

Essential NumPy Functions

[Array Creation] Array Creation Functions

np.array(object, dtype=None)

- Creates a NumPy array from a list, tuple, or iterable.
- * Arguments: object, dtype

np.zeros(shape, dtype=float)

- Creates an array filled with zeros.
- * Arguments: shape, dtype

np.ones(shape, dtype=float)

- Creates an array filled with ones.
- * Arguments: shape, dtype

np.eye(N, M=None, k=0, dtype=float)

- Creates an identity matrix or a 2D array with ones on a diagonal.
- * Arguments: N, M, k, dtype

np.arange(start, stop, step, dtype=None)

- Returns evenly spaced values within a given interval.
- * Arguments: start, stop, step, dtype

np.linspace(start, stop, num=50, endpoint=True, dtype=None)

- Returns `num` evenly spaced samples between start and stop.
- * Arguments: start, stop, num, endpoint, dtype

[Manipulation] Array Manipulation

np.reshape(a, newshape)

- Gives a new shape to an array without changing data.
- * Arguments: a, newshape

np.concatenate((a1, a2, ...), axis=0)

- Joins arrays along an existing axis.
- * Arguments: a1, a2, ..., axis

np.vstack(tup) / np.hstack(tup)

- Stacks arrays vertically or horizontally.
- * Arguments: tup

np.transpose(a, axes=None)

- Permutes the dimensions of an array.
- * Arguments: a, axes

np.flatten(order='C')

- Returns a copy of the array collapsed into one dimension.
- * Arguments: order

[Math] Mathematical Operations

np.add(x1, x2) / np.subtract(x1, x2)

Essential NumPy Functions

- Adds or subtracts two arrays element-wise.

* Arguments: x1, x2

np.multiply(x1, x2) / np.divide(x1, x2)

- Multiplies or divides arrays element-wise.

* Arguments: x1, x2

np.dot(a, b)

- Matrix product of two arrays.

* Arguments: a, b

np.sum(a, axis=None)

- Returns the sum of array elements.

* Arguments: a, axis

np.mean(a, axis=None) / np.median(a, axis=None)

- Calculates mean or median along a given axis.

* Arguments: a, axis

np.std(a, axis=None) / np.var(a, axis=None)

- Returns standard deviation or variance.

* Arguments: a, axis

[Logic] Logical & Comparison Functions

np.where(condition, x, y)

- Returns elements from x or y based on condition.

* Arguments: condition, x, y

np.all(a, axis=None) / np.any(a, axis=None)

- Tests if all/any elements are True.

* Arguments: a, axis

np.isclose(a, b, atol=1e-08)

- Returns boolean array where elements are close within tolerance.

* Arguments: a, b, atol

[Search/Sort] Indexing, Searching, and Sorting

np.argmax(a, axis=None) / np.argmin(a, axis=None)

- Returns indices of max/min elements.

* Arguments: a, axis

np.argsort(a, axis=-1)

- Returns indices that would sort the array.

* Arguments: a, axis

np.nonzero(a)

- Returns indices of non-zero elements.

* Arguments: a

Essential NumPy Functions

[Random] Random Number Generation

np.random.rand(d0, d1, ..., dn)

- Creates an array with samples from a uniform distribution over [0, 1).
- * Arguments: d0, d1, ..., dn

np.random.randn(d0, d1, ..., dn)

- Creates samples from the standard normal distribution.
- * Arguments: d0, d1, ..., dn

np.random.randint(low, high=None, size=None)

- Returns random integers between low and high.
- * Arguments: low, high, size

np.random.choice(a, size=None, replace=True, p=None)

- Randomly selects elements from an array.
- * Arguments: a, size, replace, p

[Linalg] Linear Algebra (np.linalg)

np.linalg.inv(a)

- Returns the inverse of a square matrix.
- * Arguments: a

np.linalg.det(a)

- Returns the determinant of a square matrix.
- * Arguments: a

np.linalg.eig(a)

- Computes eigenvalues and right eigenvectors.
- * Arguments: a