



# Introduction to NumPy

# Profile



 [linkedin.com/in/romansyasetyo/](https://www.linkedin.com/in/romansyasetyo/)

# Table of Content

## What will We Learn Today?

1. What is NumPy
2. NumPy usage
3. NumPy in Python

Hands on using Google Colab





# What is NumPy







# What is NumPy

- **Numerical Python**
- NumPy is a fundamental package for scientific computing in Python
- Used for working with arrays
- Used in :
  - matrix manipulation and computation
  - sorting and selecting data
  - basic linear algebra
  - basic statistical and mathematical operations
  - and much more!





# Why use NumPy?

- In Python, **lists** serve the purpose of **arrays**, but they are slow to process. NumPy aims to provide an **array** object that is up to 50x faster than traditional Python **lists**
- **Arrays** are very frequently used in data science, where speed and resources are very important.

```
import numpy
```

```
my_list = [1,2,3,4]  
print(type(my_list))  
print(my_list)
```

```
<class 'list'>  
[1, 2, 3, 4]
```

```
my_array = numpy.array(my_list)  
print(type(my_array))  
print(my_array)
```

```
<class 'numpy.ndarray'>  
[1 2 3 4]
```



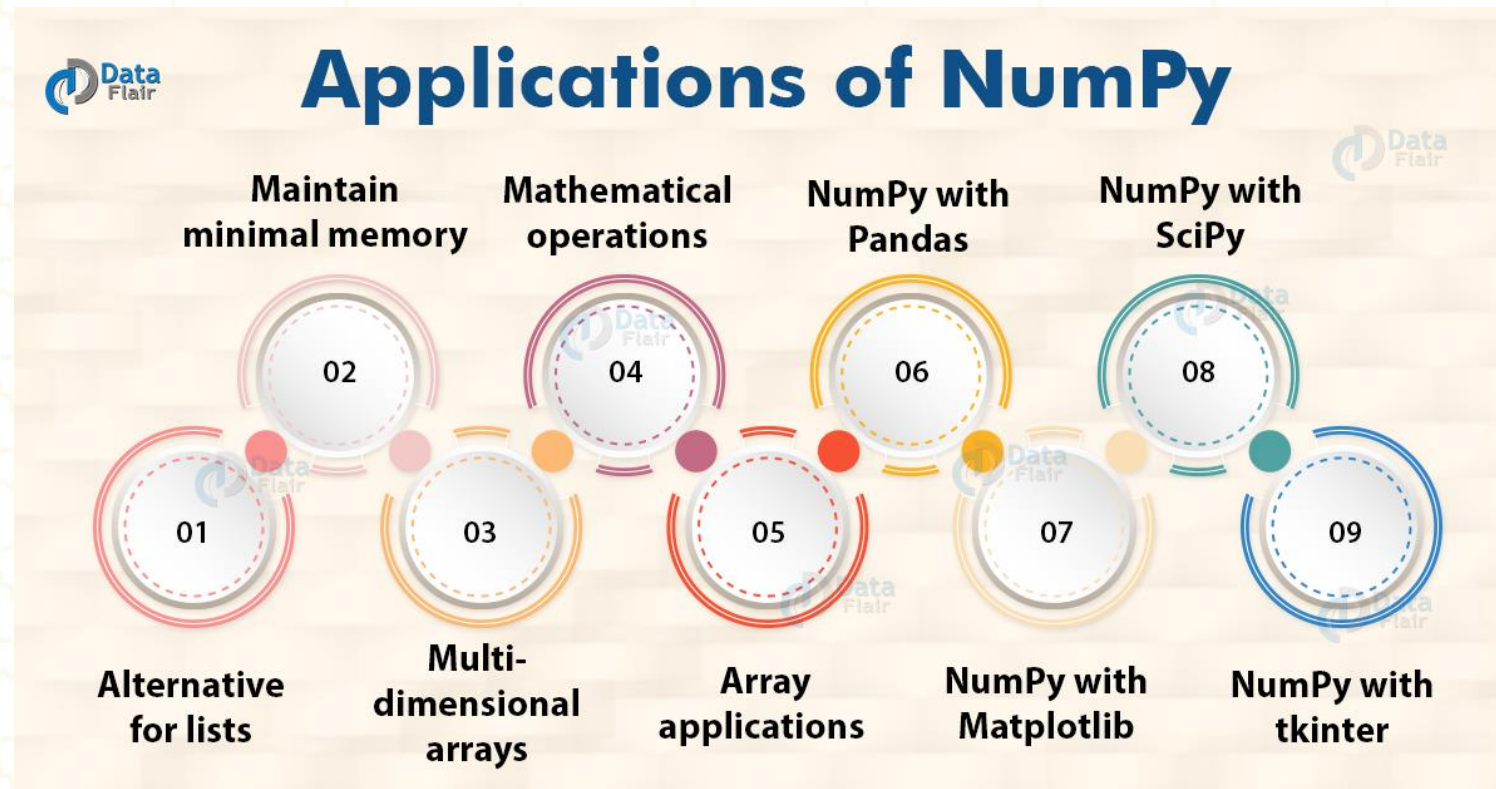


# Numpy Usage



















# NumPy Usage



Source : <https://data-flair.training/blogs/numpy-applications/>



# NumPy Usage

<b>Quantum Computing</b>  QuTiP PyQuil Qiskit	<b>Statistical Computing</b>  Pandas statsmodels Xarray Seaborn	<b>Signal Processing</b>  SciPy PyWavelets python-control	<b>Image Processing</b>  Scikit-image OpenCV Mahotas	<b>Graphs and Networks</b>  NetworkX graph-tool igraph PyGSP	<b>Astronomy Processes</b>  AstroPy SunPy SpacePy	<b>Cognitive Psychology</b>  PsychoPy
<b>Bioinformatics</b>  BioPython Scikit-Bio PyEnsembl ETE	<b>Bayesian Inference</b>  PyStan PyMC3 ArviZ emcee	<b>Mathematical Analysis</b>  SciPy SymPy cvxpy FEniCS	<b>Chemistry</b>  Cantera MDAnalysis RDKit	<b>Geoscience</b>  Pangeo Simpeg ObsPy Fatiando a Terra	<b>Geographic Processing</b>  Shapely GeoPandas Folium	<b>Architecture &amp; Engineering</b>  COMPAS City Energy Analyst Sverchok

Source : [numpy.org](https://numpy.org)



# Numpy in Python





# Using NumPy & Creating Array

```
!pip install numpy
```

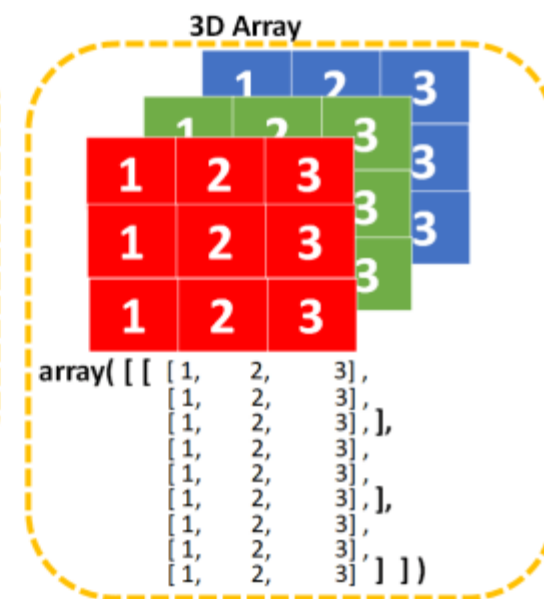
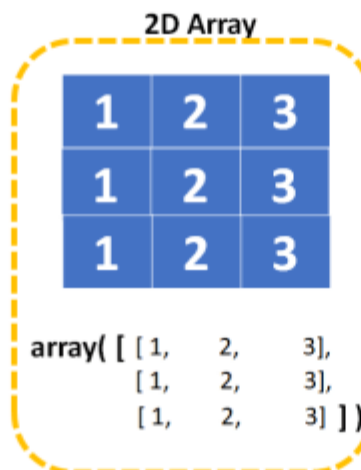
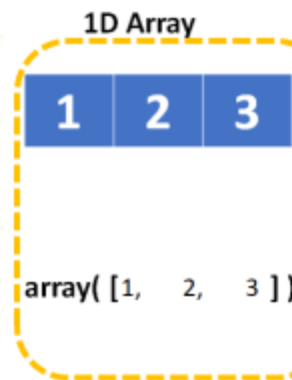
```
import numpy
```

```
my_list = [1,2,3,4]
print(type(my_list))
print(my_list)
```

```
<class 'list'>
[1, 2, 3, 4]
```

```
my_array = numpy.array(my_list)
print(type(my_array))
print(my_array)
```

```
<class 'numpy.ndarray'>
[1 2 3 4]
```





# Array Indexing & Reshaping

	Array Elements							
	10	20	30	40	50	60	70	80
Index ->	0	1	2	3	4	5	6	7
	-8	-7	-6	-5	-4	-3	-2	-1

**<- Negative Index**

**data**

1
2
3
4
5
6

**data.reshape(2,3)**

1	2	3
4	5	6

Dimensions: 2 (rows) x 3 (columns)

**data.reshape(3,2)**

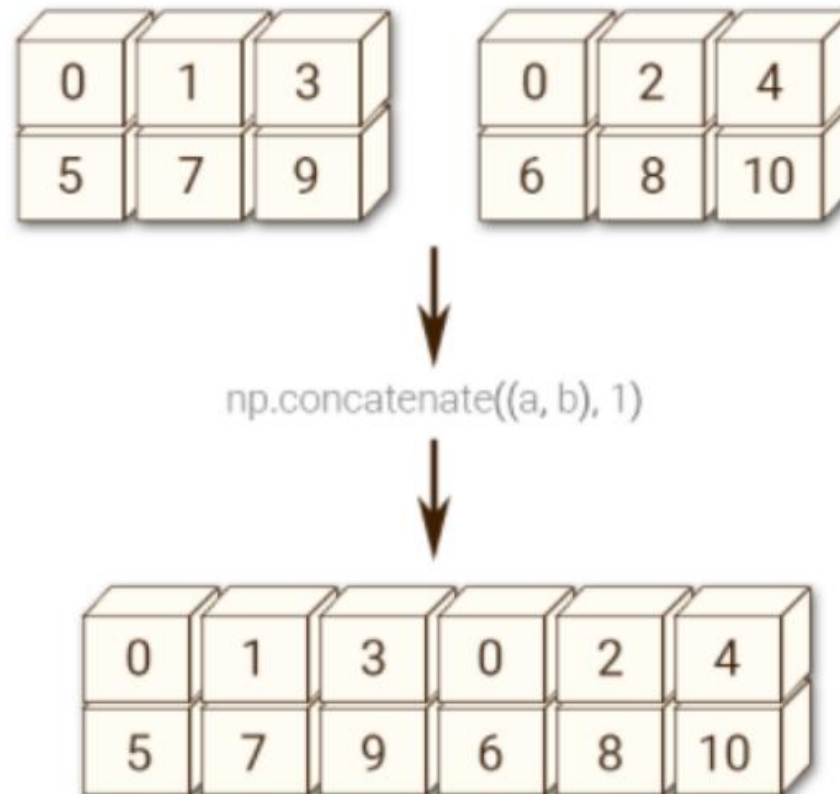
1	2
3	4
5	6

Dimensions: 3 (rows) x 2 (columns)

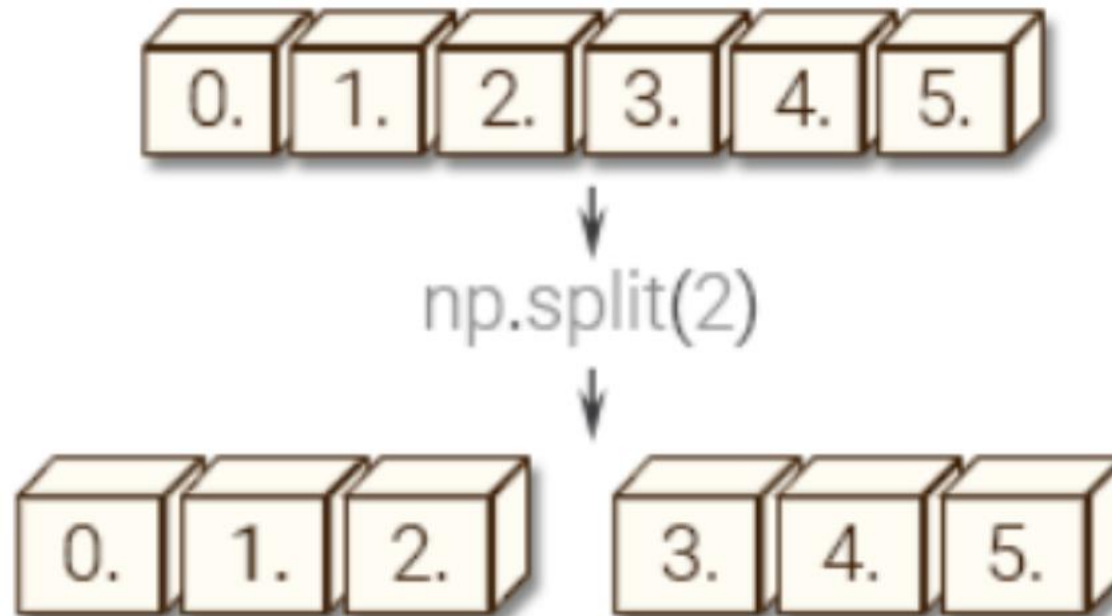




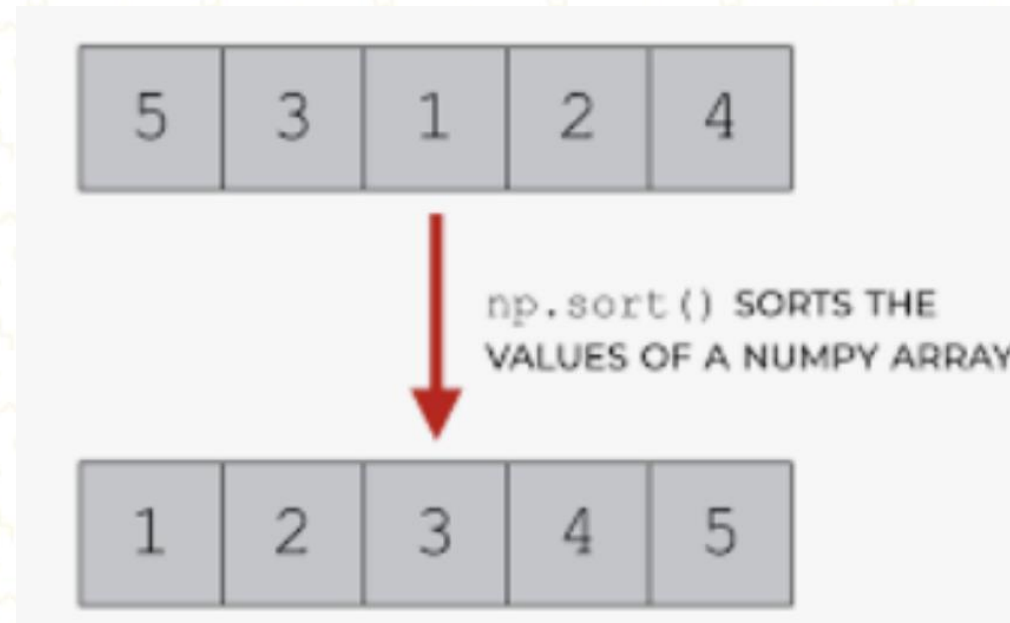
# Array Join



# Array Split

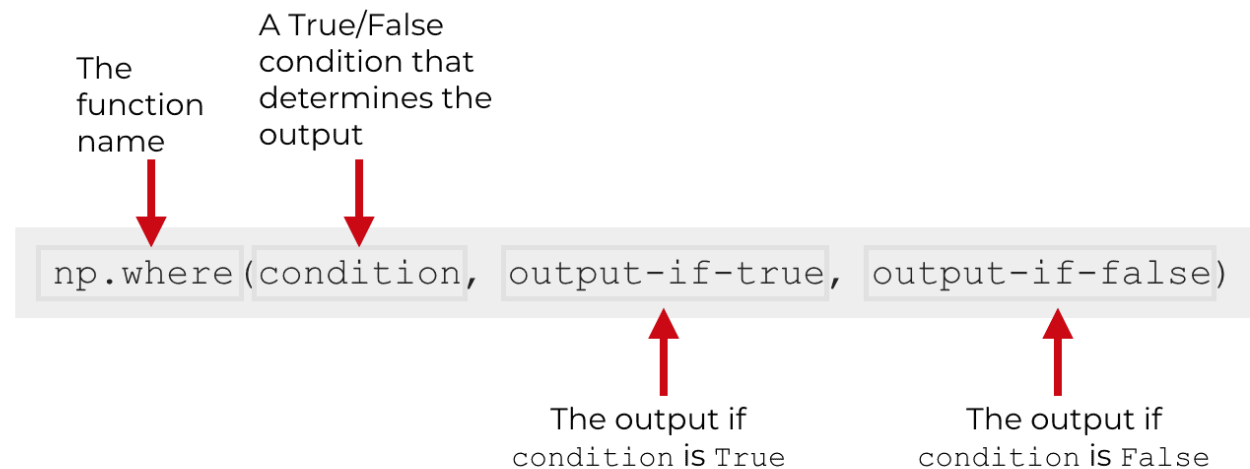


# Array Sort



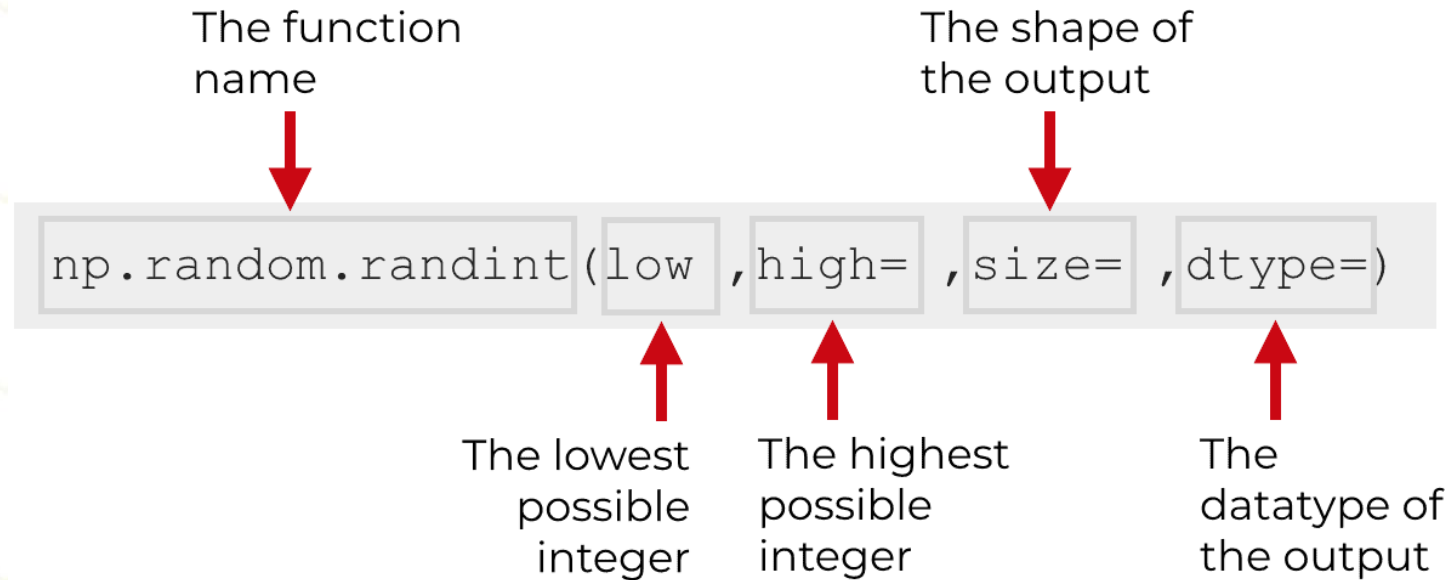


# Array Search & Filter

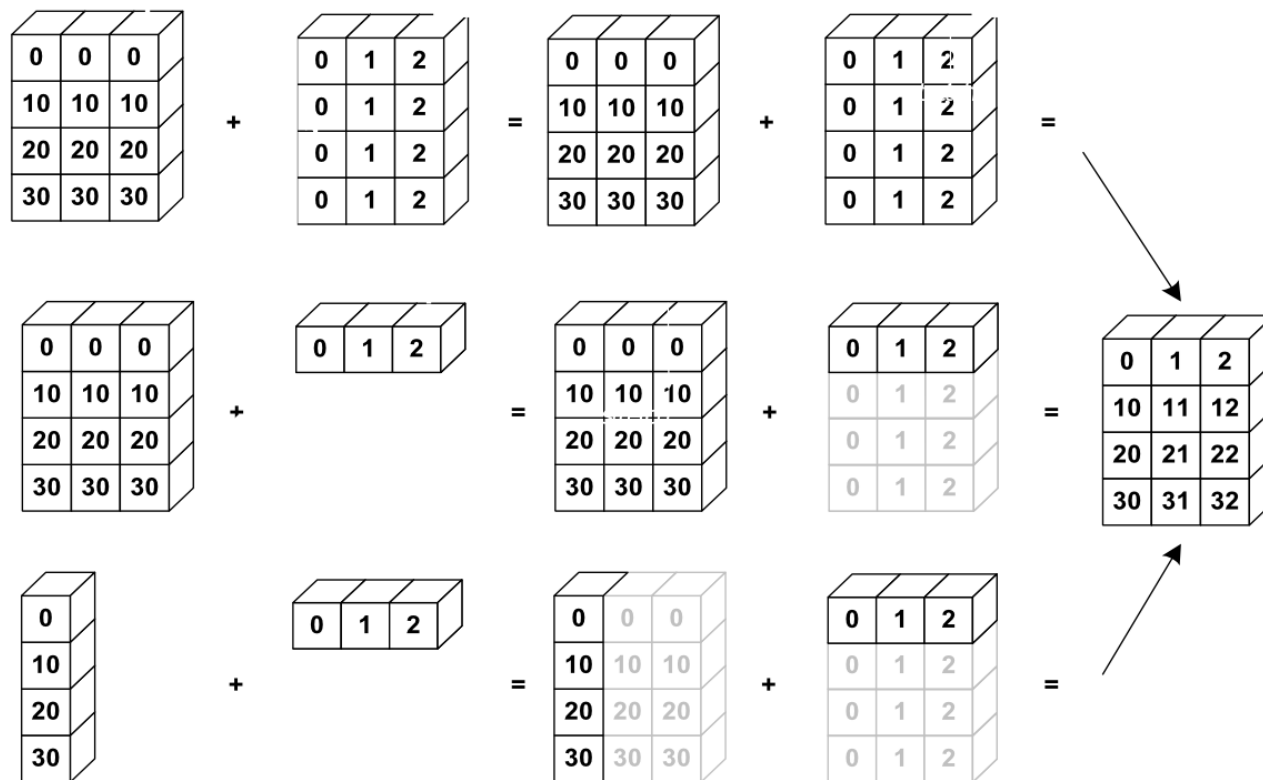




# NumPy Random



# Array Arithmetic



# Matrix Operations

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 10 & 11 \\ 20 & 21 \\ 30 & 31 \end{bmatrix}$$

$$= \begin{bmatrix} 1 \times 10 + 2 \times 20 + 3 \times 30 & 1 \times 11 + 2 \times 21 + 3 \times 31 \\ 4 \times 10 + 5 \times 20 + 6 \times 30 & 4 \times 11 + 5 \times 21 + 6 \times 31 \end{bmatrix}$$

$$= \begin{bmatrix} 10+40+90 & 11+42+93 \\ 40+100+180 & 44+105+186 \end{bmatrix} = \begin{bmatrix} 140 & 146 \\ 320 & 335 \end{bmatrix}$$



**Google Colab**

**<https://colab.research.google.com/>**





# Thank You

