NumPy MASTERY

A COMPLETE GUIDE TO NUMERICAL COMPUTING IN PYTHON

- POWERFUL N-DIMENSIONAL ARRAYS
- PERFORMANCE AND EFFICIENCY
- REAL-WORLD EXAMPLES
- BEST PRACTICES AND PATTERNS

DANIEL K. MERCER

NumPy Mastery

A Complete Guide to Numerical Computing in Python

Steve Publications

This book is available at https://leanpub.com/numpymastery

This version was published on 2026-08-09

This is a Leanpub book. Leanpub empowers authors and publishers with the Lean Publishing process. Lean Publishing is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

Contents

A Complete Guide to Numerical Computing in Python

This book takes you from zero experience with NumPy to advanced proficiency in using it for numerical computing, data analysis and scientific programming. You will learn not only how to write correct NumPy code, but also how to write fast, memory-efficient, maintainable code that leverages the full power of arrays. Every concept is explained with complete, runnable examples and grounded in real-world usage patterns. By the end, you will understand why NumPy works the way it does and be equipped to use it confidently as both a learning resource and long-term reference.

Introduction

In 2016, researchers announced the first image of the supermassive black hole at the center of galaxy M87. The data that made this possible had been processed using Python tools built on top of NumPy. In 2015, when scientists detected gravitational waves for the first time, NumPy was again part of the computational stack that turned raw detector signals into a discovery [1]. These are not niche applications. NumPy quietly underpins much of modern scientific computing, machine learning, finance, engineering simulation and data analysis worldwide.

This book is about understanding NumPy deeply enough to use it as fluently as Python itself.

If you know basic Python, you already know how to work with lists:

```
numbers = [1, 2, 3, 4, 5]
doubled = []
for n in numbers:
    doubled.append(n * 2)
print(doubled)  # [2, 4, 6, 8, 10]
```

This works fine for a handful of values. Now imagine doing this with millions of sensor readings, or multiplying two matrices each containing hundreds of thousands of elements, or computing statistics across an array representing image pixels. Python lists become painfully slow and memory-inefficient at scale. That is exactly the problem NumPy solves.

At its core, NumPy provides a single powerful data structure: the N-dimensional array, or ndarray. This object stores numeric data in a contiguous block of memory and exposes fast, vectorized operations that run at C speed without requiring you to write any C code. The result is code that is often 50 to 100 times faster than equivalent Python list operations, uses far less memory, and reads more clearly [2].

This book assumes you know basic Python: variables, loops, functions, lists and dictionaries. You do not need prior experience with NumPy or advanced mathematics. Every concept is introduced from first principles, with

runnable examples you can execute immediately. The progression is designed so that each chapter builds on the previous ones, taking you from your first array to advanced topics like memory mapping, strides, subclassing and type annotations.

Here is what you will learn:

- How NumPy arrays are structured in memory and why this design enables speed
- Every way to create, index, reshape and combine arrays
- Broadcasting, vectorization and universal functions that make numerical code both fast and readable
- Mathematical, statistical and linear algebra operations built into NumPy
- Random number generation with the modern Generator API
- Handling missing data, special values like NaN and infinity, and masked arrays
- Performance optimization techniques including in-place operations, memory layout control and profiling
- Advanced topics: structured arrays, memory-mapped files for huge datasets, array subclassing, and type annotations
- How NumPy integrates with pandas, SciPy, Matplotlib and scikit-learn

The book does not contain exercises or quizzes. Instead, every concept is taught through fully worked examples with realistic inputs, expected outputs, and detailed explanations. You will see both correct patterns and common mistakes, along with the reasoning behind each recommendation.

By the final chapter, you will be able to read any NumPy code in the wild and understand what it is doing, why it is written that way, and how to improve it. More importantly, you will be able to write your own NumPy code that is fast, correct and maintainable. Let us begin.

Chapter 1: Why NumPy - The Foundation of Scientific Python

The Problem with Python Lists for Numbers

Python lists are wonderfully flexible. They can hold any type of object, grow dynamically, and nest to arbitrary depth. This flexibility comes at a cost when you use them for numerical work.

Consider how Python stores a list of integers:

```
numbers = [1, 2, 3, 4, 5]
```

Behind the scenes, this list is not a contiguous block of integer values. It is an array of pointers, each pointing to a separate Python int object scattered somewhere in memory. Each Python int object carries overhead: a type tag, a reference count, and the actual value. On a typical 64-bit system, a single Python int object uses about 28 bytes [3]. A list of one million integers therefore consumes roughly 8 MB for the pointer array plus 28 MB for the int objects, totaling around 36 MB.

Now consider what happens when you perform arithmetic:

```
a = list(range(1_000_000))
b = list(range(1_000_000))
c = []
for i in range(len(a)):
    c.append(a[i] + b[i])
```

This loop is slow for several reasons:

- Each iteration involves Python-level indexing, which means following pointers and performing bounds checks.
- Each addition creates a new Python int object.

- Each append call may trigger list resizing.
- The Python interpreter must manage the loop itself, dispatching bytecode instructions on every iteration.

For one million elements, this can take well over a second on a modern machine. Worse, as the data grows, the time grows linearly with no relief in sight.

NumPy solves all of these problems at once.

Enter NumPy - A Brief History from Numeric to NumPy

NumPy did not appear from nowhere. Its story begins in 1995 when Jim Hugunin created Numeric, the first widely used library for numerical computing in Python [4]. Numeric introduced the concept of homogeneous N-dimensional arrays to Python and provided fast mathematical operations on them.

In the early 2000s, a competing project called Numarray emerged, offering performance improvements and new features but fragmenting the community. Researchers had to choose between Numeric and Numarray, and code written for one often did not work with the other.

In 2005, Travis Oliphant set out to unify these efforts. He merged the best features of Numeric and Numarray into a single project and released it as NumPy version 1.0 in 2006 [5]. The name "NumPy" was chosen after the preferred name "numerix" turned out to be trademarked by a company in digital signal processing [4].

Since then, NumPy has grown steadily with contributions from hundreds of developers worldwide. It is maintained openly on GitHub and released under the permissive BSD license. Major milestones include:

- NumPy 1.7 (2013): Introduction of ufunc methods like reduce and accumulate
- NumPy 1.10 (2015): The @ operator for matrix multiplication, matching Python 3.5's PEP 465
- NumPy 1.17 (2019): The modern random Generator API with default_rng()
- NumPy 2.0 (2023): A major release with breaking changes to the C and Python APIs, improved performance, and better type safety [6]

- NumPy 2.4 (late 2025) and 2.5 (mid 2026): Continued refinement, free-threaded Python support, and ecosystem alignment [7]

Today, NumPy is the foundation of the entire scientific Python stack. Libraries like pandas, SciPy, Matplotlib, scikit-learn, TensorFlow and PyTorch all depend on it directly or indirectly. Understanding NumPy is not optional if you want to work seriously with data in Python.

Installing and Verifying Your Setup

NumPy is available through standard Python packaging tools. The recommended installation method uses pip:

```
pip install numpy
```

If you use a scientific Python distribution like Anaconda or Miniconda, NumPy is typically included by default. You can install or update it with:

```
conda install numpy
```

To verify that NumPy is installed and check its version, run this in a Python interpreter or script:

```
import numpy as np
print(np.__version__)  # e.g., "2.5.1"
```

The convention of importing NumPy as np is universal across the Python data science ecosystem. You will see it in virtually every tutorial, blog post, and library documentation. Adopting this alias from the start makes your code immediately recognizable to other developers.

NumPy currently requires Python 3.10 or later for recent versions [8]. If you encounter installation issues, ensure your Python version is up to date.

Your First Array - A Taste of What's Coming

Let us see NumPy in action with a direct comparison. Here is the same element-wise addition we wrote earlier with Python lists, now using NumPy:

```
import numpy as np

a = np.arange(1_000_000)      # Create an array [0, 1, 2, ..., 999999]
b = np.arange(1_000_000)      # Another array of the same size
c = a + b                     # Element-wise addition, no loop needed

print(c[:5])                  # [0 2 4 6 8]
print(type(c))                # <class 'numpy.ndarray'>
```

Notice what happened:

- np.arange creates an array efficiently without building Python objects one by one.
- The expression a + b performs element-wise addition across all one million elements in a single operation, implemented in optimized C code.
- There is no explicit loop in our Python code. This is called vectorization, and it is the key to NumPy's speed.

Let us compare memory usage:

```
import sys

# Python list of one million integers
py_list = list(range(1_000_000))
list_size = sys.getsizeof(py_list)  # Size of the list structure itself
print(f"List pointer array: {list_size:,} bytes")

# NumPy array of one million int64 values
np_array = np.arange(1_000_000, dtype=np.int64)
print(f"NumPy array total: {np_array.nbytes:,} bytes")
```

On a typical system, this outputs something like:

```
List pointer array: 8,000,056 bytes
NumPy array total: 8,000,000 bytes
```

The NumPy array uses exactly 8 bytes per element (for int64) with no per-element overhead. The Python list alone requires the same amount for its pointer array, not counting the individual int objects it references. In practice, a NumPy array of integers uses roughly four to eight times less memory than an equivalent Python list [2].

Now let us look at speed. Using Python's timeit module:

```
import timeit

# Python list approach
list_time = timeit.timeit(
    """
a = list(range(1_000_000))
b = list(range(1_000_000))
c = [x + y for x, y in zip(a, b)]
""",
    number=5
)

# NumPy approach
numpy_time = timeit.timeit(
    """
import numpy as np
a = np.arange(1_000_000)
b = np.arange(1_000_000)
c = a + b
""",
    number=5
)

print(f"Python list: {list_time:.3f} seconds")
print(f"NumPy array: {numpy_time:.3f} seconds")
print(f"Speedup: {list_time / numpy_time:.1f}x")
```

On a modern machine, typical output looks like:

```
Python list: 0.650 seconds
NumPy array: 0.025 seconds
Speedup: 26.0x
```

The exact numbers depend on your hardware, but the pattern is consistent: NumPy is dramatically faster for numerical operations because it avoids Python-level overhead and uses optimized compiled code under the hood [9].

How This Book Is Organized

This book is structured to take you from fundamentals to mastery in a logical sequence. Here is the roadmap:

Chapter 2 introduces the ndarray, NumPy's core data structure. You will learn what an array really is under the hood, how its metadata works, and why understanding views versus copies is critical for writing correct code.

Chapter 3 covers every way to create arrays: from Python sequences, with specialized functions like zeros and ones, from files, and more. You will know exactly which creation method to use in any situation.

Chapter 4 dives deep into indexing and slicing. This is one of the most important skills in NumPy, and a source of many subtle bugs if misunderstood. You will master basic indexing, boolean masking, fancy indexing, and their interactions.

Chapter 5 covers reshaping, stacking, splitting and broadcasting. Broadcasting in particular is one of NumPy's most powerful features, enabling elegant operations between arrays of different shapes.

Chapter 6 explains vectorization and universal functions. You will learn why Python loops are slow, how ufuncs work internally, and how to write code that runs at C speed.

Chapter 7 covers mathematical and statistical computing: trigonometric functions, exponentials, aggregations, correlations, percentiles and numerical precision considerations.

Chapter 8 focuses on sorting, searching and set operations: efficient ways to organize, find and compare data within arrays.

Chapter 9 provides a complete guide to random number generation with NumPy's modern Generator API, including seeding for reproducibility and choosing appropriate distributions.

Chapter 10 covers linear algebra: matrix multiplication, solving systems of equations, eigenvalues, singular value decomposition, and numerical stability.

Chapter 11 addresses missing data and special values: NaN, infinity, masked arrays, and strategies for cleaning real-world datasets.

Chapter 12 is dedicated to performance optimization: understanding strides, memory layout, avoiding unnecessary copies, in-place operations, profiling and when to reach for tools like Numba or Cython.

Chapter 13 covers advanced topics: structured arrays, memory-mapped files for datasets larger than RAM, array subclassing, type annotations, and NumPy's API design philosophy.

Chapter 14 shows how NumPy integrates with the broader Python ecosystem: pandas, SciPy, Matplotlib, scikit-learn, and patterns for interoperability across libraries.

Each chapter is self-contained enough to serve as a reference, but they build on each other cumulatively. Read them in order for the best learning experience, or jump to specific chapters when you need to look up a topic.

Now let us move to Chapter 2 and understand NumPy's fundamental building block: the ndarray.

Chapter 2: The ndarray - Understanding NumPy's Core Data Structure

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

What Is an ndarray?

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Dimensions, Shape and Rank

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Data Types - The dtype System

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Memory Layout - Contiguous Arrays in C

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Views Versus Copies - A Critical Distinction

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Inspecting Your Arrays

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Chapter 3: Creating Arrays - From Scratch and From Data

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

From Python Lists and Tuples

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Generating Sequences with arange and linspace

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Filled Arrays - zeros, ones, empty, full

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Random Initialization Basics

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Identity Matrices and Diagonal Arrays

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

From Files and External Data

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Converting Between Types and Shapes During Creation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Chapter 4: Indexing and Slicing - Accessing Array Elements

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Basic Integer Indexing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Slicing - The start:stop:step Pattern

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Multi-dimensional Indexing and Slicing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Boolean Indexing and Masking

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Fancy Indexing with Arrays of Indices

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Combining Indexing Strategies

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Common Pitfalls and Gotchas

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Chapter 5: Reshaping, Stacking and Splitting - Rearranging Data

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Reshape and resize

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Flatten, Ravel and Squeeze

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Transpose and Axis Swapping

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Stacking Arrays - vstack, hstack, concatenate

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Splitting and Partitioning Arrays

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Broadcasting - The Key to Elegant Array Code

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

When Operations Return Views Versus Copies

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Chapter 6: Vectorization and Universal Functions - Writing Fast NumPy Code

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

The Cost of Python Loops

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

What Is Vectorization?

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Universal Functions - Element-wise Operations

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Arithmetic and Comparison Ufuncs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Logical Operations on Arrays

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Reducing Dimensionality with Aggregations

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Writing Your Own Vectorized Code

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Chapter 7: Mathematical and Statistical Computing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Basic Arithmetic and Powers

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Trigonometric and Exponential Functions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Rounding, Clipping and Modulo Operations

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Summarizing Data - Mean, Median, Variance, Standard Deviation

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Correlation and Covariance

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Percentiles and Quantiles

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Handling Numerical Precision

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Chapter 8: Sorting, Searching and Set Operations

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Sorting Arrays - sort and argsort

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Partial Sorting with Partition

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Searching for Values - where, argwhere, searchsorted

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Finding Unique Elements

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Set Operations - union, intersection, difference

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Comparing Arrays for Equality and Approximation

Chapter 9: Random Number Generation - The np.random Module

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Why Randomness Matters in Scientific Computing

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

The Modern Generator API - default_rng

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Seeding for Reproducibility

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Common Distributions - Uniform, Normal, Poisson

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Discrete and Specialized Distributions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Generating Random Samples Without Replacement

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Best Practices and Common Mistakes

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Chapter 10: Linear Algebra with NumPy

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Matrix Multiplication - dot, matmul and the @ Operator

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Transpose, Conjugate and Special Matrices

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Solving Linear Systems

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Determinants, Trace and Rank

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Eigendecomposition

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Singular Value Decomposition (SVD)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Condition Numbers and Numerical Stability

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Chapter 11: Handling Missing Data and Special Values

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

What Are NaN and Infinity?

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Detecting Special Values

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

nan-aware Functions - nansum, nanmean, etc.

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Masked Arrays for Complex Missing Data Patterns

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Replacing and Cleaning Special Values

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Design Decisions Behind NumPy's Approach

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Chapter 12: Performance Optimization and Memory Efficiency

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Understanding Strides and Memory Layout

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

C-contiguous Versus Fortran-contiguous Arrays

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Avoiding Unnecessary Copies

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

In-place Operations

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Vectorization Strategies for Complex Problems

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Profiling NumPy Code

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

When to Use Numba or Cython Alongside NumPy

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Chapter 13: Advanced Topics and Power User Patterns

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Structured Arrays and Record Arrays

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Memory-mapped Files for Huge Datasets

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Advanced Indexing Patterns

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Subclassing ndarray

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Type Annotations with NumPy Arrays

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

The NumPy API Design Philosophy

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Chapter 14: NumPy in the Ecosystem - Working with Other Libraries

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

NumPy and pandas - Arrays Underneath DataFrames

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

NumPy and SciPy - From Foundations to Specialized Functions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

NumPy and Matplotlib - Plotting Arrays Directly

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

NumPy and scikit-learn - The Array Interface

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Interoperability Patterns Across Libraries

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Choosing the Right Tool for the Job

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

Conclusion

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.

References

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/numpymastery.