NumPy Cheat Sheet

Importing NumPy

```
import numpy as np
```

Creating Arrays

```
np.array([1, 2, 3])        # Create array from list
np.zeros((2,3))            # 2x3 array of zeros
np.ones((2,3))             # 2x3 array of ones
np.full((2,2), 7)          # 2x2 array filled with 7
np.eye(3)                  # 3x3 identity matrix
np.arange(0, 10, 2)        # [0 2 4 6 8]
np.linspace(0, 1, 5)       # 5 evenly spaced values
np.random.rand(2,2)        # Random floats in [0,1)
np.random.randn(2,2)       # Standard normal distribution
np.random.randint(1, 10, size=(2,2)) # Random integers
```

Array Inspection

```
a.shape       # Shape of array
a.ndim        # Number of dimensions
a.size        # Total number of elements
a.dtype       # Data type
a.itemsize    # Size of each element in bytes
```

Indexing & Slicing

```
a[0, 1]       # Access element
a[:, 1]       # All rows, column 1
a[1:3, :]     # Rows 1-2, all columns
a[a > 5]      # Boolean indexing
```

Array Operations

```
a + b, a - b              # Element-wise ops
a * b                     # Element-wise multiplication
a @ b or np.dot(a, b)     # Matrix multiplication
a / b, a ** 2             # Division, exponentiation
np.add(a, b)              # Add arrays
np.multiply(a, b)         # Element-wise multiply
```

Aggregations

```
np.sum(a), np.mean(a), np.median(a)
```

NumPy Cheat Sheet

```
np.std(a), np.var(a)
np.min(a), np.max(a)
np.argmin(a), np.argmax(a)
np.cumsum(a), np.cumprod(a)
```

Reshaping & Combining

```
a.reshape(3, 2)              # Reshape to 3x2
a.ravel()                    # Flatten array
a.T                          # Transpose
np.vstack([a, b])            # Vertical stack
np.hstack([a, b])            # Horizontal stack
np.concatenate([a, b], axis=0)# Concatenate
```

Useful Utilities

```
np.unique(a)         # Unique values
np.sort(a)           # Sort array
np.where(a > 0)      # Indices of condition
np.isnan(a)          # Detect NaNs
np.isinf(a)          # Detect infinity
np.clip(a, min, max) # Limit values
```

Linear Algebra (np.linalg)

```
np.linalg.inv(a)     # Inverse
np.linalg.det(a)     # Determinant
np.linalg.eig(a)     # Eigenvalues & vectors
np.linalg.solve(A,b) # Solve Ax = b
```

Random Module (np.random)

```
np.random.seed(0)          # Seed RNG
np.random.rand(3)          # Random floats
np.random.randint(0,10,5)  # Random ints
np.random.choice(a, size=3)# Random sample
```