NumPy Cheat Sheet

Creating Arrays

```
import numpy as np
1D Array:
np.array([1, 2, 3])
2D Array:
np.array([[1,2,3], [4,5,6]])
Ones:
np.ones((10, 2))
Zeros:
np.zeros((7, 2, 3))
Range:
np.arange(0, 100, 3) # start, stop, step
```

Random Arrays

```
Random integers (0-10):
np.random.randint(10, size=(7, 2))
Random floats (0-1):
np.random.random((3, 5))
Normal distribution:
np.random.randn(3, 5)
Set seed (reproducibility):
np.random.seed(42)
Evenly spaced:
np.linspace(1, 100, 10) # 10 values
```

Array Attributes

```
arr.shape # Dimensions (rows, cols)
arr.ndim # Number of dimensions
arr.dtype # Data type
arr.size # Total elements
type(arr) # numpy.ndarray
np.unique(arr) # Unique values
```

Indexing Slicing

```
arr[0] # First row
arr[:2] # First 2 rows
arr[:2, :2] # First 2 rows cols
arr[arr > 7] # Boolean indexing
```
Note: Index starts at 0

Arithmetic Operations

```
arr1 + arr2 # Addition
arr1 - arr2 # Subtraction
arr1 * arr2 # Multiplication
arr1 / arr2 # Division
arr ** 2 # Power
np.square(arr) # Square
np.dot(a, b) # Dot product
```
⚠ **Arrays must have compatible shapes!**

Aggregation Functions

```
np.mean(arr) # Average
np.max(arr) # Maximum value
np.min(arr) # Minimum value
np.std(arr) # Standard deviation
np.var(arr) # Variance
np.sum(arr) # Sum of elements
np.argmax(arr) # Index of max
np.argmin(arr) # Index of min
```

Array Manipulation

```
arr.reshape(3,5,1) # Change shape
arr.T # Transpose
np.sort(arr) # Sort values
np.argsort(arr) # Sort indices
```
Transpose flips rows and columns
(3,2) → (2,3) enables matrix operations

Comparison Operations

```
arr1 > arr2 # Element-wise greater
arr1 >= arr2 # Greater or equal
arr1 arr2 # Less than
arr1 == arr2 # Equal to
arr > 7 # Compare to scalar
```
Returns boolean array of True/False

Axis Parameter

```
np.argmax(arr, axis=0) # Max index per col
np.argmax(arr, axis=1) # Max index per row
```
axis=0 → down columns (vertical)
axis=1 → across rows (horizontal)
Works with: sum, mean, max, min, argmax, argmin

Important Notes

• Broadcasting: NumPy auto-expands arrays
for compatible operations
• Dot product needs aligned dimensions:
(m,n) × (n,p) = (m,p)
• Use .T (transpose) to fix shape errors