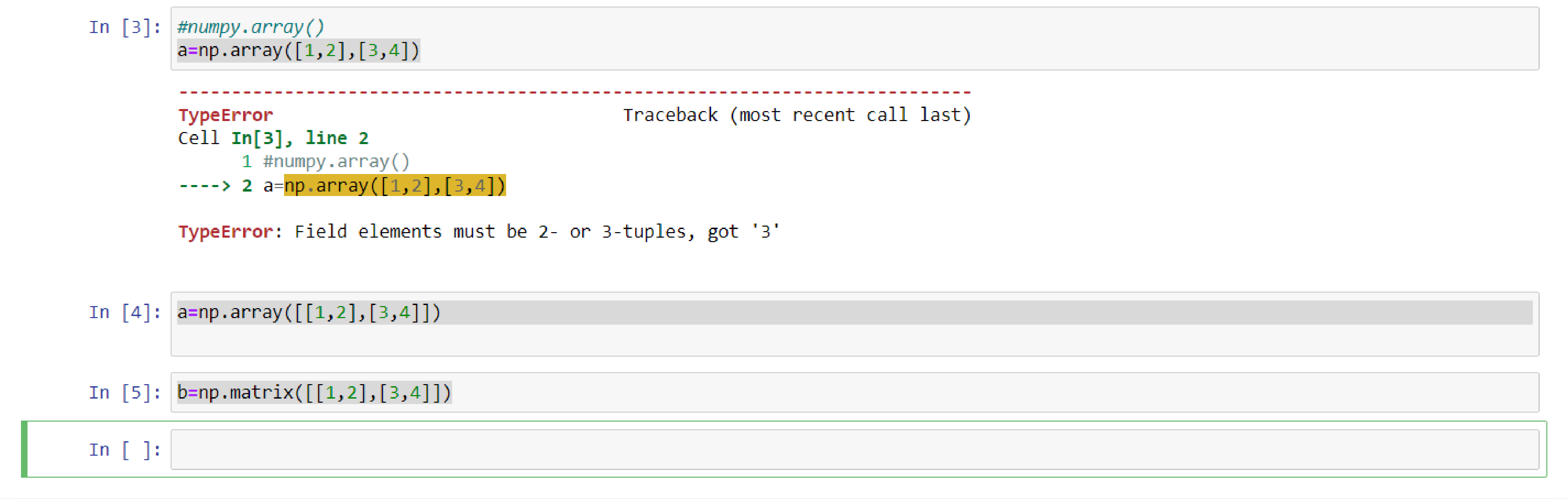
```
In [1]: import numpy as  np

In [2]: np.__version__
Out[2]: '1.25.2'

In [ ]:
```

Arrays and Matrices in NumPy

- The **NumPy** array is one of the most versatile data structures in Python and it is the foundation of most Python-based data science and machine learning applications.
- NumPy **matrices** are strictly 2-dimensional, while NumPy **arrays** (ndarrays) are N-dimensional.
- A **Python array** is a collection of elements of the **same data type**. Each element in the array has a **unique index** and is stored at a contiguous location in memory.
- Contrary to an array, a list does not constrain you to one data type. For example, you can store a string, an integer, and a Boolean in a list.
- Matrices and vectors with more than one dimensions are usually represented as multidimensional arrays in Python.
- A NumPy 2D array in Python looks like a list nested within a list.

https://programmathically.com/matrix-and-array-in-python-numpy/
https://stackoverflow.com/questions/4151128/what-are-the-differences-between-numpy-arrays-and-matrices-which-one-should-i-u

NumPy



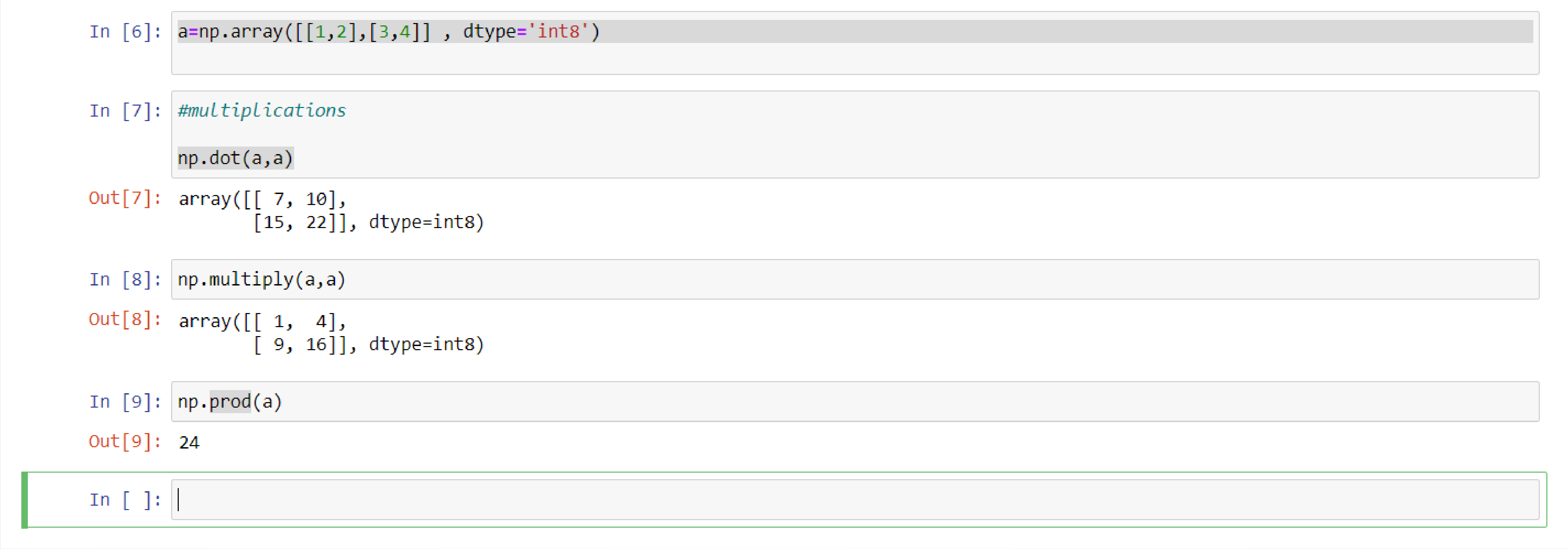
```
In [3]: #numpy.array()
        a=np.array([1,2],[3,4])

        ---------------------------------------------------------------------------
        TypeError                                 Traceback (most recent call last)
        Cell In[3], line 2
              1 #numpy.array()
        ----> 2 a=np.array([1,2],[3,4])

        TypeError: Field elements must be 2- or 3-tuples, got '3'

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

In [5]: b=np.matrix([[1,2],[3,4]])

In [ ]:
```

$$a=\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}$$

Arrays in Python can have multiple dimensions while matrices are only two-dimensional.

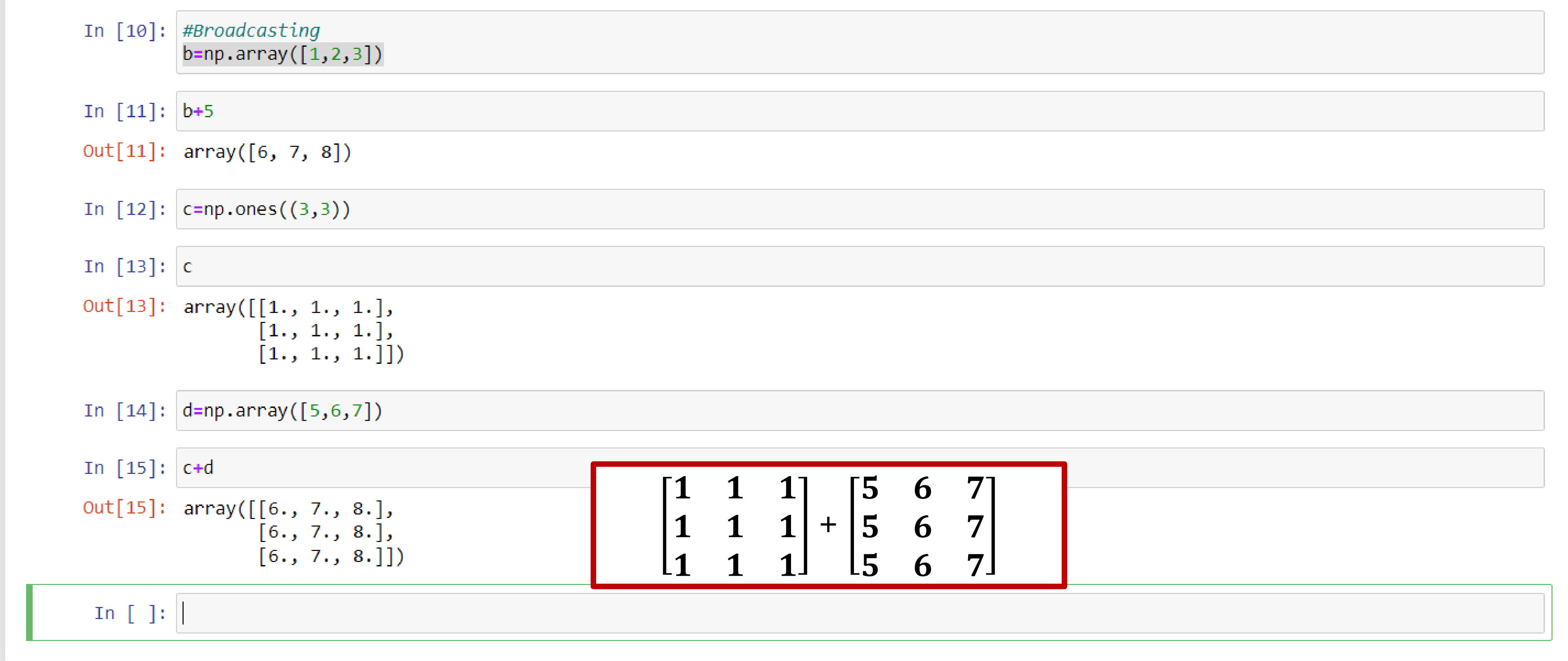
Multiplications

1- np.dot: Matrix multiplications
2-np.multiply: Multiply member to member
3-np.prod: multiply all members together to produce one single number.



```
In [6]: a=np.array([[1,2],[3,4]] , dtype='int8')

In [7]: #multiplications

        np.dot(a,a)

Out[7]: array([[ 7, 10],
               [15, 22]], dtype=int8)

In [8]: np.multiply(a,a)

Out[8]: array([[ 1,  4],
               [ 9, 16]], dtype=int8)

In [9]: np.prod(a)

Out[9]: 24

In [ ]:
```

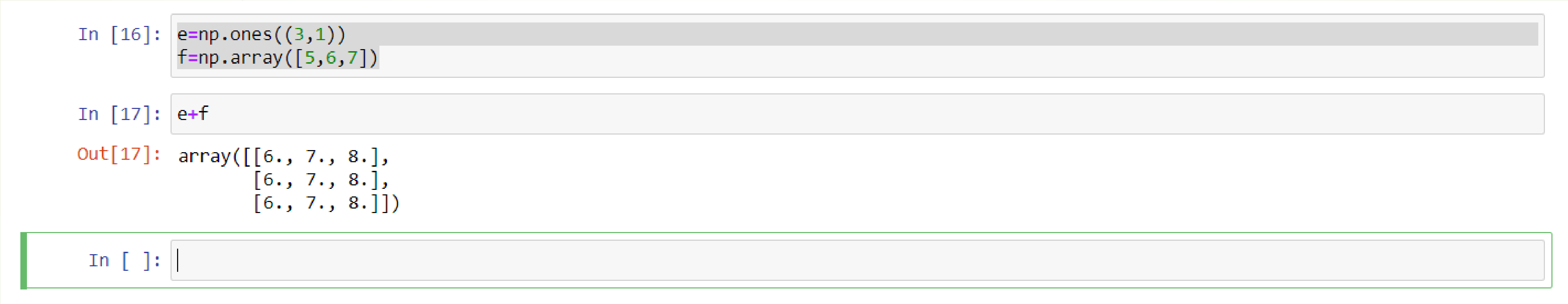
Broadcasting



```
In [10]: #Broadcasting
         b=np.array([1,2,3])

In [11]: b+5

Out[11]: array([6, 7, 8])

In [12]: c=np.ones((3,3))

In [13]: c

Out[13]: array([[1., 1., 1.],
                [1., 1., 1.],
                [1., 1., 1.]])

In [14]: d=np.array([5,6,7])

In [15]: c+d

Out[15]: array([[6., 7., 8.],
                [6., 7., 8.],
                [6., 7., 8.]])

In [ ]:
```

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} + \begin{bmatrix} 5 & 6 & 7 \\ 5 & 6 & 7 \\ 5 & 6 & 7 \end{bmatrix}$$

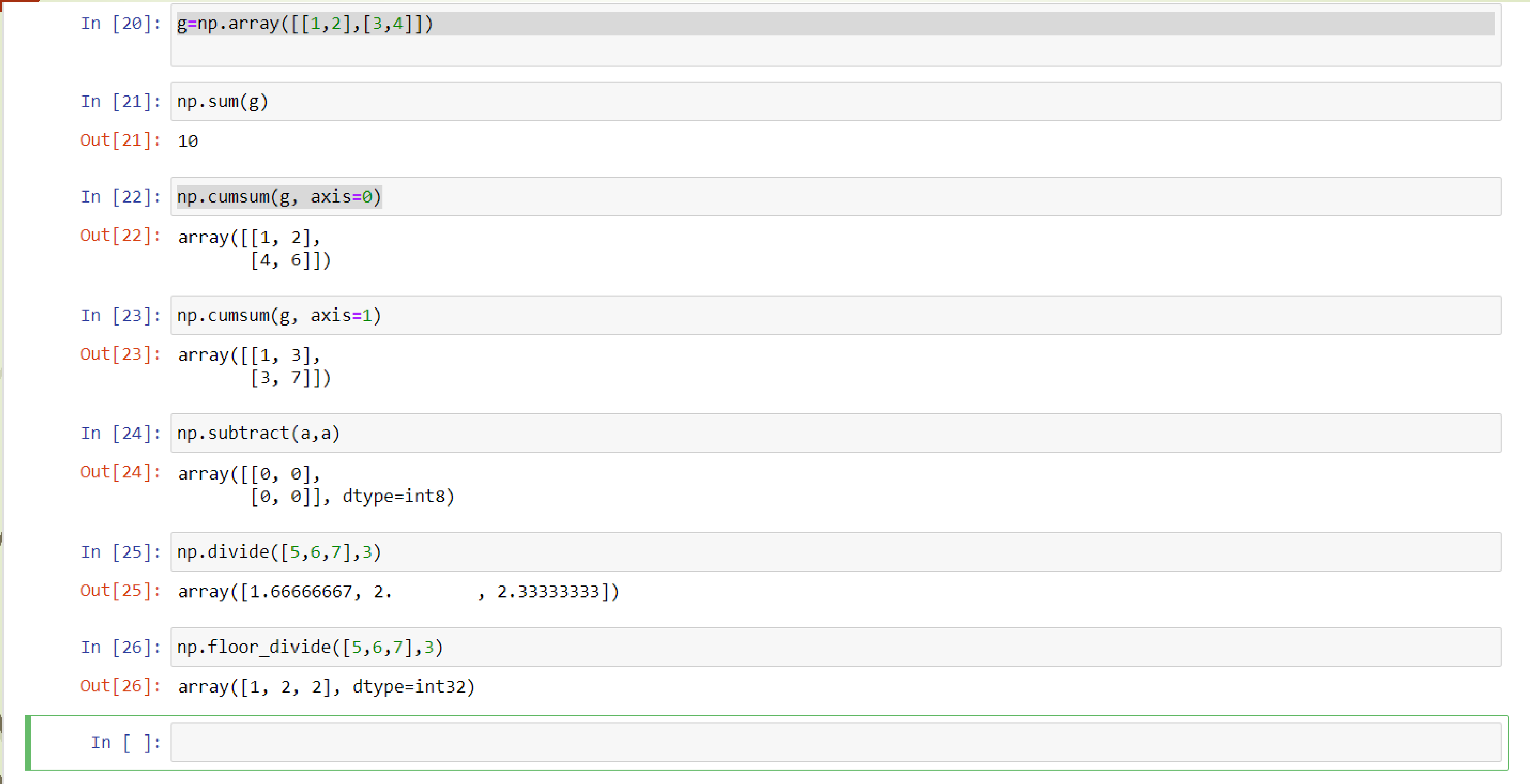
Broadcasting



```
In [16]: e=np.ones((3,1))
         f=np.array([5,6,7])

In [17]: e+f

Out[17]: array([[6., 7., 8.],
                [6., 7., 8.],
                [6., 7., 8.]])

In [ ]:
```

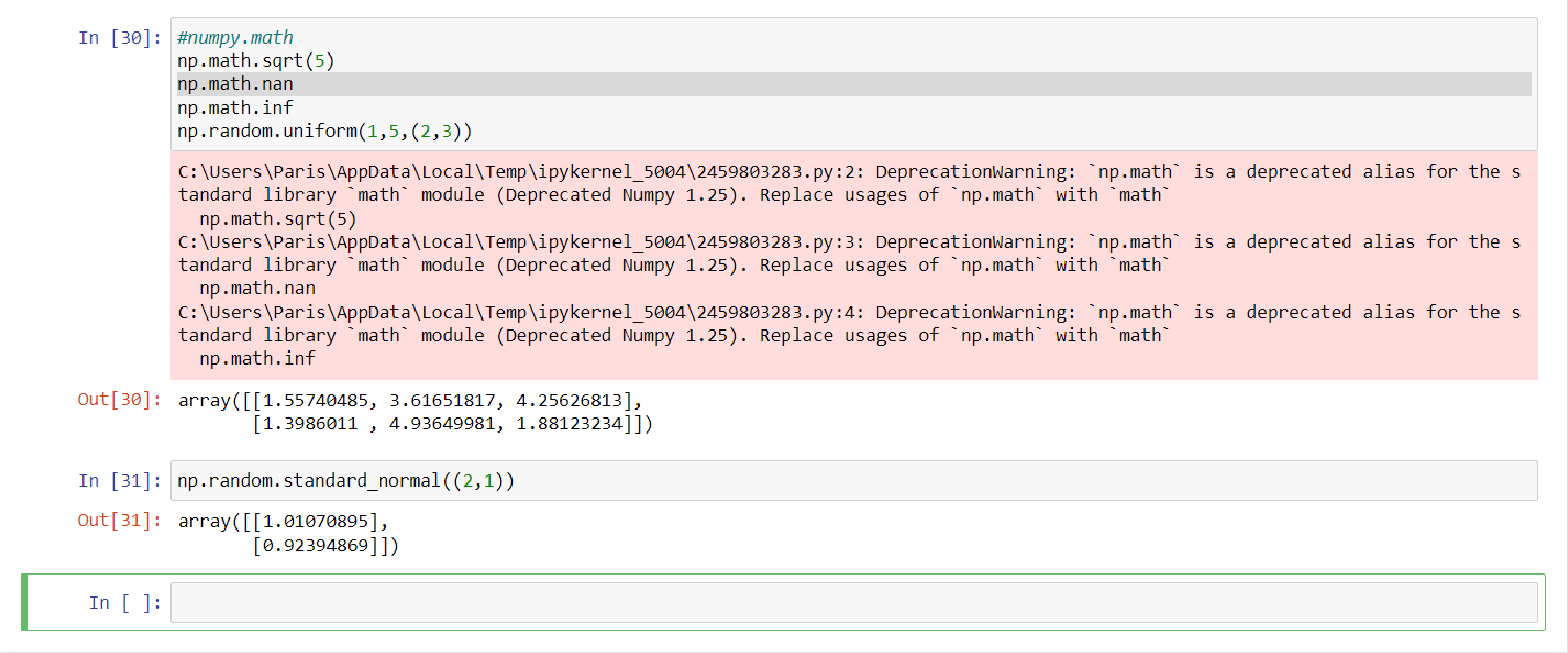
$$\begin{bmatrix}1\\1\\1\end{bmatrix}+\begin{bmatrix}5 & 6 & 7\end{bmatrix}=\begin{bmatrix}1 & 1 & 1\\1 & 1 & 1\\1 & 1 & 1\end{bmatrix}+\begin{bmatrix}5 & 6 & 7\\5 & 6 & 7\\5 & 6 & 7\end{bmatrix}$$

Basic Math Operations



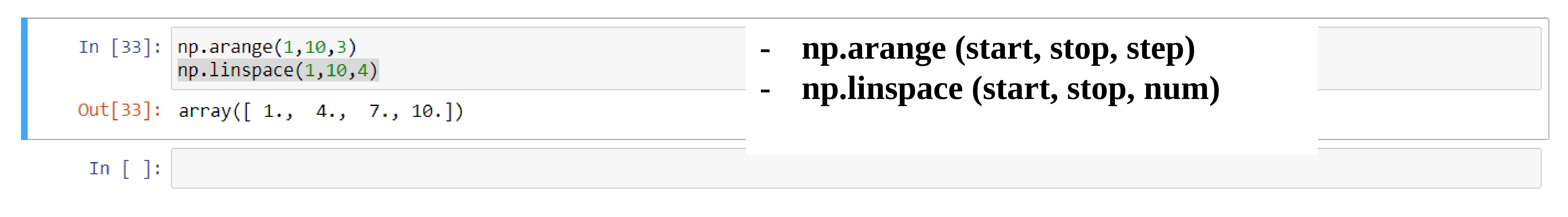
```
In [20]: g=np.array([[1,2],[3,4]])

In [21]: np.sum(g)

Out[21]: 10

In [22]: np.cumsum(g, axis=0)

Out[22]: array([[1, 2],
                [4, 6]])

In [23]: np.cumsum(g, axis=1)

Out[23]: array([[1, 3],
                [3, 7]])

In [24]: np.subtract(a,a)

Out[24]: array([[0, 0],
                [0, 0]], dtype=int8)

In [25]: np.divide([5,6,7],3)

Out[25]: array([1.66666667, 2.        , 2.33333333])

In [26]: np.floor_divide([5,6,7],3)

Out[26]: array([1, 2, 2], dtype=int32)

In [ ]:
```

Basic Math Operations



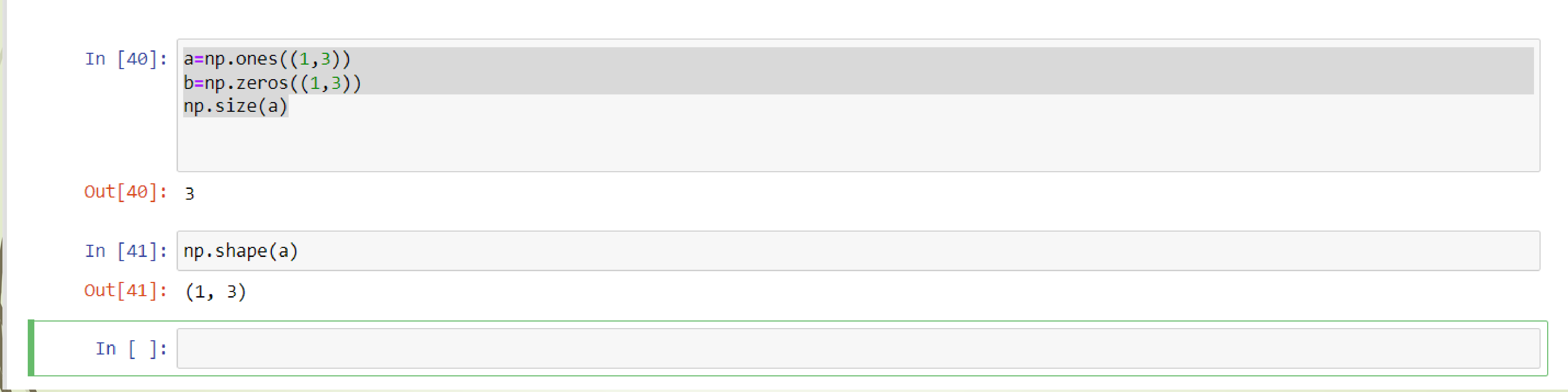
```
In [30]: #numpy.math
         np.math.sqrt(5)
         np.math.nan
         np.math.inf
         np.random.uniform(1,5,(2,3))

         C:\Users\Paris\AppData\Local\Temp\ipykernel_5004\2459803283.py:2: DeprecationWarning: `np.math` is a deprecated alias for the s
         tandard library `math` module (Deprecated Numpy 1.25). Replace usages of `np.math` with `math`
           np.math.sqrt(5)
         C:\Users\Paris\AppData\Local\Temp\ipykernel_5004\2459803283.py:3: DeprecationWarning: `np.math` is a deprecated alias for the s
         tandard library `math` module (Deprecated Numpy 1.25). Replace usages of `np.math` with `math`
           np.math.nan
         C:\Users\Paris\AppData\Local\Temp\ipykernel_5004\2459803283.py:4: DeprecationWarning: `np.math` is a deprecated alias for the s
         tandard library `math` module (Deprecated Numpy 1.25). Replace usages of `np.math` with `math`
           np.math.inf

Out[30]: array([[1.55740485, 3.61651817, 4.25626813],
                [1.3986011 , 4.93649981, 1.88123234]])

In [31]: np.random.standard_normal((2,1))

Out[31]: array([[1.01070895],
                [0.92394869]])

In [ ]:
```

Basic Math Operations

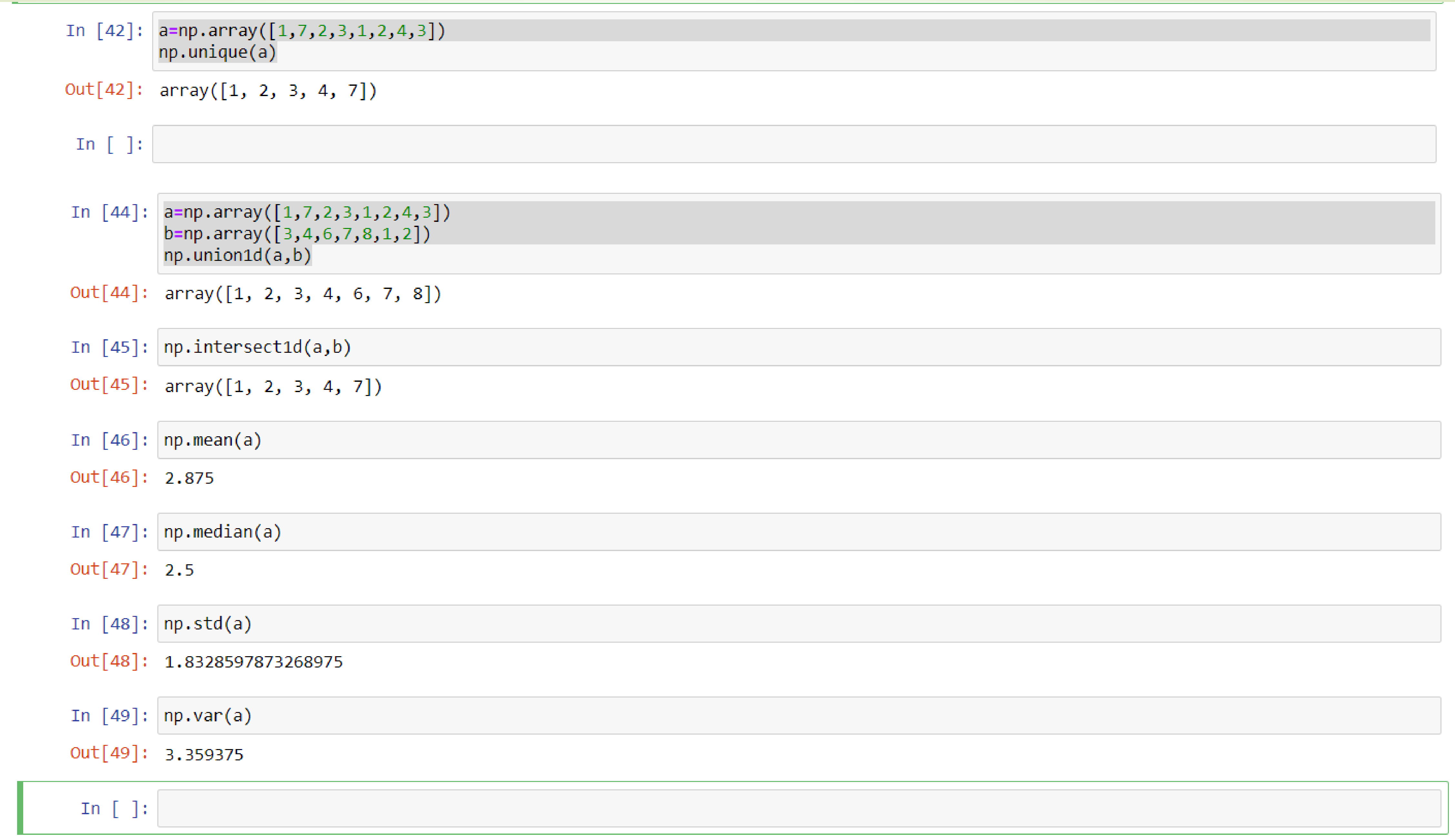


```
In [33]: np.arange(1,10,3)
         np.linspace(1,10,4)

Out[33]: array([ 1.,  4.,  7., 10.])

In [ ]:
```

- np.arange (start, stop, step)
- np.linspace (start, stop, num)

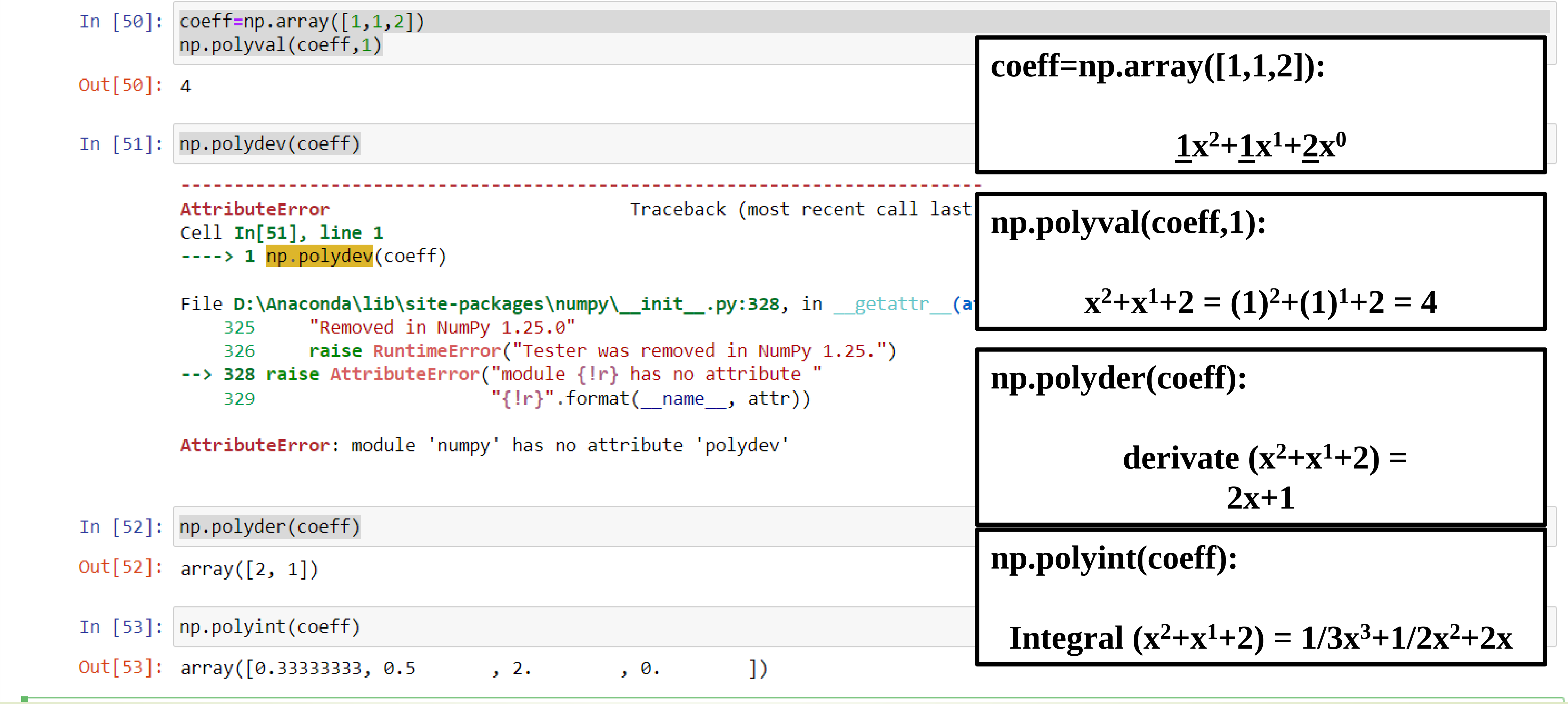


```
In [40]: a=np.ones((1,3))
         b=np.zeros((1,3))
         np.size(a)

Out[40]: 3

In [41]: np.shape(a)

Out[41]: (1, 3)

In [ ]:
```



```
In [42]: a=np.array([1,7,2,3,1,2,4,3])
         np.unique(a)

Out[42]: array([1, 2, 3, 4, 7])

In [ ]:

In [44]: a=np.array([1,7,2,3,1,2,4,3])
         b=np.array([3,4,6,7,8,1,2])
         np.union1d(a,b)

Out[44]: array([1, 2, 3, 4, 6, 7, 8])

In [45]: np.intersect1d(a,b)

Out[45]: array([1, 2, 3, 4, 7])

In [46]: np.mean(a)

Out[46]: 2.875

In [47]: np.median(a)

Out[47]: 2.5

In [48]: np.std(a)

Out[48]: 1.8328597873268975

In [49]: np.var(a)

Out[49]: 3.359375

In [ ]:
```

Polynomials



```
In [50]: coeff=np.array([1,1,2])
         np.polyval(coeff,1)

Out[50]: 4

In [51]: np.polydev(coeff)

         ---------------------------------------------------------------------------
         AttributeError                            Traceback (most recent call last
         Cell In[51], line 1
         ----> 1 np.polydev(coeff)

         File D:\Anaconda\lib\site-packages\numpy\__init__.py:328, in __getattr__(a
             325     "Removed in NumPy 1.25.0"
             326     raise RuntimeError("Tester was removed in NumPy 1.25.")
         --> 328 raise AttributeError("module {!r} has no attribute "
             329                      "{!r}".format(__name__, attr))

         AttributeError: module 'numpy' has no attribute 'polydev'

In [52]: np.polyder(coeff)

Out[52]: array([2, 1])

In [53]: np.polyint(coeff)

Out[53]: array([0.33333333, 0.5       , 2.        , 0.        ])
```

coeff=np.array([1,1,2]):

$\underline{1}x^2+\underline{1}x^1+\underline{2}x^0$

np.polyval(coeff,1):

$x^2+x^1+2 = (1)^2+(1)^1+2 = 4$

np.polyder(coeff):

derivate $(x^2+x^1+2) =$
$2x+1$

np.polyint(coeff):

Integral $(x^2+x^1+2) = 1/3x^3+1/2x^2+2x$