Here’s a detailed list of **NumPy functions** with their purposes, starting from basic array creation to advanced operations:

**1. Basic Array Creation & Initialization**

1. **np.array()** – Create an array from lists, tuples, or other iterables.
2. **np.arange()** – Create an array with a range of values (similar to Python's range()).
3. **np.linspace()** – Create an array of evenly spaced numbers between two values.
4. **np.zeros()** – Create an array filled with zeros.
5. **np.ones()** – Create an array filled with ones.
6. **np.eye()** – Create an identity matrix (square matrix with 1s on the diagonal).
7. **np.full()** – Create an array filled with a specified value.
8. **np.empty()** – Create an uninitialized array (filled with random values).
9. **np.random.rand()** – Generate an array of random numbers from a uniform distribution between 0 and 1.
10. **np.random.randn()** – Generate random numbers from a standard normal distribution (mean 0, variance 1).
11. **np.random.randint()** – Generate random integers between specified low and high values.
12. **np.random.choice()** – Randomly select elements from an array.

**2. Array Shape & Reshaping**

1. **np.reshape()** – Reshape an array without changing its data.
2. **np.ravel()** – Flatten a multi-dimensional array into a 1D array.
3. **np.transpose()** – Transpose the dimensions of an array (swap rows and columns for 2D arrays).
4. **np.expand\_dims()** – Add an extra dimension to an array.
5. **np.squeeze()** – Remove single-dimensional entries from the shape of an array.
6. **np.concatenate()** – Concatenate (join) two or more arrays along a specified axis.
7. **np.stack()** – Stack arrays along a new axis.
8. **np.hstack()** – Stack arrays horizontally (column-wise).
9. **np.vstack()** – Stack arrays vertically (row-wise).
10. **np.split()** – Split an array into multiple sub-arrays.

**3. Mathematical Operations**

1. **np.add()** – Element-wise addition of two arrays.
2. **np.subtract()** – Element-wise subtraction of two arrays.
3. **np.multiply()** – Element-wise multiplication of two arrays.
4. **np.divide()** – Element-wise division of two arrays.
5. **np.dot()** – Dot product of two arrays.
6. **np.power()** – Element-wise exponentiation.
7. **np.sqrt()** – Element-wise square root.
8. **np.sin(), np.cos(), np.tan()** – Trigonometric functions applied element-wise.
9. **np.exp()** – Element-wise exponential (e^x).
10. **np.log()** – Element-wise natural logarithm.
11. **np.sum()** – Sum the elements along a specified axis.
12. **np.prod()** – Compute the product of elements along a specified axis.
13. **np.cumsum()** – Cumulative sum of array elements.
14. **np.cumprod()** – Cumulative product of array elements.
15. **np.mean()** – Compute the mean along a specified axis.
16. **np.median()** – Compute the median along a specified axis.
17. **np.std()** – Compute the standard deviation.
18. **np.var()** – Compute the variance.
19. **np.min()** – Find the minimum value in an array.
20. **np.max()** – Find the maximum value in an array.
21. **np.argmin()** – Find the index of the minimum value.
22. **np.argmax()** – Find the index of the maximum value.

**4. Logical & Comparison Operations**

1. **np.where()** – Return elements chosen based on a condition.
2. **np.all()** – Check if all elements in an array evaluate to True.
3. **np.any()** – Check if any element in an array evaluates to True.
4. **np.isfinite()** – Return True for elements that are finite.
5. **np.isnan()** – Return True for elements that are NaN (Not a Number).
6. **np.equal()** – Element-wise comparison for equality.
7. **np.greater()** – Element-wise comparison for greater than.
8. **np.less()** – Element-wise comparison for less than.

**5. Advanced Array Manipulation**

1. **np.tile()** – Repeat an array a specified number of times.
2. **np.repeat()** – Repeat elements of an array.
3. **np.unique()** – Find unique elements in an array.
4. **np.sort()** – Sort an array along a specified axis.
5. **np.argsort()** – Return the indices that would sort an array.
6. **np.meshgrid()** – Create a rectangular grid from two 1D arrays.
7. **np.flip()** – Reverse the order of elements along a specified axis.
8. **np.roll()** – Roll the elements of an array along a specified axis.

**6. Linear Algebra**

1. **np.linalg.inv()** – Compute the inverse of a matrix.
2. **np.linalg.det()** – Compute the determinant of a matrix.
3. **np.linalg.eig()** – Compute the eigenvalues and eigenvectors of a matrix.
4. **np.linalg.svd()** – Perform singular value decomposition (SVD) of a matrix.
5. **np.linalg.qr()** – Compute the QR decomposition of a matrix.
6. **np.linalg.norm()** – Compute the matrix or vector norm.
7. **np.linalg.solve()** – Solve a system of linear equations.

**7. Random Functions**

1. **np.random.seed()** – Seed the random number generator.
2. **np.random.permutation()** – Return a random permutation of a sequence.
3. **np.random.shuffle()** – Shuffle the elements of an array in-place.

**8. Statistical Functions**

1. **np.percentile()** – Compute the percentile of the data along a specified axis.
2. **np.histogram()** – Compute the histogram of a dataset.
3. **np.bincount()** – Count the occurrences of each value in an array of non-negative integers.
4. **np.corrcoef()** – Compute the correlation coefficient matrix.