NumPy Tutorial — Lesson Notes

What is NumPy?

NumPy (Numerical Python) is the fundamental library for numerical computing in Python. It provides fast, memory-efficient multi-dimensional arrays and mathematical functions.

Core Concepts

Creating Arrays

```
import numpy as np

arr = np.array([1, 2, 3, 4, 5])
zeros = np.zeros((3, 4))          # 3x4 matrix of zeros
ones = np.ones((2, 3))            # 2x3 matrix of ones
rng = np.arange(0, 10, 2)         # [0, 2, 4, 6, 8]
lin = np.linspace(0, 1, 5)        # [0, 0.25, 0.5, 0.75, 1.0]
```

Array Properties

```
arr.shape      # Dimensions: (5,)
arr.dtype      # Data type: int64
arr.ndim       # Number of dimensions: 1
arr.size       # Total elements: 5
```

Indexing & Slicing

```
matrix = np.array([[1,2,3],[4,5,6],[7,8,9]])
matrix[0, 1]        # 2 (row 0, col 1)
matrix[:, 0]         # [1, 4, 7] (all rows, col 0)
matrix[1:, :2]       # [[4,5],[7,8]]
```

Mathematical Operations

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

```
a + b          # [5, 7, 9]  (element-wise)
a * b          # [4, 10, 18]
np.dot(a, b)   # 32 (dot product)
np.mean(a)     # 2.0
np.std(a)      # 0.816
```

Broadcasting

```
matrix = np.ones((3, 3))
matrix + 10    # Adds 10 to every element
```

Key Takeaways

1. NumPy arrays are 50x faster than Python lists for numerical operations.
2. Use vectorized operations — avoid Python `for` loops over arrays.
3. Broadcasting lets you combine arrays of different shapes.
4. NumPy is the foundation for Pandas, Scikit-learn, and TensorFlow.
