Python Cheat Sheet

Getting Help

`help()`	interactive help
`help(object)`	help for object
`object?`	iPython: help for object
`dir(object)`	display members of object

Import Syntax

`import numpy`	use: numpy.pi
`import numpy as np`	use np.pi
`from numpy import pi`	use: pi

Types

`i = 1`	integer
`f = 1.0`	float
`True/False`	boolean
`l = [3,2,1]`	list
`d = {'three':3, 'two':2}`	dictionary
`i = int(f)`	integer conversion
`f = float(i)`	float conversion

Operators

mathematics		comparison	
`+`	addition	`=`	assign
`-`	subtraction	`==`	equal
`*`	multiplication	`!=`	not equal
`/`	division	`<`	less
`**`	power	`<=`	less-equal
`%`	modulo	`>=`	greater-equal
		`>`	greater

Basic Syntax

`def bar(args): ...`	function definition
`if c: .. elif c: ... else:`	conditionals
`try: ... except: ...`	error handling
`try: ... except Error as e:`	error handling
`while condition: ...`	while loop
`for item in list: ...`	for loop

NumPy

The following assumes that NumPy has been imported using `import numpy as np`.

Maths

`np.abs(f)`	absolute value of f
`np.floor(f)`	round f downwards
`np.ceil(f)`	round f upwards
`np.sqrt(f)`	square root of f
`np.sin(f)`	sinus of f (in radians)
`np.cos, np.tan, ...`	similar
`np.arctan2(y,x)`	arctangent of point (x, y)

Defining arrays

`l = [1,2,3,4]`	basic list
`np.array([1,2,3,4])`	1D array
`np.array([[1,2],[3,4]])`	2D array
`np.arange(min,max,step)`	integer list: min to max
`np.linspace(min,max,num)`	num samples: min to max
`np.zeros((2,3))`	array of zeros, shape (2,3)
`np.ones((2,3))`	array of ones, likewise

Slicing

`l[row][col]`	list: basic access
`l[min:max:step]`	list: slicing
`arr[row][col]`	array: basic access 1
`a[row,col]`	array: basic access 2
`arr[min:max,min:max]`	array: slicing
`arr[list]`	select indices in list
`arr[mask]`	select where mask=True

Array properties

`len(l)`	length of first dimension
`arr.size`	total number of entries
`arr.ndim`	number of dimensions
`arr.shape`	shape off arr
`arr.reshape((N,M))`	reshape array to (N,M)

Linear Algebra

`a1*a2`	element-wise product (a1[0]*a2[0], ...)
`np.dot(a1,a2)`	vector dot product
`np.dot(a1,a2)`	matrix mult (if both 2D)
`np.cross(a1,a2)`	cross product
`np.linalg.inv(a)`	inverse of a
`np.linalg.det(a)`	determinant of a
`a.T`	transpose of a

Array statistics

`arr.sum(axis=i)`	sum of array elements along axis i
`arr.sum(axis=i)`	mean of array elements along axis i
`arr.std(axis=i)`	std. deviation along axis i
`arr.min(axis=i)`	min value along axis i
`arr.max(axis=i)`	max value along axis i
`arr.argmax`	index of maximum value
`arr.argmin`	index of minimum value

Miscellany

`np.loadtxt(file)`	read values from file
`np.genfromtxt(file)`	more flexible version
`np.any(arr)`	True if any of arr is True
`np.all(arr)`	True if all of arr is True
`np.random.normal()`	Gaussian random numbers
`np.random.uniform()`	Uniform random numbers

OS

Interaction with the operating system can be acheived using `import os` and `import shutil`

`os.mkdir(name)`	make directory 'name'
`os.unlink(file)`	delete file 'name'
`os.listdir(path)`	list all files in path
`os.rename(old,new)`	rename file/dir old to new
`os.path.exists(file)`	check if file/dir exists
`os.path.join(dir,file)`	join path and filename
`shutil.copy(src,dst)`	copy src to dst

Plotting

```
from matplotlib import pyplot as plt.
```

Plot Types

`fig,ax=plt.subplots`	create fig and axis
`ax.plot(x,y,'ro')`	plot x vs y with red points
`ax.plot(x,y,'k-')`	plot x vs y with black line
`ax.hist(vals,n_bins)`	histogram of vals
`ax.errorbar(x,y,yerr=e)`	like plot, with error bars
`ax.set_yscale('log')`	put y(x)-axis on log scale
`ax.set_title()`	set plot title
`ax.set_ylabel()`	set y(x) axis labels
`ax.set_ylim(min,max)`	set y(x) scale

File IO

`f = open(name)`	open name: read-only
`f = open(name,'w')`	open name: writing
`f = open(name,'a')`	open name: append
`f.read()`	read file into one string
`f.readlines()`	read lines into list of strings
`f.write(s)`	write string s to file

String Methods

`s.isdigit()`	True if s is all digit chars
`s.lower()`	lower case copy of s
`s.upper()`	upper case copy of s
`s.lstrip()`	strip leading whitespace
`s.lstrip()`	strip leading whitespace
`s.rstrip()`	strip trailing whitespace
`s.split(char)`	split string at char
`s.endswith(s)`	ends with s?
`s.replace(old,new)`	swap old for new

List Methods

`l.append(item)`	add item to list
`l.count(item)`	how often is item is l?
`l.index(item)`	loc of item in l
`l.insert(pos,item)`	insert item at pos
`l.remove(item)`	remove item from l
`l.reverse()`	reverse l
`l.sort()`	sort l

Times and Dates

Astropy has a very useful library for dealing with times and dates. Examples here assume that this library has been imported with `from astropy.time import Time` and `from astropy.time import TimeDelta`.

A string can be converted to a `Time` object using the following syntax examples:

```
t=Time('2015-10-22 12:15:22')
t=Time('2015-10-22 12:15')
t=Time('2015-10-22')
```

`TimeDelta` objects store the difference between two times. They can be created using the following syntax:

`dt = TimeDelta(3,format='sec')` - 3 seconds
`dt = TimeDelta(3,format='jd')` - 3 days

Astropy is quite clever at interpreting different formats of time strings. For full docs see `http://astropy.readthedocs.org/en/latest/time/`.

`t=Time.now()`	get current time and store in t
`t.iso`	get a Y-M-D H:M:S string from t
`t.mjd`	convert t to modified Julian date
`dt = t1-t2`	get time between t1 and t2
`t = t + dt`	add TimeDelta and Time

String Formatting

String formatting follows the general pattern ``{} {}''.format(arg1, arg2) to replace the curly braces with the values supplied in the arguments.

The exact appearance of the text that replaces the curly braces can be controlled by format characters. The format characters are preceded by a colon within the curly braces. To run the examples below use

`print(''FORMAT''.format(NUMBER))`.

So to get the output of the first example, you would run: `print(''{:.2f}''.format(3.1415926))`.

Number	Format	Output	Description
3.1415926	{:.2f}	3.14	2 d.p
3.1415926	{:+.2f}	3.14	2 d.p with sign
-1	{:+.2f}	-1.00	2 d.p with sign
5	{:05d}	00005	5 digits, pad with 0s
5	{:5d}	5	right aligned, width 10
10000	{:,}	10,000	comma seperator
1000000	{:.2e}	1.00e+06	sci. notation

SciPy