

Why Use NumPy

You may be wondering why people use NumPy - after all, Python can handle lists, as you learned in the Intro to Python lessons

Here is the code demonstrated in the video above:

```
# Why use NumPy?
```

```
import time
```

```
import numpy as np
```

```
x = np.random.random(100000000)
```

```
# Case 1
```

```
start = time.time()
```

```
sum(x) / len(x)
```

```
print(time.time() - start)
```

```
# Case 2
```

```
start = time.time()
```

```
np.mean(x)
```

```
print(time.time() - start)
```

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:

1. 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.
2. 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.

3. 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](#) by Walt et. al., 2011. The article is available [here](#).

Supporting Official Resource

If you are new to NumPy, we recommend you develop the practice of referring to the official [NumPy User Guide](#), whenever you are looking for any *numerical* utility function.