**Creation Functions:**

numpy.array(): Creates a NumPy array from a Python list or tuple.

numpy.zeros(): Creates an array filled with zeros.

numpy.ones(): Creates an array filled with ones.

numpy.empty(): Creates an array with uninitialized data.

numpy.arange(): Creates an array with evenly spaced values.

numpy.linspace(): Creates an array with evenly spaced values over a specified range.

numpy.eye(): Creates an identity matrix.

**Array Indexing and Slicing:**

numpy.array[index]: Accesses elements of an array using indexing.

numpy.array[start:stop:step]: Slices elements of an array.

**Random Number Functions:**

numpy.random.rand(): Generates random numbers from a uniform distribution.

numpy.random.randn(): Generates random numbers from a standard normal distribution.

numpy.random.randint(): Generates random integers.

numpy.random.choice(): Generates random samples from an array.

**Logical Functions:**

numpy.logical\_and(): Element-wise logical AND operation.

numpy.logical\_or(): Element-wise logical OR operation.

numpy.logical\_not(): Element-wise logical NOT operation.

**Miscellaneous Functions:**

numpy.where(): Returns indices of elements that satisfy a condition.

numpy.unique(): Finds the unique elements in an array.

numpy.isin(): Checks if elements of one array are present in another.

**Mathematical Functions:**

numpy.add(): Element-wise addition of arrays.

numpy.subtract(): Element-wise subtraction of arrays.

numpy.multiply(): Element-wise multiplication of arrays.

numpy.divide(): Element-wise division of arrays.

numpy.power(): Element-wise exponentiation of arrays.

numpy.sqrt(): Element-wise square root.

numpy.exp(): Element-wise exponential function.

numpy.log(): Element-wise natural logarithm.

numpy.sin(), numpy.cos(), numpy.tan(): Trigonometric functions.

**Statistical Functions:**

numpy.mean(): Computes the arithmetic mean along a specified axis.

numpy.median(): Computes the median along a specified axis.

numpy.sum(): Computes the sum along a specified axis.

numpy.min(): Finds the minimum value along a specified axis.

numpy.max(): Finds the maximum value along a specified axis.

numpy.var(): Computes the variance along a specified axis.

numpy.std(): Computes the standard deviation along a specified axis.

numpy.percentile(): Computes the nth percentile of an array.

**Linear Algebra Functions:**

numpy.dot(): Computes the dot product of two arrays.

numpy.matmul(): Computes the matrix product of two arrays.

numpy.linalg.inv(): Computes the inverse of a square matrix.

numpy.linalg.det(): Computes the determinant of a square matrix.

numpy.linalg.eig(): Computes the eigenvalues and eigenvectors of a square matrix.

**Array Manipulation:**

numpy.ravel(): Flattens an array into a 1-D array.

numpy.flatten(): Flattens an array into a 1-D array (similar to ravel, but returns a copy).

numpy.squeeze(): Removes single-dimensional entries from the shape of an array.

numpy.expand\_dims(): Inserts a new axis at the specified position.

numpy.stack(): Joins arrays along a new axis.

numpy.split(): Splits an array into multiple sub-arrays along a specified axis.

numpy.tile(): Constructs an array by repeating it a specified number of times.

numpy.repeat(): Repeats elements of an array.

numpy.delete(): Removes specified elements from an array.

**Aggregation Functions**:

numpy.any(): Checks if any element in an array evaluates to True.

numpy.all(): Checks if all elements in an array evaluate to True.

numpy.nonzero(): Returns the indices of non-zero elements in an array.

numpy.argmax(): Returns the indices of the maximum value along a specified axis.

numpy.argmin(): Returns the indices of the minimum value along a specified axis.

**Advanced Mathematical Functions:**

numpy.fft(): Fast Fourier Transform functions.

numpy.linalg.norm(): Computes a vector or matrix norm.

numpy.random.seed(): Seeds the random number generator for reproducibility.

numpy.clip(): Clips (limits) the values of an array to a specified range.

**Statistical Functions:**

numpy.histogram(): Computes histogram of a set of data.

numpy.cov(): Computes the covariance matrix of an array.

numpy.correlate(): Computes the correlation between two arrays.

**Custom Universal Functions (ufuncs)**:

NumPy allows users to create custom universal functions (ufuncs) for element-wise operations on arrays.

**Broadcasting:**

NumPy allows operations between arrays of different shapes and dimensions by broadcasting them to a common shape. Broadcasting is an important feature that makes it easier to perform element-wise operations on arrays with different shapes.

**Fancy Indexing:**

NumPy allows advanced indexing using boolean arrays or integer arrays (index arrays). It enables selecting or modifying array elements based on certain conditions or specific indices.

**Masked Arrays:**

NumPy supports masked arrays, where certain elements are marked as invalid or masked. This allows for more flexible handling of missing or invalid data.

**Vectorization:**

NumPy encourages vectorized operations, which allow performing computations on entire arrays rather than using loops. This improves performance and code readability.

**Matrix Operations:**

NumPy provides specialized matrix classes (numpy.matrix) and related operations, such as matrix multiplication using the @ operator.

**Handling Missing Data:**

NumPy provides functions for handling missing data in arrays, such as numpy.nan and numpy.isnan().

**Polynomial Operations:**

NumPy supports polynomial operations, including evaluation, addition, subtraction, multiplication, and integration.