

Numpy - Single dimensional arrays		Numpy - Single dimensional arrays (cont)		Useful Numpy Functions (cont)	
Creating an array from a list	<code>x = np.array(['a', 'b', '9', '8'])</code>	Saving to binary file	<code>np.save(open('data.npy', 'wb'), data)</code>	Basic Linear Algebra:	<code>cross, dot, outer, linalg.svd, vdot</code>
Array Data Types	Consist of integers, floating-point numbers, or strings. Data type must be consistent. Numpy's record array gives mixed DTypes	Useful Numpy Functions		Combining Arrays	
Find the length on an array	<code>np.len(x)</code>	Array Creation:	<code>arange, array, copy, empty, empty_like, eye, fromfile, fromfunction, identity, linspace, logspace, mgrid, ogrid, ones, ones_like, r_, zeros, zeros_like</code>	<code>np.vstack((a1, a2)) =</code>	<code>np.concatenate((a1, a2, axis = 0)</code>
Accessing elements from an array	<code>x[index_num] x[1]</code>	Conversions	<code>ndarray.astype, atleast_1d, atleast_2d, atleast_3d, mat</code>	<code>np.hstack((a1,a2)) =</code>	<code>np.concatenate((a1, a2, axis = 1)</code>
Assign elements from index	<code>x[1]=c</code>	Manipulations:	<code>array_split, column_stack, concatenate, diagonal, dsplit, dstack, hsplit, hstack, ndarray.item, newaxis, ravel, repeat, reshape, resize, squeeze, swapaxes, take, transpose, vsplit, vstack</code>	<code>vsplit</code>	splits along the vertical axis
Slicing <code>x[start:end]</code>	<code>print(x[1:2]) ['b', '9']</code>	Questions:	<code>all, any, nonzero, where</code>	<code>hsplit</code>	splits along the horizontal axis
Slicing <code>x[start:end:step]</code>	<code>print(x[1:3:2]) ['b', '8']</code>	Ordering:	<code>argmax, argmin, argsort, max, min, ptp, searchsorted, sort</code>	Universal Functions (ufuncs)	
Modify a new version of an array without changing the original	<code>y = x.copy()</code>	Operations:	<code>choose, compress, cumprod, cumsum, inner, ndarray.fill, imag, prod, put, putmask, real, sum</code>	Implement vectorization (operations applied to whole arrays instead of individual elements) in NumPy which is way faster than iterating over elements.	Also provide broadcasting (when smaller array is cast across the larger array so that they have compatible shapes)
Negative slices, single value <code>x[-Distance from end]</code>	<code>x[-2] b</code>	Basic Statistics:	<code>cov, mean, std, var</code>	<code>x = [1, 2, 3, 4] y = [4, 5, 6, 7] z = np.add(x, y)</code>	<code>[5 7 9 11]</code>
Negative slices, reversal <code>x[start:end: -step]</code>	<code>x[3:0:-2]) 8, b</code>	Check if a function is a ufunc:		<code>arr1 = np.array([10, 20, 30, 40, 50, 60])</code>	<code>arr2 = np.array([20, 21, 22, 23, 24, 25])</code>
Adding to an array	<code>x.append('7')</code>	<code>np.subtract(arr1, arr2)</code>		<code>newarr = np.multiply(arr1, arr2)</code>	



By **datamansam**

cheatography.com/datamansam/

Published 23rd November, 2021.

Last updated 23rd November, 2021.

Page 1 of 2.

Sponsored by **CrosswordCheats.com**

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>

Universal Functions (ufuncs) (cont)

```
newarr = np.divide-   newarr = np.pow-
(arr1, arr2)          er(arr1, arr2)
```

```
newarr = np.remain-   newarr = np.abs-
der(arr1, arr2)        olute(arr)
```

Iterating

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

```
for x in arr: print(x)      [1, 2,
                             3], [4,
                             5, 6]
```

```
for x in arr: for y in x: print(y) 123456
```

```
for x in np.nditer(arr): 123456
```

```
for x in np.nditer(arr, flags=['buff- b '1',
ered'], op_dtypes=['S']): print(x) b'2', b'3'
```

```
for x in np.nditer(arr[:, ::2]): 1, 3, 5,
print(x)                          7
```

Enumeration means mentioning sequence number of somethings one by one.

```
for idx, x in np.ndenumerate(arr): print(idx, x)
```

```
for idx, x in np.ndenumerate(- 0,) 1
arr): print(idx, x) (      (1,) 2
                        (2,) 3
```

Iterating (cont)

```
for idx, x in np.ndenumerate(arr): print(idx, x)
Result Size: 1425 x 1251 import
numpy as np arr = np.array([[1,
2, 3, 4], [5, 6, 7, 8]]) for idx, x in
np.ndenumerate(arr): print(idx,
x) (0, 0) 1 (0, 1) 2 (0, 2) 3 (0, 3)
4 (1, 0) 5 (1, 1) 6 (1, 2) 7 (1, 3) 8
```



By **datamansam**

cheatography.com/datamansam/

Published 23rd November, 2021.

Last updated 23rd November, 2021.

Page 2 of 2.

Sponsored by **CrosswordCheats.com**

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>