

Important topics in arrays (NumPy)

◆ 1. Array Creation

- `np.array()`
 - `np.zeros()`, `np.ones()`, `np.full()`
 - `np.arange()`, `np.linspace()`
 - `np.eye()` (identity matrix)
 - `np.random.rand()`, `np.random.randint()`
-

◆ 2. Array Attributes

- `.shape` – dimensions of the array
 - `.ndim` – number of dimensions
 - `.size` – total number of elements
 - `.dtype` – data type of elements
-

◆ 3. Slicing and Indexing

- 1D, 2D, and 3D slicing
 - Fancy indexing: `arr[[0, 2]]`
 - Boolean indexing: `arr[arr > 5]`
 - Reversing arrays: `arr[::-1]`
 - Accessing rows, columns, or sub-matrices
-

◆ 4. Array Operations

- Arithmetic: `+`, `-`, `*`, `/`
 - Element-wise operations
 - Broadcasting rules
 - Comparison: `==`, `!=`, `>`, `<`
-

◆ 5. Reshaping and Manipulation

- `reshape()`, `ravel()`, `flatten()`
 - `transpose()`, `T`
 - `expand_dims()`, `squeeze()`
 - `concatenate()`, `stack()`, `split()`
-

◆ 6. Mathematical Functions

- `np.sum()`, `np.mean()`, `np.median()`
 - `np.min()`, `np.max()`, `np.std()`
 - `np.argmax()`, `np.argmin()`
 - `np.cumsum()`, `np.diff()`
-

◆ 7. Linear Algebra

- `np.dot()`, `np.matmul()`, `@`
 - `np.linalg.inv()` – inverse
 - `np.linalg.det()` – determinant
 - `np.linalg.eig()` – eigenvalues/vectors
 - Matrix multiplication vs. element-wise
-

◆ 8. Random Numbers

- `np.random.rand()`, `randn()`, `randint()`
 - `np.random.seed()` – for reproducibility
-

◆ 9. Handling Missing or Special Values

- `np.isnan()`, `np.isinf()`
 - Replace NaN: `np.nan_to_num()`
 - Filter or mask arrays
-

◆ 10. Performance Tips

- Vectorization instead of loops
- Memory-efficient slicing
- Avoid Python loops for speed