Python for Engineers Numpy and Matplotlib

Contents

🚦 No AI use.

Time required: 90 minutes

NumPy Tutorials

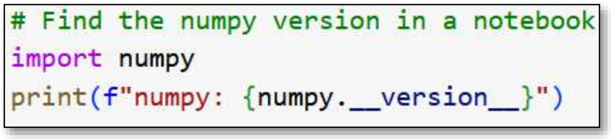
As you go through these tutorials, put notes and code into your Learning Journal.

- https://www.w3schools.com/python/numpy/default.asp
- https://www.w3schools.com/python/numpy/numpy_intro.asp
- https://www.w3schools.com/python/numpy/numpy_getting_started.asp
- https://www.w3schools.com/python/numpy/numpy_creating_arrays.asp
- https://www.w3schools.com/python/numpy/numpy_array_indexing.asp
- https://www.w3schools.com/python/numpy/numpy_array_slicing.asp
- https://www.w3schools.com/python/numpy/numpy_array_shape.asp
- https://www.w3schools.com/python/numpy/numpy_array_reshape.asp
- https://www.w3schools.com/python/numpy/numpy_array_iterating.asp
- https://www.w3schools.com/python/numpy/numpy_array_sort.asp
- https://www.w3schools.com/python/numpy/numpy_random.asp

Tutorial 1: Getting Started with NumPy

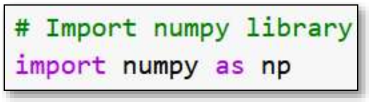
Type the code in each screenshot into Colab.

NumPy is a popular library for storing arrays of numbers and performing computations on them. This enables us to write often more succinct code, this also makes the code faster, since most NumPy routines are implemented in C or C++ for speed.



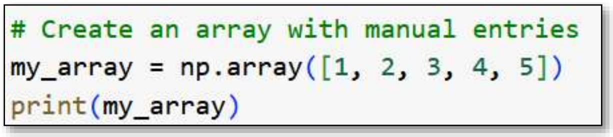
```python
# Find the numpy version in a notebook
import numpy
print(f"numpy: {numpy.__version__}")
```

To use NumPy in your program, import it as follows. The alias np is commonly used for NumPy to make it easier to reference the library in your code.



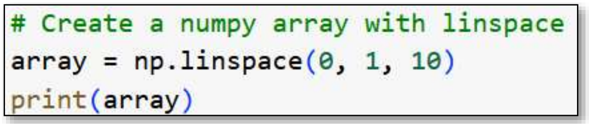
```python
# Import numpy library
import numpy as np
```

Arrays in NumPy are equivalent to vectors in MATLAB.



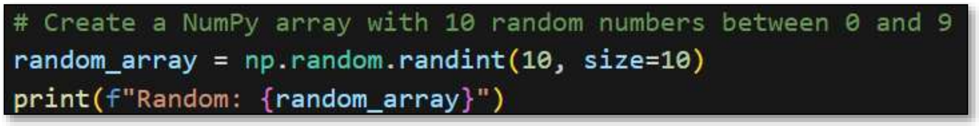
```python
# Create an array with manual entries
my_array = np.array([1, 2, 3, 4, 5])
print(my_array)
```

Create an array using linspace. This code generates an array of 10 evenly spaced values between 0 and 1.



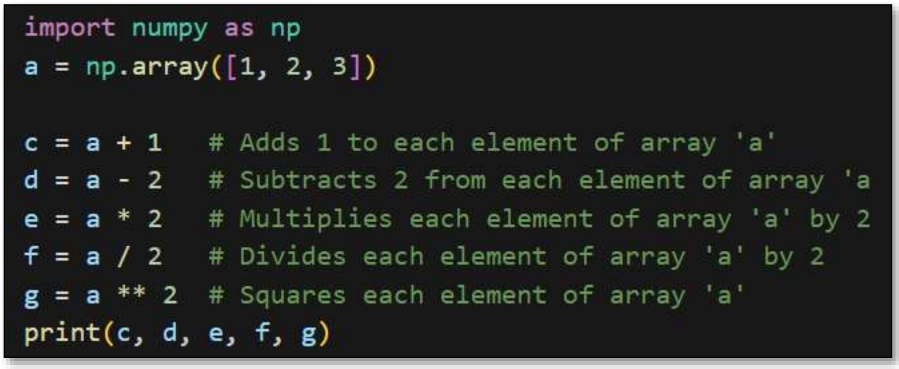
```python
# Create a numpy array with linspace
array = np.linspace(0, 1, 10)
print(array)
```

Create a NumPy array with random integers.



```python
# Create a NumPy array with 10 random numbers between 0 and 9
random_array = np.random.randint(10, size=10)
print(f"Random: {random_array}")
```

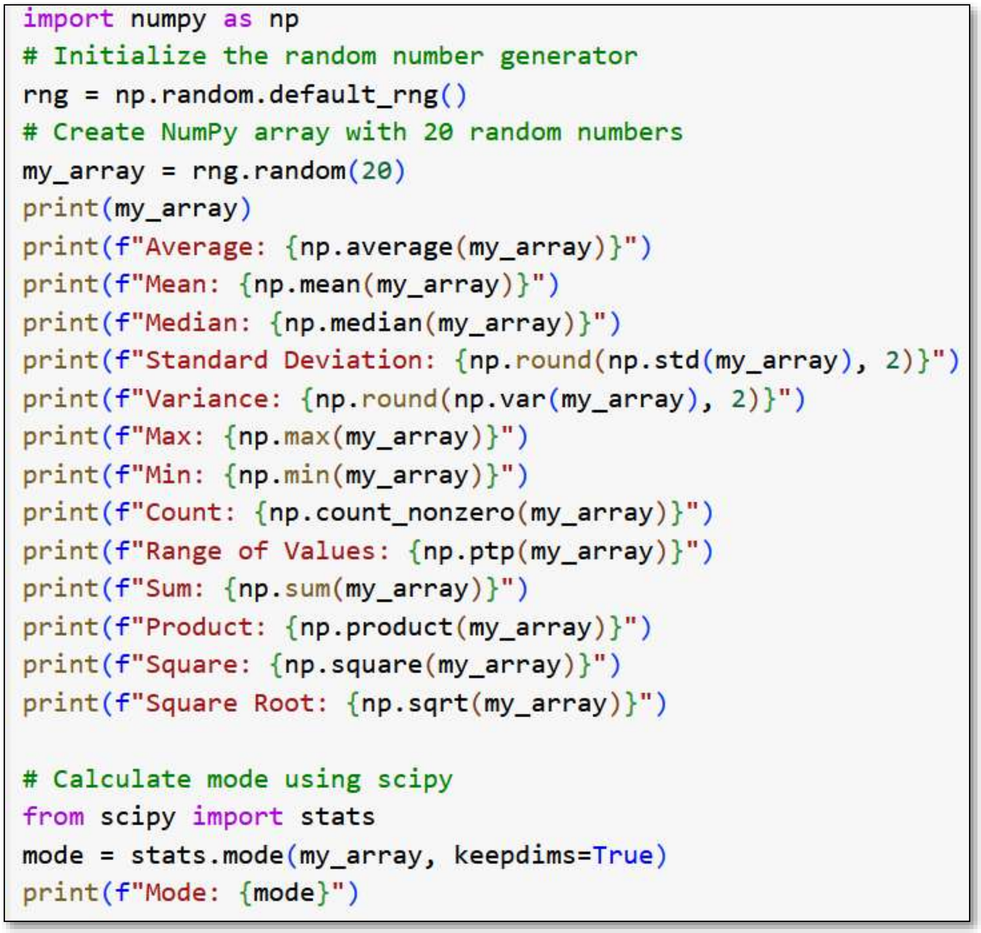
Math operations are the same in NumPy. They operate simultaneously on each element of the array.



```
import numpy as np
a = np.array([1, 2, 3])

c = a + 1   # Adds 1 to each element of array 'a'
d = a - 2   # Subtracts 2 from each element of array 'a
e = a * 2   # Multiplies each element of array 'a' by 2
f = a / 2   # Divides each element of array 'a' by 2
g = a ** 2  # Squares each element of array 'a'
print(c, d, e, f, g)
```

Math functions in NumPy.

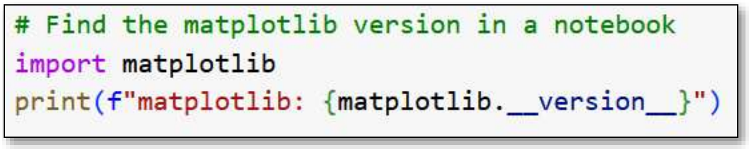


```
import numpy as np
# Initialize the random number generator
rng = np.random.default_rng()
# Create NumPy array with 20 random numbers
my_array = rng.random(20)
print(my_array)
print(f"Average: {np.average(my_array)}")
print(f"Mean: {np.mean(my_array)}")
print(f"Median: {np.median(my_array)}")
print(f"Standard Deviation: {np.round(np.std(my_array), 2)}")
print(f"Variance: {np.round(np.var(my_array), 2)}")
print(f"Max: {np.max(my_array)}")
print(f"Min: {np.min(my_array)}")
print(f"Count: {np.count_nonzero(my_array)}")
print(f"Range of Values: {np.ptp(my_array)}")
print(f"Sum: {np.sum(my_array)}")
print(f"Product: {np.product(my_array)}")
print(f"Square: {np.square(my_array)}")
print(f"Square Root: {np.sqrt(my_array)}")

# Calculate mode using scipy
from scipy import stats
mode = stats.mode(my_array, keepdims=True)
print(f"Mode: {mode}")
```

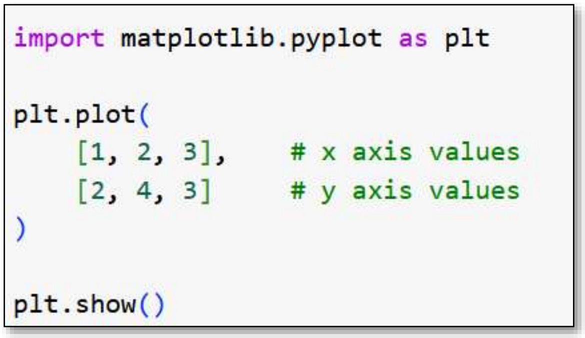
Tutorial 2: Matplotlib

Type the code in each screenshot into Colab.



```python
# Find the matplotlib version in a notebook
import matplotlib
print(f"matplotlib: {matplotlib.__version__}")
```

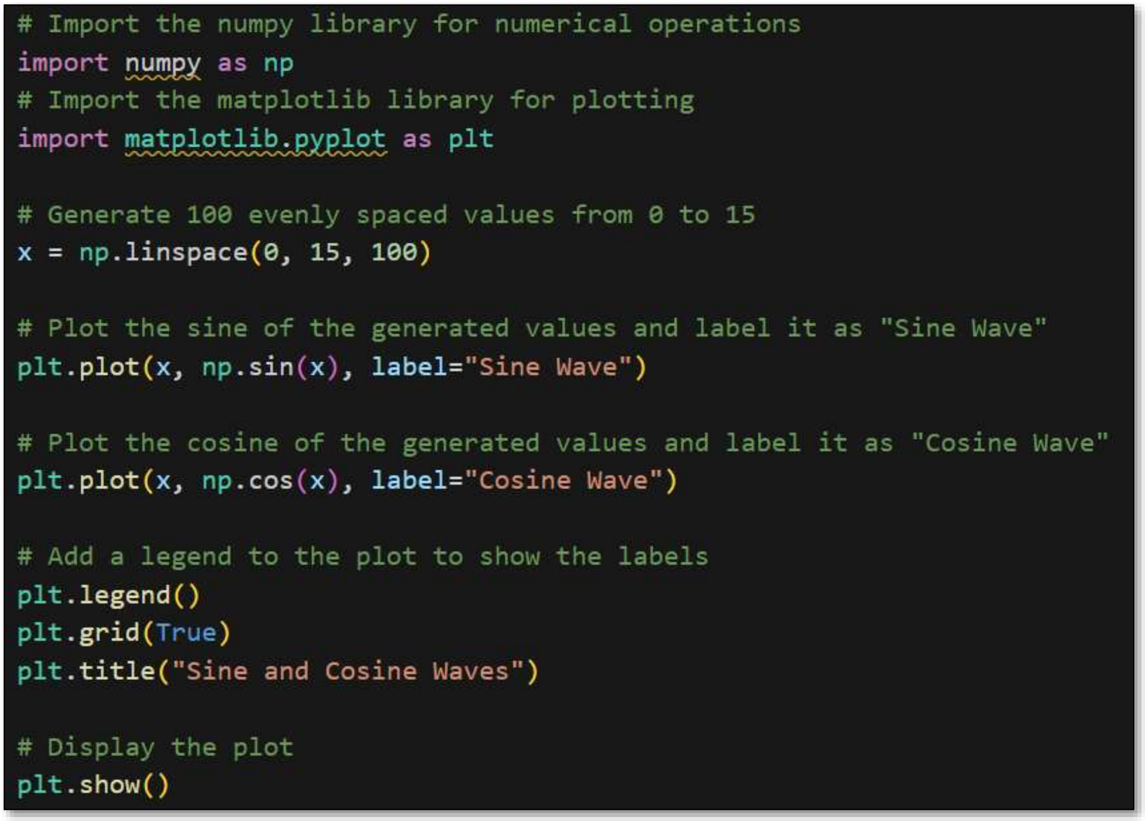
The line plot is kind of the “Hello World” of matplotlib. The following code shows how to chart 2 lists with a very simple line graph using the x and y-axis.



```python
import matplotlib.pyplot as plt

plt.plot(
    [1, 2, 3],     # x axis values
    [2, 4, 3]      # y axis values
)

plt.show()
```



```python
# Import the numpy library for numerical operations
import numpy as np
# Import the matplotlib library for plotting
import matplotlib.pyplot as plt

# Generate 100 evenly spaced values from 0 to 15
x = np.linspace(0, 15, 100)

# Plot the sine of the generated values and label it as "Sine Wave"
plt.plot(x, np.sin(x), label="Sine Wave")

# Plot the cosine of the generated values and label it as "Cosine Wave"
plt.plot(x, np.cos(x), label="Cosine Wave")

# Add a legend to the plot to show the labels
plt.legend()
plt.grid(True)
plt.title("Sine and Cosine Waves")

# Display the plot
plt.show()
```

Assignment Submission

1. In Google Colab → Click the Share button in the upper right hand side.

    a. Change General Access → Anyone with the link → Click Copy link.

2. Submit in Blackboard.