

Python Cheat Sheet: NumPy

“A puzzle a day to learn, code, and play” → Visit finxter.com

Name	Description	Example
<code>a.shape</code>	The shape attribute of NumPy array a keeps a tuple of integers. Each integer describes the number of elements of the axis.	<pre>a = np.array([[1,2],[1,1],[0,0]]) print(np.shape(a)) # (3, 2)</pre>
<code>a.ndim</code>	The ndim attribute is equal to the length of the shape tuple.	<pre>print(np.ndim(a)) # 2</pre>
<code>*</code>	The asterisk (star) operator performs the Hadamard product, i.e., multiplies two matrices with equal shape element-wise.	<pre>a = np.array([[2, 0], [0, 2]]) b = np.array([[1, 1], [1, 1]]) print(a*b) # [[2 0] [0 2]]</pre>
<code>np.matmul(a,b)</code> , <code>a@b</code>	The standard matrix multiplication operator. Equivalent to the <code>@</code> operator.	<pre>print(np.matmul(a,b)) # [[2 2] [2 2]]</pre>
<code>np.arange([start,]stop, [step,])</code>	Creates a new 1D numpy array with evenly spaced values	<pre>print(np.arange(0,10,2)) # [0 2 4 6 8]</pre>
<code>np.linspace(start, stop, num=50)</code>	Creates a new 1D numpy array with evenly spread elements within the given interval	<pre>print(np.linspace(0,10,3)) # [0. 5. 10.]</pre>
<code>np.average(a)</code>	Averages over all the values in the numpy array	<pre>a = np.array([[2, 0], [0, 2]]) print(np.average(a)) # 1.0</pre>
<code><slice> = <val></code>	Replace the <code><slice></code> as selected by the slicing operator with the value <code><val></code> .	<pre>a = np.array([0, 1, 0, 0, 0]) a[::2] = 2 print(a) # [2 1 2 0 2]</pre>
<code>np.var(a)</code>	Calculates the variance of a numpy array.	<pre>a = np.array([2, 6]) print(np.var(a)) # 4.0</pre>
<code>np.std(a)</code>	Calculates the standard deviation of a numpy array	<pre>print(np.std(a)) # 2.0</pre>
<code>np.diff(a)</code>	Calculates the difference between subsequent values in NumPy array a	<pre>fibs = np.array([0, 1, 1, 2, 3, 5]) print(np.diff(fibs, n=1)) # [1 0 1 1 2]</pre>
<code>np.cumsum(a)</code>	Calculates the cumulative sum of the elements in NumPy array a.	<pre>print(np.cumsum(np.arange(5))) # [0 1 3 6 10]</pre>
<code>np.sort(a)</code>	Creates a new NumPy array with the values from a (ascending).	<pre>a = np.array([10,3,7,1,0]) print(np.sort(a)) # [0 1 3 7 10]</pre>
<code>np.argsort(a)</code>	Returns the indices of a NumPy array so that the indexed values would be sorted.	<pre>a = np.array([10,3,7,1,0]) print(np.argsort(a)) # [4 3 1 2 0]</pre>
<code>np.max(a)</code>	Returns the maximal value of NumPy array a.	<pre>a = np.array([10,3,7,1,0]) print(np.max(a)) # 10</pre>
<code>np.argmax(a)</code>	Returns the index of the element with maximal value in the NumPy array a.	<pre>a = np.array([10,3,7,1,0]) print(np.argmax(a)) # 0</pre>
<code>np.nonzero(a)</code>	Returns the indices of the nonzero elements in NumPy array a.	<pre>a = np.array([10,3,7,1,0]) print(np.nonzero(a)) # [0 1 2 3]</pre>