

# PYTHON3 CHEAT SHEET

Klaus Rheinberger, FH Vorarlberg, *Energietechnik und Energiewirtschaft*  
2016-08-17, Version 3

## Help

command	usage
help()	interactive help, type quit to quit
object?, help(object)	help for object
object??	extended help for object
type(X)	type of object X

## Importing Packages and Functions

command	usage
import numpy as np	explicitly import numpy using name np
from scipy.optimize import linprog	import function from module

## Arrays: Creation

command	usage
array([[1,2,3],[4,5,6]])	create array
arange(min, max, step)	integer 1-D array evenly spaced
frange(min, max, step)	float 1-D array evenly spaced
linspace(min, max, num)	float 1-D array evenly spaced
A*B	<b>elementwise</b> multiplication
A**n	<b>elementwise</b> exponentiation
A@B, dot(A,B), A.dot(B)	matrix product
matrix_power(A, n)	exponentiation
zeros, ones, eye, identity	special arrays
diag(v)	diagonal array from vector
meshgrid(x,y)	create coordinate matrices

## Arrays: Indexing

command	usage
A[row,col]	access element row,col
A[min:max,min:max]	range access [min,max)
A[k,:]	row k
A[:,k]	column k
diag(A)	diagonal elements of A
A[row_index_list,col_index_list]	access elements via index lists
A[where(A ≥ 2)]	elements greater 2

## Arrays: Properties and Methods

command	usage
A.ndim	number of dimensions
A.shape, shape(A)	tuple (n,m) for an n times m matrix
A.size	number of entries
A.ravel(), ravel(A)	convert to 1-D
A.dtype	data type of entries
multiply(M, M)	elementwise multiplication
power(M,n)	elementwise exponentiation
sum(A, axis=d)	sum elements along axis d
mean(A, axis=d)	mean along axis d
std(A, axis=d)	standard deviation along axis d
min(A, axis=d)	min along axis d
max(A, axis=d)	max along axis d
matrix_rank(A)	rank
det(A)	determinant
inv(A)	inverse matrix of A
eig(A)	eigenvalues and eigenvectors
A.T, transpose(A)	transpose of A
bmat	block matrix construction
concatenate(tuple, axis)	join tuple of matrices/arrays
vstack	vertical stack of n-D arrays
hstack	horizontal stack of n-D arrays
column_stack	stack 1-D arrays as columns into a 2-D array
row_stack	stack 1-D arrays as rows into a 2-D array

## Data IO

command	usage
savetxt	save an array to a text file
loadtxt, genfromtxt	load data from a text file
savefig	save figure to file

## Graphics: 2D

command	usage
figure(figsize=(10, 8))	create figure
subplot(nrows, ncols, plot_number)	create subplot
plot(x, y, '-og', kwargs)	plot x versus y in green circle markers
xlim, ylim	set x and y limits
xlabel, ylabel	set x and y labels
title	set title
legend	create legend
grid	turn the axes grids on or off
hold	set the hold state
semilogx, semilogy	plot with semi-log axis
loglog	double log plot
stem	stem plot
bar	bar plot
polar	polar plot
hist	histogram
errorbar	plot with error bars
contour	contour plot

## Graphics: 3D

Example: 3D-Plot of the function  $z(x, y) = 3x^2 - 2xy + 5y^2$

```
from mpl_toolkits.mplot3d.axes3d import Axes3D
x = linspace(-3, 3, 100)
y = linspace(-3, 3, 100)
X, Y = meshgrid(x, y)
Z = 3*X**2 - 2*X*Y + 5*Y**2
fig = figure(figsize=(9,9))
ax = fig.add_subplot(1, 1, 1, projection='3d')
p = ax.plot_surface(X, Y, Z, rstride=4, cstride=4,
                   linewidth=0, cmap=cm.coolwarm)
```

## Links

- [SciPy Array Tip Sheet](#)
- [NumPy for MATLAB users](#)
- [Python & PyLab Cheat Sheet](#)