All Rights Reserved. Author @ Rajendra Phani

Numpy Cheet Sheet : Author @ Rajendra Phani

Create Commands

Function / Method	Description	Example
np.array()	Create an array from a list/tuple.	a = np.array([1, 2, 3])
np.arange()	Return evenly spaced values within a given interval.	x = np.arange(0, 10, 2)
np.linspace()	Return evenly spaced numbers over a specified interval (inclusive).	y = np.linspace(0, 1, 5)
np.zeros()	Create an array of zeros.	z = np.zeros((2,3))
np.ones()	Create an array of ones.	o = np.ones((3,1))
np.empty()	Allocate space without initializing.	e = np.empty((4,4))
np.full()	Create an array filled with a scalar value.	f = np.full((2,2), 7)
np.eye()	Identity matrix.	I = np.eye(3)
np.identity()	Same as eye.	I2 = np.identity(3)
np.random.rand()	Uniform random values in [0,1).	r = np.random.rand(2,3)
np.random.randn()	Standard normal distribution.	n = np.random.randn(3)
np.random.randint()	Random integers in a range.	ints = np.random.randint(0,10,size=(3,2))
np.reshape()	Change shape without copying.	a = a.reshape(3,2)
np.ravel() / .flatten()	Convert to 1-D array (copy vs view).	b = a.ravel()
np.transpose() / .T	Swap axes.	c = a.T
np.swapaxes()	Swap two axes.	d = np.swapaxes(a, 0, 1)
np.concatenate()	Join arrays along an axis.	e = np.concatenate([a,b], axis=0)
np.stack()	Stack arrays along a new dimension.	f = np.stack([a,b], axis=0)
np.hstack() / np.vstack()	Horizontal/vertical stack.	g = np.hstack([a,b])
np.split() / np.array_split()	Split array into sub-arrays.	h = np.split(a, 2)
np.dstack()	Depth (z-axis) stacking.	i = np.dstack([a,b])
np.squeeze()	Remove single-dimensional entries.	j = np.squeeze(a)
np.expand_dims()	Add a new axis.	k = np.expand_dims(a, 0)
np.pad()	Pad array borders.	l = np.pad(a, 1, mode='constant')
np.take()	Take elements from an array along an axis.	m = np.take(a, [0,2], axis=1)
np.put()	Assign to array using flat indices.	np.put(a, [1], 99)
np.where()	Return indices where a condition is true (or conditional selection).	idx = np.where(a>0)
np.take_along_axis()	Take elements along a specified axis.	n = np.take_along_axis(a, idx, axis=0)
np.unique()	Find unique elements.	u = np.unique(a)

Indexing and Slicing Commands

Operation	Description	Example
Basic slicing a[1:3, 2]	Select rows 1-2, column 2.	b = a[1:3, 2]
Boolean mask a[a>5]	Elements > 5.	c = a[a>5]
Fancy indexing a[[0,2], [1,3]]	Specific element pairs.	d = a[[0,2], [1,3]]
Ellipsis a[..., 0]	Index all but last dim.	e = a[..., 0]
Newaxis a[:, np.newaxis, :]	Insert a new axis.	f = a[:, None, :]
Slice assignment a[0] = 10	Modify slice.	a[0] = np.ones_like(a[0])*10
In-place operations a += 1	Modify array directly.	a += 1

Element-wise Arithmetic

Function	Description	Example
np.add(), +	Element-wise addition.	c = a + b
np.subtract(), −	Subtraction.	d = a − b
np.multiply(), *	Multiplication.	e = a * b
np.divide(), /	Division (float).	f = a / b
np.floor_divide()	Integer division.	g = np.floor_divide(a, b)
np.mod() / %	Modulus.	h = a % 3
np.remainder()	Remainder.	i = np.remainder(a, 3)
np.fmod()	Floating-point remainder.	j = np.fmod(a, 3)
np.power() / **	Exponentiation.	k = a ** 2
np.exp()	e^x.	l = np.exp(a)
np.log(), np.log10()	Natural / base-10 log.	m = np.log(a)
np.sqrt()	Square root.	n = np.sqrt(a)
np.abs()	Absolute value.	o = np.abs(a)
np.sign()	Sign of each element.	p = np.sign(a)

Aggregations & Reductions

Function	Description	Example
np.sum()	Sum over axes.	s = np.sum(a, axis=0)
np.mean()	Arithmetic mean.	mean = np.mean(a)
np.std(), np.var()	Standard deviation / variance.	std = np.std(a)
np.min(), np.max()	Minimum / maximum.	min_val = np.min(a)
np.argmin(), np.argmax()	Indices of min/max.	idx = np.argmax(a)
np.cumsum()	Cumulative sum.	csum = np.cumsum(a)
np.cumprod()	Cumulative product.	cprod = np.cumprod(a)
np.nanmean(), np.nansum()	Ignore NaNs.	nan_mean = np.nanmean(a)
np.percentile()	Percentiles.	p90 = np.percentile(a, 90)
np.histogram()	Histogram of data.	hist, edges = np.histogram(a)

Linear Algebra

Function	Description	Example
np.dot() / @	Matrix multiplication.	c = a @ b
np.linalg.inv()	Matrix inverse.	A_inv = np.linalg.inv(A)
np.linalg.det()	Determinant.	det = np.linalg.det(A)
np.linalg.eig()	Eigenvalues & eigenvectors.	w, v = np.linalg.eig(A)
np.linalg.svd()	Singular Value Decomposition.	U, s, Vt = np.linalg.svd(A)
np.linalg.norm()	Norm of a vector/matrix.	norm = np.linalg.norm(a)
np.linalg.solve()	Solve linear system Ax=b.	x = np.linalg.solve(A, b)
np.linalg.lstsq()	Least-squares solution.	x, residuals = np.linalg.lstsq(A,b,rcond=None)
np.linalg.eigh()	Eigen decomposition for Hermitian matrices.	w, v = np.linalg.eigh(H)
np.linalg.qr()	QR decomposition.	Q, R = np.linalg.qr(A)

Random & Probability Utilities

Function	Description	Example
np.random.rand()	Uniform [0,1).	x = np.random.rand(5)
np.random.randn()	Standard normal.	y = np.random.randn(5)
np.random.randint()	Random integers.	z = np.random.randint(0,10,size=5)
np.random.choice()	Random sample from array.	sample = np.random.choice(a, size=3, replace=False)
np.random.shuffle()	In-place shuffling.	np.random.shuffle(a)
np.random.permutation()	Return a permuted array.	perm = np.random.permutation(a)
np.random.normal()	Normal with given mean/std.	n = np.random.normal(0, 1, size=5)
np.random.binomial()	Binomial distribution.	b = np.random.binomial(10, 0.5, size=5)
np.random.multivariate_normal	Multivariate normal.	mvn = np.random.multivariate_normal(mean, cov, size=5)
np.random.dirichlet()	Dirichlet distribution.	d = np.random.dirichlet(alpha, size=5)

Other-Commands

Function	Description	Example
np.isnan(), np.isfinite()	Boolean masks for NaNs / infinities.	mask = np.isnan(a)
np.isclose()	Element-wise tolerance comparison.	c = np.isclose(a, b)
np.clip()	Clip values to a specified range.	c = np.clip(a, 0, 1)
np.round() / np.floor() / np.cei	Rounding functions.	r = np.round(a, 2)
np.sort() / np.argsort()	Sort arrays.	sorted_a = np.sort(a)
np.unique(return_index=True,	Get unique values and related info.	u, idx, inv, cnt = np.unique(a, return_index=True, return_inverse=True, return_counts=True)
np.meshgrid()	Create coordinate matrices.	X,Y = np.meshgrid(np.linspace(0,1,3), np.linspace(0,2,4))
np.histogram2d()	2-D histogram.	H, xedges, yedges = np.histogram2d(x, y)
np.interp()	1-D linear interpolation.	y_interp = np.interp(x_new, x_old, y_old)