

# Introduction to NumPy

`.npy` and `np.save / np.load`: NumPy's binary file format (`.npy`) and corresponding functions for efficiently saving and loading `ndarray` objects to and from disk

**Aggregation methods:** Array summary functions such as `sum`, `min`, `max`, `mean`, and `cumsum` that compute totals, extrema, averages, or cumulative sums across an array or along a specified axis, often supporting `keepdims` to preserve reduced dimensions

`astype`: An `ndarray` method that returns a copy of the array with its elements converted to a specified `dtype`

`axis`: An integer index identifying a specific dimension of an `ndarray` used to control the direction of operations, where axis 0 is the first dimension (rows in 2D) and axis 1 is the second (columns in 2D)

**Broadcasting:** A set of rules that allow NumPy to perform arithmetic between arrays of different shapes by virtually stretching the smaller array across the larger one when trailing dimensions are compatible (equal or one)

`concatenate`: A NumPy function that joins a sequence of arrays along an existing axis, requiring identical shapes on all other axes and the same number of dimensions

`dtype`: The data type of an `ndarray`'s elements, including information about kind (e.g., `integer`, `float`, `Unicode`) and bit-size (e.g., `int32`, `int64`)

`flip`: A NumPy function that reverses element order along one or more axes of an array, effectively mirroring the array contents along those axes

**Indexing:** Accessing individual elements or subarrays of an `ndarray` using zero-based integer positions inside square brackets

**Mask (Boolean mask):** An `ndarray` of Boolean values with the same shape as the original array that marks True for elements to select and False for elements to exclude

`ndarray` (n-dimensional array): The primary NumPy object, a homogeneous, grid-like container for elements arranged in one or more dimensions where every element shares the same `dtype`

`np.where`: A NumPy function that returns indices where a condition is true or can produce a new array by choosing elements from two alternatives based on a condition

**NumPy:** A fundamental Python library for numerical and scientific computing that provides fast, memory-efficient array objects and functions built on optimized C code

`reshape`: An `ndarray` method that returns a new view or array with the same data but a different shape, provided the total number of elements remains the same

**Scalar:** A single numeric value (e.g., 3 or 2.5) that can be used in vectorized operations and is conceptually broadcast across an array

`shape`: An `ndarray` attribute that returns a tuple of integers describing the length of the array along each dimension (e.g., (rows, columns))

**Slicing:** Extracting a contiguous subset of an `ndarray` using `start:stop:step` notation, where the start index is included and the stop index is excluded

`stack`: A NumPy function that joins arrays along a new axis, increasing dimensionality and requiring all input arrays to have the same shape and number of dimensions

`transpose`: An `ndarray` method (or `np.transpose`) that reorders the axes of an array, by default reversing the axis order or using a specified axes tuple for custom reordering

**Tuple:** An immutable Python sequence, commonly used to specify `ndarray` shapes and function arguments where the order and group of integers matter.

**Type coercion:** The implicit conversion NumPy performs when creating an array from mixed Python types, promoting all elements to a single common `dtype` (e.g., ints and floats to floats, non-numeric to strings)

**Vectorization:** The practice of expressing array operations so NumPy performs them in optimized C code over entire arrays at once, avoiding explicit Python loops for performance gains