

Untitled

18 November 2025 06:39

NUMPY COMPLETE CHEATSHEET

IMPORT

```
import numpy as np
```

ARRAY CREATION

```
np.array()  
np.asarray()  
np.copy()  
np.zeros()  
np.ones()  
np.empty()  
np.full()  
np.eye()  
np.identity()  
np.diag()  
np.arange()  
np.linspace()  
np.logspace()  
np.meshgrid()  
np.indices()  
np.frombuffer()  
np.fromfile()  
np.fromfunction()
```

RANDOM (np.random)

```
np.random.rand()  
np.random.randn()  
np.random.randint()  
np.random.random()  
np.random.uniform()
```

```
np.random.normal()  
np.random.binomial()  
np.random.choice()  
np.random.permutation()  
np.random.shuffle()  
np.random.seed()  
RNG (modern API)  
np.random.default_rng()  
rng.random()  
rng.integers()  
rng.normal()  
rng.uniform()
```

DTYPES

```
arr.dtype  
arr.astype()  
np.int32  
np.int64  
np.float32  
np.float64  
np.bool_  
np.complex64  
np.str_  
np.finfo()  
np.iinfo()
```

ARRAY INFO

```
arr.shape  
arr.ndim  
arr.size  
arr.itemsize  
arr nbytes  
arr.strides  
arr.T  
arr.real
```

arr.imag
arr.flat
arr.flags

INDEXING & SLICING

arr[i]
arr[i:j]
arr[i:j:k]
arr[:, :]
arr[:, 0]
arr[0, :]
arr[..., 1]
arr[[1,5,7]]
arr[arr > 0]
np.where()
np.take()
np.put()
np.nonzero()
np.extract()
np.choose()

RESHAPING & DIMENSIONS

arr.reshape()
arr.ravel()
arr.flatten()
arr.squeeze()
np.expand_dims()
arr[..., np.newaxis]
np.newaxis
np.resize()
arr.resize()
np.transpose()
np.moveaxis()
np.swapaxes()

MERGING & SPLITTING

np.concatenate()
np.stack()
np.vstack()
np.hstack()
np.dstack()
np.column_stack()
np.row_stack()
np.block()
np.split()
np.array_split()
np.hsplit()
np.vsplit()
np.dsplits()

ELEMENTWISE ARITHMETIC

np.add()
np.subtract()
np.multiply()
np.divide()
np.power()
np.mod()

BASIC MATH FUNCTIONS

np.abs()
np.round()
np.floor()
np.ceil()
np.trunc()
np.sign()

ADVANCED MATH

np.exp()
np.log()
np.log2()

```
np.log10()  
np.sqrt()  
np.square()  
np.sin()  
np.cos()  
np.tan()  
np.arcsin()  
np.arccos()  
np.arctan()  
np.arctan2()  
np.sinh()  
np.cosh()  
np.tanh()  
np.minimum()  
np.maximum()  
np.clip()
```

STATISTICS

```
np.mean()  
np.median()  
np.std()  
np.var()  
np.sum()  
np.prod()  
np.min()  
np.max()  
np.argmin()  
np.argmax()  
np.percentile()  
np.quantile()  
np.cumsum()  
np.cumprod()
```

LINEAR ALGEBRA (np.linalg)

```
np.linalg.dot()
```

```
np.dot()  
np.matmul()  
np.vdot()  
np.inner()  
np.outer()  
np.tensordot()  
np.linalg.norm()  
np.linalg.inv()  
np.linalg.det()  
np.linalg.eig()  
np.linalg.eigh()  
np.linalg.svd()  
np.linalg.solve()  
np.linalg.matrix_rank()
```

BROADCASTING

```
np.broadcast()  
np.broadcast_to()  
np.expand_dims()  
arr.reshape()  
(arr reshaping for compatible dims)
```

SORTING

```
np.sort()  
arr.sort()  
np.argsort()  
np.partition()  
np.argpartition()
```

SET OPERATIONS

```
np.unique()  
np.union1d()  
np.intersect1d()  
np.setdiff1d()  
np.setxor1d()
```

`np.in1d()`

`np.isin()`

LOGICAL OPS

`np.logical_and()`

`np.logical_or()`

`np.logical_not()`

`np.logical_xor()`

`np.all()`

`np.any()`

COMPARISON OPS

`arr == x`

`arr != x`

`arr > x`

`arr < x`

`arr >= x`

`arr <= x`

`np.equal()`

`np.not_equal()`

`np.greater()`

`np.less()`

`np.greater_equal()`

`np.less_equal()`

MATRIX CREATION SHORTCUTS

`np.tri()`

`np.tril()`

`np.triu()`

`np.vander()`

`np.fromiter()`

`np.tile()`

`np.repeat()`

I/O OPERATIONS

```
np.load()  
np.save()  
np.savez()  
np.savetxt()  
np.loadtxt()
```

BITWISE (rare but complete)

```
np.bitwise_and()  
np.bitwise_or()  
np.bitwise_xor()  
np.invert()  
np.left_shift()  
np.right_shift()
```

SPECIAL FUNCTIONS (SciPy-like, NumPy subset)

```
np.gcd()  
np.lcm()
```

NUMPY PERFORMANCE

```
np.vectorize()  
np.frompyfunc()  
numpy broadcasting  
ufunc.reduce()  
ufunc.accumulate()  
ufunc.outer()
```

MASKING

```
np.ma.masked_array()  
np.ma.masked_where()  
np.ma.filled()
```

ADVANCED (ML-SIDE USAGE)

```
np.stack for batch tensors  
np.expand_dims for channel dims  
np.tile for dataset augmentation
```

`np.random.normal` for weight init
`np.linalg.svd / eig` for PCA
`np.mean(axis=0)` for feature averaging
`np.where()` for label mapping
`np.clip()` for gradient clipping