NumPy Important Functions — Syntax · Example · Output

A compact, printable cheat-sheet showing **function**, **syntax**, **one example**, and **output**. Copy and run the examples in a Python REPL or Jupyter to verify.

> Note: outputs shown are exact literals you would see printing the result in a typical NumPy session.

---

1) Array creation

Function	Syntax	Example	Output
`array`	`np.array(object, dtype=None)`	`np.array([1,2,3])`	`array([1, 2, 3])`
`zeros`	`np.zeros(shape, dtype=float)`	`np.zeros((2,3))`	`array([[0., 0., 0` `[0., 0., 0.]])`
`ones`	`np.ones(shape)`	`np.ones((2,2))`	`array([[1., 1.],\` `1.]])`
`empty`	`np.empty(shape)`	`np.empty((2,2))`	`array([[0., 0.],\` `0.]])` *(uninitialized va*
`full`	`np.full(shape, fill_value)`	`np.full((2,3),7)`	`array([[7,7,7],\n` `[7,7,7]])`
`arange`	`np.arange(start, stop, step)`	`np.arange(0,9).reshape(3,3)`	`array([[0,1,2],\n` `[3,4,5],\n [`
`linspace`	`np.linspace(start, stop, num)`	`np.linspace(0,1,5)`	`array([0. ,0.25,` `0.75,1. ])`
`identity`	`np.identity(n)`	`np.identity(3)`	`array([[1.,0.,0.]` `[0.,0.,1.]])`
`eye`	`np.eye(N, M=None, k=0)`	`np.eye(3,4)`	`array([[1.,0.,0.,` `[0.,1.,0.,0.],[0.,`
`random.rand`	`np.random.rand(d0,d1,...)`	`np.random.rand(2,2)`	`array([[0.3745401` `\n` `[0.73199394,0.5986`
`random.randn`	`np.random.randn(d0,d1,...)`	`np.random.randn(2)`	`array([ 1.7640523` `0.40015721])` *(varies*

Function	Syntax	Example	Output
`random.randint`	`np.random.randint(low, high, size)`	`np.random.randint(0,10,5)`	`array([5,0,3,3,7]`
`copy`	`np.copy(arr)`	`a=np.array([1,2]); b=np.copy(a); b[0]=9; a`	`array([1,2])` (origi

2) Inspection

Function/ attr	Syntax	Example	Output
`shape`	`arr.shape`	`np.array([[1,2],[3,4]]).shape`	`(2, 2)`
`ndim`	`arr.ndim`	`np.array([1,2,3]).ndim`	`1`
`size`	`arr.size`	`np.array([[1,2],[3,4]]).size`	`4`
`dtype`	`arr.dtype`	`np.array([1,2.0]).dtype`	`dtype('float64')`
`itemsize`	`arr.itemsize`	`np.array([1], dtype=np.int32).itemsize`	`4`

3) Reshape & manipulate

Function	Syntax	Example	Output
`reshape`	`arr.reshape(newshape)`	`np.arange(6).reshape(2,3)`	`array([[0,1,2], [3,4,5]])`
`flatten`	`arr.flatten()`	`np.arange(4).reshape(2,2).flatten()`	`array([0,1,2,3])`
`ravel`	`arr.ravel()`	`np.arange(4).reshape(2,2).ravel()`	`array([0,1,2,3])` (view if possible)
`resize`	`np.resize(arr, new_shape)`	`np.resize(np.array([1,2]), (2,3))`	`array([[1,2,1], [2,1,2]])`
`expand_dims`	`np.expand_dims(arr, axis)`	`np.expand_dims(np.array([1,2]),0)`	`array([[1,2]])`
`squeeze`	`np.squeeze(arr)`	`np.squeeze(np.array([[[1],[2]]]))`	`array([[1],[2]])` (removes size-1 axes)
`concatenate`	`np.concatenate([a,b], axis=0)`	`np.concatenate([ [1,2],[3,4] ])`	`array([1,2,3,4])`

Function	Syntax	Example	Output
`vstack`	`np.vstack([a,b])`	`np.vstack(( [1,2],[3,4]))`	`array([[1,2], [3,4]])`
`hstack`	`np.hstack([a,b])`	`np.hstack(( [1,2],[3,4]))`	`array([1,2,3,4])`
`split`	`np.split(arr, indices_or_sections, axis=0)`	`np.split(np.arange(6),3)`	`[array([0,1]), array([2,3]), array([4,5])]`

4) Indexing & slicing

Feature	Syntax	Example	Output
Basic slice	`arr[start:stop:step]`	`np.arange(10)[2:8:2]`	`array([2,4,6])`
Column/ Row	`arr[:,i]` / `arr[i,:]`	`a=np.arange(9).reshape(3,3); a[:,1]`	`array([1,4,7])`
Boolean indexing	`arr[arr>5]`	`np.arange(10)[np.arange(10)>7]`	`array([8,9])`
`where`	`np.where(condition)`	`np.where(np.array([1,0,2])>0)`	`(array([0,2]),)`
`take`	`np.take(arr, indices)`	`np.take([10,20,30], [0,2])`	`array([10,30])`
`put`	`np.put(arr, indices, values)`	`a=np.array([1,2,3]); np.put(a, [0,2],[9,8]); a`	`array([9,2,8])`

5) Math (elementwise & reductions)

Function	Syntax	Example	Output
`add`	`np.add(a,b)`	`np.add([1,2],[3,4])`	`array([4,6])`
`subtract`	`np.subtract(a,b)`	`np.subtract([5,4], [2,1])`	`array([3,3])`
`multiply`	`np.multiply(a,b)`	`np.multiply([1,2], [3,4])`	`array([3,8])`
`divide`	`np.divide(a,b)`	`np.divide([8,6], [2,3])`	`array([4.,2.])`

Function	Syntax	Example	Output
power	np.power(a, b)	np.power(2,3)	8
round	np.round(arr, decimals=0)	np.round(3.1415,2)	3.14
floor / ceil	np.floor(x) / np.ceil(x)	np.floor(1.9)	1.0
log / exp	np.log(x) / np.exp(x)	np.exp(1)	2.718281828459045
sin	np.sin(x)	np.sin(np.pi/2)	1.0
mean	np.mean(arr, axis=None)	np.mean([1,2,3])	2.0
median	np.median(arr)	np.median([1,3,2])	2.0
std	np.std(arr)	np.std([1,2,3])	0.816496580927726
var	np.var(arr)	np.var([1,2,3])	0.6666666666666666
min / max	np.min(arr) / np.max(arr)	np.min([5,2,9])	2
sum	np.sum(arr, axis=None)	np.sum([1,2,3])	6
prod	np.prod(arr)	np.prod([2,3,4])	24
cumsum	np.cumsum(arr)	np.cumsum([1,2,3])	array([1,3,6])

6) Linear algebra (numpy.linalg)

Function	Syntax	Example	Output
dot	np.dot(a,b)	np.dot([1,2],[3,4])	11
matmul	np.matmul(a,b)	np.matmul([[1,2]], [[3],[4]])	array([[11]])
inv	np.linalg.inv(a)	np.linalg.inv(np.array([[1,2],[3,4]]))	array([[-2. , 1.], [1.5,-0.5]])
det	np.linalg.det(a)	np.linalg.det(np.array([[1,2],[3,4]]))	-2.0000000000000004
eig	np.linalg.eig(a)	np.linalg.eig(np.array([[2,0],[0,3]]))	(array([2.,3.]), array([[1.,0.],[0.,1.]]))

Function	Syntax	Example	Output
`norm`	`np.linalg.norm(a)`	`np.linalg.norm([3,4])`	`5.0`
`transpose`	`a.T` or `np.transpose(a)`	`np.array([[1,2],[3,4]]).T`	`array([[1,3], [2,4]])`
`trace`	`np.trace(a)`	`np.trace(np.array([[1,2], [3,4]]))`	`5`

7) Sorting & searching

Function	Syntax	Example	Output
`sort`	`np.sort(arr)`	`np.sort([3,1,2])`	`array([1,2,3])`
`argsort`	`np.argsort(arr)`	`np.argsort([3,1,2])`	`array([1,2,0])`
`unique`	`np.unique(arr)`	`np.unique([1,2,1,3])`	`array([1,2,3])`
`argmax` / `argmin`	`np.argmax(arr)`	`np.argmax([1,5,3])`	`1`
`nonzero`	`np.nonzero(arr)`	`np.nonzero([0,2,0,3])`	`(array([1,3]),)`
`in1d`	`np.in1d(a, b)`	`np.in1d([1,2,3], [2,3,4])`	`array([False, True, True])`

8) Random utilities

Function	Syntax	Example	Output
`seed`	`np.random.seed(seed)`	`np.random.seed(0); np.random.rand(2)`	`array([0.5488135 , 0.71518937])`
`shuffle`	`np.random.shuffle(x)`	`a=[1,2,3]; np.random.shuffle(a); a`	`[2,1,3]` *(varies)*
`choice`	`np.random.choice(a, size, replace=True, p=None)`	`np.random.choice([10,20,30], 2,replace=False)`	`array([20,10])` *(varies)*

9) Special / utility

Function	Syntax	Example	Output
`meshgrid`	`np.meshgrid(x, y)`	`X,Y = np.meshgrid([0,1], [0,1]); X`	`array([[0,1],[0,1]])`
`all`	`np.all(condition)`	`np.all([True, True])`	`True`
`any`	`np.any(condition)`	`np.any([False, True])`	`True`
`isnan`	`np.isnan(x)`	`np.isnan([1, np.nan])`	`array([False, True])`
`isinf`	`np.isinf(x)`	`np.isinf([1, np.inf])`	`array([False, True])`
`clip`	`np.clip(arr, a_min, a_max)`	`np.clip([0,5,10], 1, 6)`	`array([1,5,6])`
`tile`	`np.tile(arr, reps)`	`np.tile([1,2],2)`	`array([1,2,1,2])`
`repeat`	`np.repeat(arr, repeats)`	`np.repeat([1,2],3)`	`array([1,1,1,2,2,2])`

Quick tips

- Use `help(np.function)` or `np.function?` in IPython to see detailed docs.
- For large arrays, printing will be truncated — use `.shape` and `.dtype` to inspect.
- Many functions accept an `axis` argument to operate along rows/columns.

End of cheat-sheet. If you want this exported as a downloadable PDF, or want me to include more functions (e.g., NumPy `fft`*,* `polynomial`*, or* `random.Generator` *APIs), tell me and I will add them.*