NumPy Cheat Sheet

This cheat sheet offers a quick and practical reference for essential NumPy commands, focusing on array creation, manipulation, and analysis, using examples drawn from the **NYC Taxis Dataset**. It covers critical topics such as importing data from files, creating and reshaping arrays, and performing scalar and vector math.

You'll also find easy-to-follow instructions on inspecting array properties, combining and splitting arrays, Boolean filtering, and computing statistics like mean, variance, and standard deviation. Whether you're analyzing 1D or 2D arrays, this cheat sheet helps you leverage NumPy's capabilities for efficient data handling.

Designed to be clear and actionable, this reference ensures that you can quickly apply NumPy's powerful array operations in your data analysis workflow.

Table of Contents

Importing Data

IMPORT, LOADTXT, GENFROMTXT, SAVETXT

Creating Arrays

ARRAY, ZEROS, ONES, EYE, LINSPACE, ARANGE, FULL, RANDOM

Inspecting Properties

SIZE, SHAPE, DTYPE, ASTYPE, TOLIST, INFO

Copying, Sorting, & Reshaping COPY, VIEW, SORT, FLATTEN, T, RESHAPE, RESIZE

Adding & Removing Elements APPEND, INSERT, DELETE

Combining & Splitting CONCATENATE, SPLIT, HSPLIT

Indexing & Slicing

INDEXING, SLICING, CONDITIONAL STATEMENTS

Scalar Math

ADD, SUBTRACT, MULTIPLY, DIVIDE, POWER, ARRAY_EQUAL, SQRT, SIN, LOG, ABS, CEIL,

ADD, SUBTRACT, MULTIPLY, DIVIDE, POWER

FLOOR, ROUND

Vector Math

Statistics

MEAN, SUM, MIN, MAX, VAR, STD, **CORRCOEF**

Working with Data

CREATING NDARRAYS, CONVERTING A LIST OF LISTS, SELECTING ROWS/COLUMNS, VECTOR OPERATIONS, STATISTICS FOR 1D/2D NDARRAYS, CREATING AN NDARRAY FROM CSV FILE, WORKING WITH BOOLEAN ARRAYS, ASSIGNING NDARRAY VALUES

Importing Data

Syntax for	How to use	Explained	Syntax for	How to use	Explained
IMPORT	<pre>import numpy as np</pre>	Imports NumPy using its standard alias, np	LINSPACE	arr = np.linspace(0, 100, 6)	Array of 6 evenly divided values from 0 to 100 ([0, 20, 40, 60, 80, 100])
LOADTXT GENFROMTXT	<pre>np.loadtxt('file.txt') np.genfromtxt('file.csv', delimiter=',')</pre>	Create an array from a .txt file Create an array from a .csv	ARANGE	arr = np.arange(0, 10, 3)	Array of values from 0 to less than 10 with step 3 ([0, 3,
SAVETXT	<pre>np.genfromtxt('fite.csv', detimiter=',') np.savetxt('file.txt', arr, delimiter=' ')</pre>	file Writes an array to a .txt file	FULL	arr = np.full((2, 3), 8)	6, 9]) 2x3 array with all values set to 8
	<pre>np.savetxt('file.csv', arr, delimiter=',')</pre>	Writes an array to a .csv file	RAND	<pre>arr = np.random.rand(4, 5)</pre>	4x5 array of random floats between 0 and 1
Creating Arrays				arr = np.random.rand(6, 7) * 100	6x7 array of random floats between 0-100
		RANDINT	arr = np.random.randint(5, size=(2, 3))	2x3 array with random integers between 0 and 4	

Syntax for	How to use	Explained
ARRAY	arr = np.array([1, 2, 3])	Create a 1D array
	arr = np.array([(1, 2, 3), (4, 5, 6)])	Create a 2D array
ZEROS	arr = np.zeros(3)	1D array of length 3; all values set to 0
ONES	arr = np.ones((3, 4))	3x4 array with all values set to 1
EYE	arr = np.eye(5)	5x5 array of 0 with 1 on diagonal (identity matrix)

























Inspecting Properties

Explained Syntax for How to use Convert arr elements to arr.astype(dtype) **ASTYPE** type dtype Convert arr to a Python list arr.tolist() **TOLIST** View documentation for np.info(np.eye) INFO np.eye Returns number of elements arr.size SIZE in arr Returns dimensions of arr arr.shape SHAPE (rows, columns) Returns type of elements in arr.dtype **DTYPE** arr

Copying, Sorting, & Reshaping

Syntax for	How to use	Explained
СОРУ	np.copy(arr)	Copies arr to new memory
VIEW	arr.view(dtype)	Creates view of arr elements with type dtype
SORT	arr.sort()	Sorts arr
SORT	arr.sort(axis=0)	Sorts specific axis of arr
FLATTEN	<pre>two_d_arr.flatten()</pre>	Flattens 2D array two_d_arr to 1D
Т	arr.T	Transposes arr (rows become columns and vice versa)
RESHAPE	arr.reshape(3, 4)	Reshapes arr to 3 rows, 4 columns without changing data
RESIZE	arr.resize((5, 6))	Changes arr shape to 5x6 and fills new values with 0























Adding & Removing Elements

Explained Syntax for How to use Appends values to end **APPEND** np.append(arr, values) of arr Inserts values into arr np.insert(arr, 2, values) **INSERT** before index 2 Deletes row on index 3 of np.delete(arr, 3, axis=0) DELETE arr Removes the 5th column np.delete(arr, 4, axis=1) from arr

ODE Combining & Splitting

Syntax for	How to use	Explained
CONCATENATE	<pre>np.concatenate((arr1, arr2), axis=0)</pre>	Adds arr2 as rows to the end of arr1
	<pre>np.concatenate((arr1, arr2), axis=1)</pre>	Adds arr2 as columns to end of arr1
SPLIT	np.split(arr, 3)	Splits arr into 3 sub-arrays
HSPLIT	np.hsplit(arr, 5)	Splits arr horizontally on the index 5

Indexing & Slicing

Syntax for	How to use	Explained
INDEXING	arr[5]	Returns the element at index 5
	arr[2, 5]	Returns the 2D array element on index [2][5]
	arr[1] = 4	Assigns array element on index 1 the value 4
	arr[1, 3] = 10	Assigns array element on index [1][3] the value 10
SLICING	arr[0:3]	Returns the elements at indices 0, 1, 2
	arr[0:3, 4]	Returns the elements on rows 0, 1, 2 in column index 4
	arr[:2]	Returns the elements at indices 0, 1
	arr[:, 1]	Returns column index 1, all rows
CONDITIONAL STATEMENTS	arr < 5	Returns an array of boolean values





















Indexing & Slicing

√ Vector Math

Syntax for	How to use			
CONDITIONAL STATEMENTS	(arr1 < 3) &			
	~arr			
	arr[arr < 5]			

Syntax for

SUBTRACT

MULTIPLY

DIVIDE

POWER

ADD

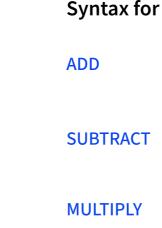


Inverts a boolean array

Returns array elements less than 5

To be **True**, both must be

To be **True**, at least one must be True





DIVIDE

POWER

SQRT

SIN

LOG

ABS

CEIL

ARRAY_EQUAL

np.divide(arr1, arr2)

How to use

np.power(arr1, arr2)



How to use

np.add(arr, 1)

np.subtract(arr, 2)

np.multiply(arr, 3)

np.divide(arr, 4)

np.power(arr, 5)

(arr1 < 3) | (arr2 > 5)

Explained

Explained

True

Add 1 to each array element

Subtract 2 from each array element

Multiply each array element by 3

Divide each array element by 4 (returns np.nan for division by zero)

Raise each array element to the power of 5



np.multiply(arr1, arr2)

np.add(arr1, arr2)

np.subtract(arr1, arr2)

np.array_equal(arr1, arr2)

np.sqrt(arr)

np.sin(arr)

np.log(arr)

np.abs(arr)

np.ceil(arr)



Elementwise add arr1 to arr2

Elementwise subtract arr2 from arr1

Elementwise multiply arr1 by arr2

Elementwise divide arr1 by arr2

Elementwise, raise arr1 to the power of arr2

Returns **True** if the arrays have the same elements and shape

Square root of each element in the array

Sine of each element in the array

Natural log of each element in the array

Absolute value of each element in the array

Rounds up each element to the nearest integer























√x Vector Math

Working with Data

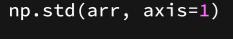
Syntax for	How to use	Explained	Syntax for	How to use	Explained
FLOOR	np.floor(arr)	Rounds down each element to the nearest integer	CREATING NDARRAYS	<pre>import numpy as np array_1d = np.array([1, 2, 3, 4, 5])</pre>	Create a 1D or 2D ndarray
ROUND	np.round(arr)	Rounds each element to the nearest integer		array_2d = np.array([[1, 2, 3], [4, 5, 6]])	
∷ Stati	istics		CONVERTING A LIST OF LISTS	<pre>import csv f = open("nyc_taxis.csv", "r") taxi_list = list(csv.reader(f)) taxi = np.array(taxi_list)</pre>	Convert a list of lists into a 2D ndarray
Syntax for	How to use	Explained			
MEAN	np.mean(arr, axis=0)	Returns mean of arr along specified axis	SELECTING ROWS	second_row = taxi[1]	Select the second row in
SUM	arr.sum()	Returns the sum of elements in arr		all_but_first_row = taxi[1:]	Select all rows from the second row onward in taxi
MIN	arr.min()	Returns minimum value of arr		<pre>fifth_row_second_column = taxi[4, 1]</pre>	Select the element from the fifth row and second column
MAX	arr.max(axis=0)	Returns maximum value of	SELECTING	cocond column - taxi[. 1]	in taxi Select all values from the











arr.corrcoef()

np.var(arr) np.std(arr, axis=1)

arr along specified axis **COLUMNS**

Returns the standard deviation of arr along specified axis

Returns the variance of arr

Returns correlation coefficient of arr

second_column = taxi[:, 1]

second_third_columns = taxi[:, 1:3] cols = [1, 3, 5] second_fourth_sixth_columns = taxi[:, cols]

twod_slice = taxi[1:4, :3]

second column in taxi

Select the second and third columns, then the second, fourth, and sixth columns in taxi

Select a slice of rows 2 to 4 and columns 1 to 3 in taxi





















Working with Data

Syntax for	How to use	Explained	Syntax for	How to use	Explained	
VECTOR OPERATIONS	vector_a + vector_b	Element-wise addition of two ndarray objects	CREATING AN NDARRAY FROM CSV FILE	<pre>import numpy as np taxi = np.genfromtxt('nyc_taxis.csv',</pre>	Load data from the nyc_taxis.csv file into an	
	vector_a - vector_b	Element-wise subtraction of two ndarray objects			ndarray, skipping the header row	
	vector_a * vector_b	Element-wise multiplication of two ndarray objects	WORKING WITH BOOLEAN	np.array([2, 4, 6, 8]) < 5	Create a Boolean array for elements less than 5	
	vector_a / vector_b	Element-wise division of two ndarray objects	ARRAYS		Use Boolean filtering to return elements less than 5	
STATISTICS FOR 1D NDARRAYS	array_1d.min()	Return the minimum value of array_1d		<pre>filter = a < 5 a[filter] # returns [2, 4]</pre>	from an ndarray	
	array_1d.max()	Return the maximum value of array_1d	ASSIGNING NDARRAY VALUES	<pre>tip_amount = taxi[:, 12] tip_bool = tip_amount > 50 top_tips = taxi[tip_bool, 5:14]</pre>	Use Boolean filtering to return rows with	
	array_1d.mean()	Calculate the average of values in array_1d			tip_amount > 50 and columns 6 to 14	
STATISTICS FOR 2D NDARRAYS	array_1d.sum()	Calculate the sum of the values in array_1d		NDARRAY	<pre>taxi[1066, 5] = 1 taxi[:, 0] = 16 taxi[550:552, 7] = taxi[:, 7].mean()</pre>	Assign values to specific elements, a column, and a slice in taxi
	array_2d.max()	Return the maximum value for the entire array_2d		taxi[taxi[:, 5] == 2, 15] = 1	Use Boolean indexing to	
	array_2d.max(axis=1) # returns a 1D ndarray	Return the maximum value in each row in array_2d			assign a value of 1 in column index 15 to rows where the 6th column equals 2	
	array_2d.max(axis=0) # returns a 1D ndarray	Return the maximum value in each column in array_2d				





























