# **Introduction to NumPy**

****NumPy**** stands for *Numerical Python* and it's a fundamental package for scientific computing in Python. NumPy provides Python with an extensive math library capable of performing numerical computations effectively and efficiently. These lessons are intended as a basic overview of NumPy and introduce some of its most important features.

In the following lessons you will learn:

* How to import NumPy
* How to create multidimensional NumPy ndarrays using various methods
* How to access and change elements in ndarrays
* How to load and save ndarrays
* How to use slicing to select or change subsets of an ndarray
* Understand the difference between a view and a copy an of ndarray
* How to use Boolean indexing and set operations to select or change subsets of an ndarray
* How to sort ndarrays
* How to perform element-wise operations on ndarrays
* Understand how NumPy uses broadcasting to perform operations on ndarrays of different sizes.

## **Download NumPy**

NumPy is included with ****Anaconda****. If you don't already have Anaconda installed on your computer, please refer to the Anaconda section to get clear instructions on how to install Anaconda on your PC or Mac.

## **Updating NumPy to a Specific Version**

You can choose to upgrade/downgrade your NumPy to a specific version by using the command shown below in the Terminal/Anaconda prompt.

***Tip*** - Create virtual environments, which helps us to have multiple versions of Python and its packages.

*# Use either one command*

conda install numpy=X.XX

pip install --upgrade numpy==X.XX

where X.XX could be one of the **[specific version number](https://numpy.org/devdocs/release.html" \t "https://classroom.udacity.com/nanodegrees/nd027/parts/212819c2-e958-449f-bc17-3f949b04fe9f/modules/8b964443-f29c-45a7-b2e2-330f960b4432/lessons/8b1c5460-63fc-4a45-a8a5-3564f160497f/concepts/_blank)**.

## **NumPy Documentation**

NumPy is a remarkable math library and it has many functions and features. In these introductory lessons, we will only scratch the surface of what NumPy can do. If you want to learn more about NumPy, make sure you check out the NumPy Documentation:

**[NumPy Manual](https://docs.scipy.org/doc/numpy-1.13.0/contents.html" \t "https://classroom.udacity.com/nanodegrees/nd027/parts/212819c2-e958-449f-bc17-3f949b04fe9f/modules/8b964443-f29c-45a7-b2e2-330f960b4432/lessons/8b1c5460-63fc-4a45-a8a5-3564f160497f/concepts/_blank)**  
**[NumPy User Guide](https://numpy.org/devdocs/user/index.html" \t "https://classroom.udacity.com/nanodegrees/nd027/parts/212819c2-e958-449f-bc17-3f949b04fe9f/modules/8b964443-f29c-45a7-b2e2-330f960b4432/lessons/8b1c5460-63fc-4a45-a8a5-3564f160497f/concepts/_blank)**  
**[NumPy Reference](https://numpy.org/devdocs/reference/index.html" \t "https://classroom.udacity.com/nanodegrees/nd027/parts/212819c2-e958-449f-bc17-3f949b04fe9f/modules/8b964443-f29c-45a7-b2e2-330f960b4432/lessons/8b1c5460-63fc-4a45-a8a5-3564f160497f/concepts/_blank)**  
**[Scipy Lectures](http://www.scipy-lectures.org/intro/numpy/index.html" \t "https://classroom.udacity.com/nanodegrees/nd027/parts/212819c2-e958-449f-bc17-3f949b04fe9f/modules/8b964443-f29c-45a7-b2e2-330f960b4432/lessons/8b1c5460-63fc-4a45-a8a5-3564f160497f/concepts/_blank)**

## **Benefits of using NumPy**

Even though Python lists are great on their own, NumPy has a number of key features that give it great advantages over Python lists. Below are a few convincingly strong features:

One such feature is ****speed****. When performing operations on large arrays NumPy can often perform several orders of magnitude faster than Python lists. This speed comes from the nature of NumPy arrays being memory-efficient and from optimized algorithms used by NumPy for doing arithmetic, statistical, and linear algebra operations.

Another great feature of NumPy is that it has ****multidimensional array data structures**** that can represent vectors and matrices. You will learn all about vectors and matrices in the Linear Algebra section of this course later on, and as you will soon see, a lot of machine learning algorithms rely on matrix operations. For example, when training a Neural Network, you often have to carry out many matrix multiplications. NumPy is optimized for matrix operations and it allows us to do Linear Algebra operations effectively and efficiently, making it very suitable for solving machine learning problems.

Another great advantage of NumPy over Python lists is that NumPy has a large number of ****optimized built-in mathematical functions****. These functions allow you to do a variety of complex mathematical computations very fast and with very little code (avoiding the use of complicated loops) making your programs more readable and easier to understand.

These are just some of the key features that have made NumPy an essential package for scientific computing in Python. In fact, NumPy has become so popular that a lot of Python packages, such as Pandas, are built on top of NumPy.

## **Good to Read**

You can read how to use NumPy for efficient computation, from the research article titled **[The NumPy array: a structure for efficient numerical computation](https://hal.inria.fr/inria-00564007/en" \t "https://classroom.udacity.com/nanodegrees/nd027/parts/212819c2-e958-449f-bc17-3f949b04fe9f/modules/8b964443-f29c-45a7-b2e2-330f960b4432/lessons/8b1c5460-63fc-4a45-a8a5-3564f160497f/concepts/_blank)** by Walt et. al., 2011. The article is available **[here](https://hal.inria.fr/inria-00564007/document" \t "https://classroom.udacity.com/nanodegrees/nd027/parts/212819c2-e958-449f-bc17-3f949b04fe9f/modules/8b964443-f29c-45a7-b2e2-330f960b4432/lessons/8b1c5460-63fc-4a45-a8a5-3564f160497f/concepts/_blank)**.

## **Supporting Official Resource**

If you are new to NumPy, we recommend you develop the practice of referring to the official **[NumPy User Guide](https://numpy.org/devdocs/user/index.html" \t "https://classroom.udacity.com/nanodegrees/nd027/parts/212819c2-e958-449f-bc17-3f949b04fe9f/modules/8b964443-f29c-45a7-b2e2-330f960b4432/lessons/8b1c5460-63fc-4a45-a8a5-3564f160497f/concepts/_blank)**, whenever you are looking for any *numerical* utility function.