

Advanced Python Programming Course

Lecture 9.

Intro to NumPy

Assoc. Prof. Kovalenko S.M.
Department of software engineering and
intelligent management technologies,
NTU “KhPI”

SciPy ecosystem



The SciPy ecosystem refers to a collection of open-source software packages, libraries, and tools built on top of the Python programming language for scientific computing and data analysis. It includes various packages, such as

- NumPy,
- Pandas,
- Matplotlib,
- SymPy,
- Scikit-learn
- SciPy.

NumPy



- NumPy (Numerical Python) is a Python library used for working with arrays.
- It is a fundamental package for scientific computing with Python and provides tools for working with multidimensional arrays and matrices.
- NumPy provides mathematical functions to perform operations on these arrays such as linear algebra, Fourier transform, and random number generation
- <https://numpy.org/>

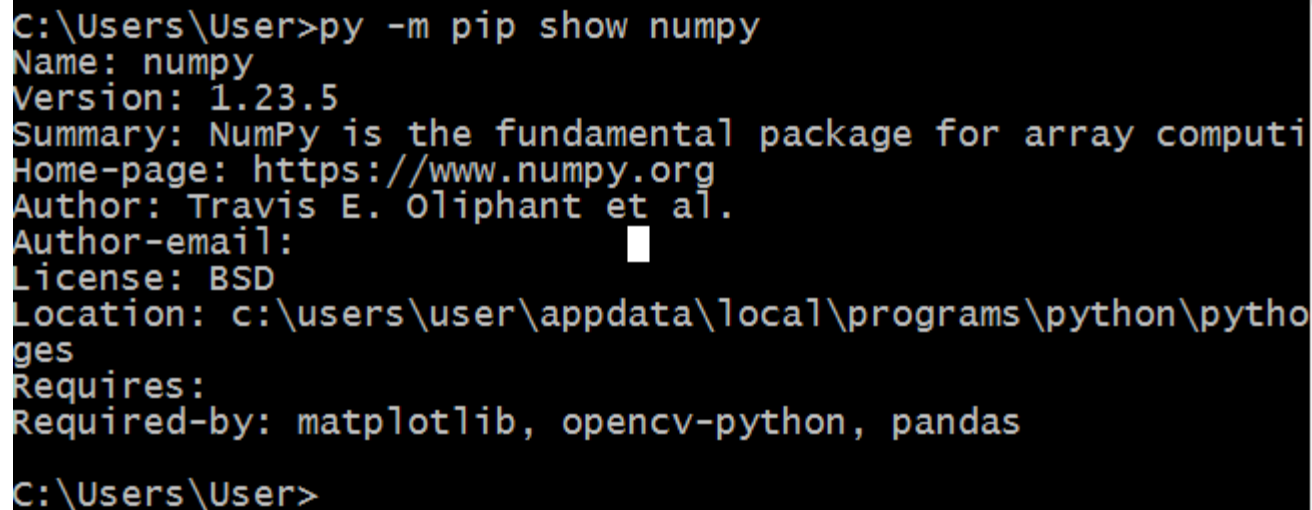
NumPy



- It also provides interoperability with other libraries that use NumPy arrays (ndarrays) as their primary data structure, such as :
 - **SciPy**: A collection of algorithms for scientific computing, built on top of NumPy. It includes modules for optimization, signal processing, interpolation, and other scientific and engineering tasks.
 - **Matplotlib**: A plotting library that works seamlessly with NumPy arrays to create high-quality visualizations.
 - **Pandas**: A library for data manipulation and analysis that uses NumPy arrays as the underlying data structure.
 - **Scikit-learn**: A machine learning library that relies heavily on NumPy arrays for data representation and manipulation.
 - **openCV-python** uses ndarray as a data structure to represent images and other multidimensional arrays

Check if numpy is installed

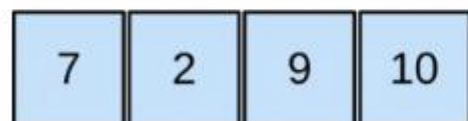
```
py -m pip show numpy
```

A screenshot of a Windows command prompt window with a black background and white text. The prompt shows the command 'py -m pip show numpy' being executed. The output displays various details about the NumPy package, including its name, version (1.23.5), a summary, home page, author, license (BSD), location, and required dependencies (matplotlib, opencv-python, pandas).

```
C:\Users\User>py -m pip show numpy
Name: numpy
Version: 1.23.5
Summary: NumPy is the fundamental package for array computi
Home-page: https://www.numpy.org
Author: Travis E. Oliphant et al.
Author-email: 
License: BSD
Location: c:\users\user\appdata\local\programs\python\pytho
ges
Requires:
Required-by: matplotlib, opencv-python, pandas
C:\Users\User>
```

Ndarray (n-dimensional array)

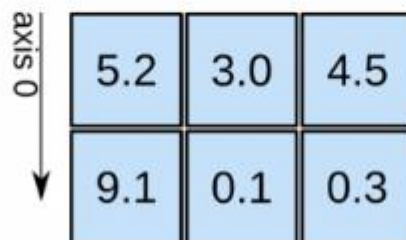
1D array



axis 0 →

shape: (4,)

2D array

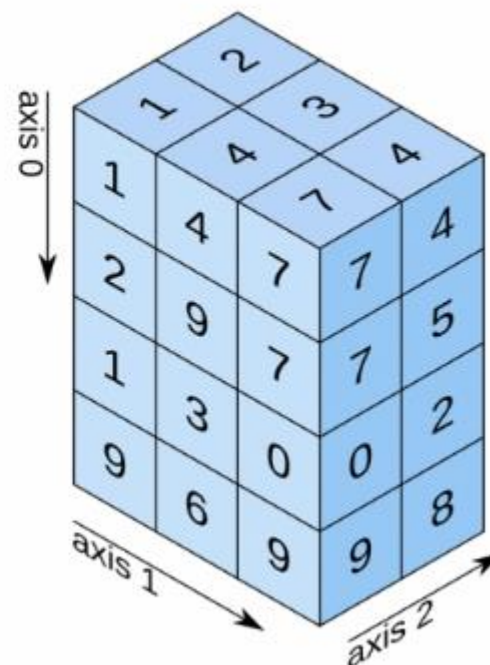


axis 0 ↓

axis 1 →

shape: (2, 3)

3D array



axis 0 ↓

axis 1 ↘

axis 2 ↗

shape: (4, 3, 2)

NumPy Standard Data Types

Data type	Description
<code>bool_</code>	Boolean (True or False) stored as a byte
<code>int_</code>	Default integer type (same as C long; normally either int64 or int32)
<code>intc</code>	Identical to C int (normally int32 or int64)
<code>intp</code>	Integer used for indexing (same as C ssize_t; normally either int32 or int64)
<code>int8</code>	Byte (-128 to 127)
<code>int16</code>	Integer (-32768 to 32767)
<code>int32</code>	Integer (-2147483648 to 2147483647)
<code>int64</code>	Integer (-9223372036854775808 to 9223372036854775807)
<code>uint8</code>	Unsigned integer (0 to 255)

NumPy Standard Data Types

<code>uint16</code>	Unsigned integer (0 to 65535)
<code>uint32</code>	Unsigned integer (0 to 4294967295)
<code>uint64</code>	Unsigned integer (0 to 18446744073709551615)
<code>float_</code>	Shorthand for <code>float64</code> .
<code>float16</code>	Half precision float: sign bit, 5 bits exponent, 10 bits mantissa
<code>float32</code>	Single precision float: sign bit, 8 bits exponent, 23 bits mantissa
<code>float64</code>	Double precision float: sign bit, 11 bits exponent, 52 bits mantissa
<code>complex_</code>	Shorthand for <code>complex128</code> .
<code>complex64</code>	Complex number, represented by two 32-bit floats
<code>complex128</code>	Complex number, represented by two 64-bit floats

Arithmetic operators implemented in NumPy

Operator	Equivalent ufunc	Description
+	np.add	Addition (e.g., $1 + 1 = 2$)
-	np.subtract	Subtraction (e.g., $3 - 2 = 1$)
-	np.negative	Unary negation (e.g., -2)
*	np.multiply	Multiplication (e.g., $2 * 3 = 6$)
/	np.divide	Division (e.g., $3 / 2 = 1.5$)
//	np.floor_divide	Floor division (e.g., $3 // 2 = 1$)
**	np.power	Exponentiation (e.g., $2 ** 3 = 8$)
%	np.mod	Modulus/remainder (e.g., $9 \% 4 = 1$)