Creater :- Der Ravi
Linkedin:- www.linkedin.com/in/ravi-der
Email: derravi00@gmail.com

NumPy Cheat Sheet — With Examples

Introduction

NumPy = Numerical Python, a library for fast numerical operations with multi-dimensional arrays.
Advantages over lists:

- Faster & memory-efficient
- Supports 1D, 2D, 3D+ arrays
- Many built-in math functions
- Used in AI/ML, Data Science

Creating Arrays

From List → Array:

python

```
import numpy as np
arr = np.array([1, 2, 3, 4])
print(arr)
```

1D, 2D Arrays:

Creater :- Der Ravi
Linkedin:- www.linkedin.com/in/ravi-der
Email: derravi00@gmail.com

python
arr1 = np.array([1, 2, 3])
arr2 = np.array([[1, 2, 3], [4, 5, 6]])
print(arr2)

Special Arrays:
python

print(np.zeros((2, 3)))
print(np.ones((2, 3)))
print(np.full((2, 3), 7))
print(np.arange(1, 10, 2))
print(np.eye(3))

Array Properties

python
arr = np.array([[1, 2, 3], [4, 5, 6]])
print(arr.shape)
print(arr.size)
print(arr.ndim)
print(arr.dtype)

Type Conversion

```python
arr = np.array([1.5, 2.8, 3.3])
arr_int = arr.astype(int)
print(arr_int)
```

Creater :- Der Ravi
Linkedin:- www.linkedin.com/in/ravi-der
Email: derravi00@gmail.com

Math Operations

Element-wise:

```python
arr = np.array([10, 20, 30])
print(arr + 5)
print(arr * 2)
print(arr ** 2)
```

Aggregations:

```python
arr = np.array([10, 20, 30, 40])
print(np.sum(arr))
print(np.mean(arr))
print(np.min(arr))
print(np.max(arr))
print(np.std(arr))
print(np.var(arr))
```

Indexing & Slicing

```python
arr = np.array([10, 20, 30, 40, 50])
print(arr[0])
print(arr[1:4])
print(arr[::-1])
print(arr[[0, 3]])
print(arr[arr > 25])
```

Reshaping & Flattening

python
arr = np.array([1, 2, 3, 4, 5, 6])
print(arr.reshape(2, 3))

Creater :- Der Ravi
Linkedin:- www.linkedin.com/in/ravi-der
Email: derravi00@gmail.com

mat = np.array([[1, 2, 3], [4, 5, 6]])
print(mat.ravel())
print(mat.flatten()) -> Convert multidimensional array into the one dimensional array.

Adding, Removing, Stacking

python
arr = np.array([1, 2, 3])
print(np.insert(arr, 1, 99))
print(np.append(arr, [4, 5]))
a = np.array([1, 2])
b = np.array([3, 4])
print(np.concatenate((a, b)))
print(np.delete(arr, 0))
a = np.array([[1, 2]])
b = np.array([[3, 4]])
print(np.vstack((a, b)))
print(np.hstack((a, b)))

Splitting Arrays

Python
arr = np.array([[1, 2, 3], [4, 5, 6]])
print(np.hsplit(arr, 3))
print(np.vsplit(arr, 2))

Broadcasting

python
arr = np.array([100, 200, 300])
discount = 10

Creater :- Der Ravi
Linkedin:- www.linkedin.com/in/ravi-der
Email: derravi00@gmail.com

```
print(arr - (arr * discount / 100))
```

Handling Missing & Infinite Values

```
python
arr = np.array([1, np.nan, 3])
print(np.isnan(arr))
print(np.nan_to_num(arr, nan=0))
arr_inf = np.array([1, np.inf, -np.inf])
print(np.isinf(arr_inf))
print(np.nan_to_num(arr_inf, posinf=999, neginf=-999))
```

Quick Summary Table

Function	Purpose	Example
np.array()	Create array	np.array([1,2,3])
np.zeros()	Array of 0's	np.zeros((2,3))
np.ones()	Array of 1's	np.ones((2,3))
np.full()	Fill with value	np.full((2,3),7)
np.arange()	Range array	np.arange(1,10,2)
np.eye()	Identity matrix	np.eye(3)
arr.shape	Shape of array	(2,3)
arr.size	No. of elements	6
arr.ndim	Dimensions	2
arr.dtype	Data type	int64
arr.astype()	Type cast	arr.astype(float)
np.sum()	Sum	np.sum(arr)

Creater :- Der Ravi
Linkedin:- www.linkedin.com/in/ravi-der
Email: derravi00@gmail.com

np.mean()	Mean	np.mean(arr)
np.min()	Min value	np.min(arr)
np.max()	Max value	np.max(arr)
np.std()	Std deviation	np.std(arr)
np.insert()	Insert	np.insert(arr,1,99)
np.append()	Append	np.append(arr,[4,5])
np.concatenate()	Merge arrays	np.concatenate((a,b))
np.delete()	Delete	np.delete(arr,0)
np.hstack()	Horizontal stack	np.hstack((a,b))
np.vstack()	Vertical stack	np.vstack((a,b))
np.hsplit()	Split horizontally	np.hsplit(arr,3)
np.vsplit()	Split vertically	np.vsplit(arr,2)
np.isnan()	Detect NaN	np.isnan(arr)
np.nan_to_num()	Replace NaN/Inf	np.nan_to_num(arr,nan=0)
np.isinf()	Detect Inf	np.isinf(arr)