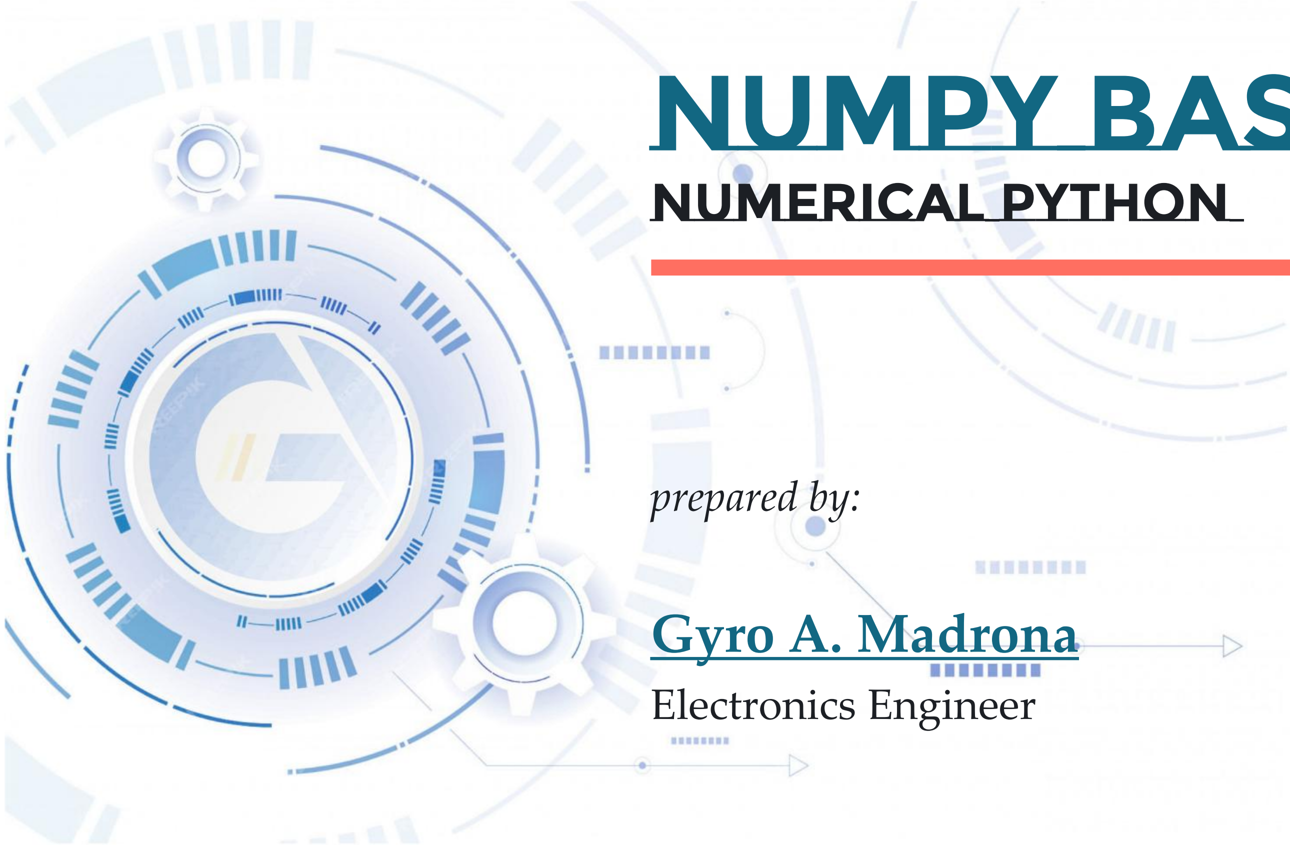
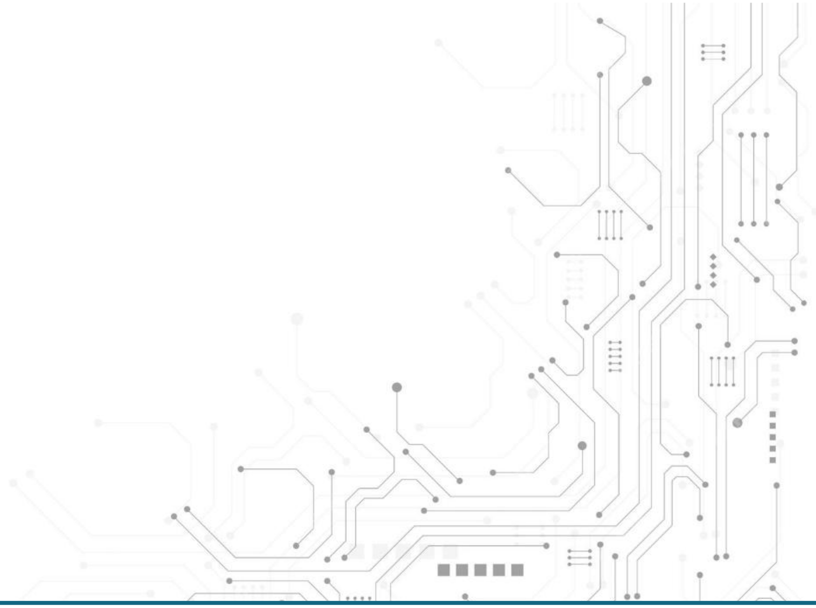


NUMPY BASICS

NUMERICAL PYTHON

prepared by:

Gyro A. Madrona
Electronics Engineer



@gyromadrona @gyro-madrona @OrbitGNX @OrbitGNX @gyronics

TOPIC OUTLINE

Numpy

N-D Array

Descriptive Statistics Functions

Axis Parameter

NUMPY

NUMPY

NumPy, short for Numerical Python, is a foundational library for **numerical computing** in Python, providing support for efficient arrays, matrices, and mathematical operations. It serves as the backbone for many other scientific libraries, including pandas, scipy, and matplotlib.

https://numpy.org

NUMPY PACKAGE

To load NumPy package:

```
import numpy as np
```

The community agreed alias for NumPy is **np**, so loading NumPy as np is assumed standard practice for all of the NumPy documentation.

NumPy

https://numpy.org

N-D ARRAY

An **n-dimensional array** is a versatile and powerful data structure that allows you to work with multi-dimensional data efficiently.

Multi-dimensional:

- 0D array: Scalar
- 1D array: Vector
- 2D array: Matrix
- 3D array: Tensor
- General n-dimensional arrays

1-D ARRAY

Examples:

```
array_a = np.array([1,2,3)]

array_b = np.array([4,5,6)]
```

Syntax to create 1-D array:

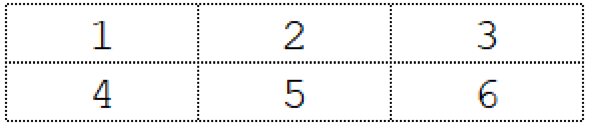
```
array_name = np.array([elements])
```

2-D ARRAY

Syntax to create 2-D array:

```
array_name = np.array([
    [row1_elements],
    [row2_elements],
    ...,
    [rown_elements]
    ])
```

Example:

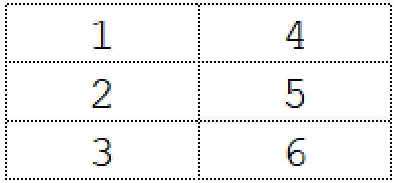


1	2	3
4	5	6

```
my_array = np.array([
    [1,2,3],[4,5,6]])

# Transpose array

my_array.T
```



1	4
2	5
3	6

DESCRIPTIVE STATISTICS

```
np.mean(array)

np.median(array)

stats.mode(array) #from scipy import stats


np.min(array)

np.max(array)

np.percentile(array,25) #1st quartile

np.percentile(array,25) #2nd quartile

np.var(array)

np.std(array)
```

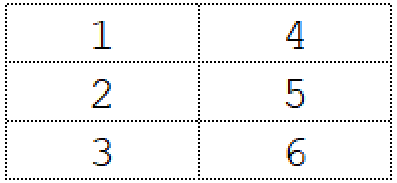
AXIS

The **axis** parameter defines the direction in which an operation (e.g., **sum**, **mean** ,**median**) is performed.

For a 2-D array:

- **axis = 0** refers to rows (vertical direction)
- **axis = 1** refers to columns (horizontal direction)

Example:



1	4
2	5
3	6

```
np.sum(array,axis=0)

# output:[6, 15]


np.sum(array,axis=1)

# output:[5, 7, 9]
```

LABORATORY