

Python Cheatsheet

Steve Young

ABSTRACT: Everything I know about Python.

Contents

1	Python	1
1.1	Lists and Dicts	1
1.1.1	Lists	1
1.1.2	Dicts	2
1.2	Sorting	2
1.3	Data Processing	2
1.4	Misc	2
1.5	re	2
2	Numpy	3
2.1	Data Processing	3
2.2	Number generation	3
2.3	NDArray handling	3
2.4	NDArray ops	3
2.5	Linear Algebra	4
3	Matplotlib	5
4	Pandas	5
5	Scikit-learn	5
6	packages to try	5

TODO: make sidebar marker telling what python library command depends on (and needs to be loaded beforehand)

1 Python

1.1 Lists and Dicts

1.1.1 Lists

With list name `foo`:

Init list of length `<n>` with `<constant>`: `foo = [<constant>] * <n>`

Add `<val>` to end of list: `foo.append(<val>)`

Remove list item at pos <n> (or end if <n> unspecified): `foo.pop(<n>)`

Insert <val> into list before position <n>: `foo.insert(<n>, <val>)`

Sum of list: `sum(foo)`

Get indexed elements as tuples (index, el) from list or ndarray: `enumerate(list)`

1.1.2 Dicts

Dictionaries are key-value stores, *i.e.* hashtables. With dictionary name `foo`:

Add <key>-<value> pair: `foo[<key>] = <value>`

Iterate through <key>-<value> pairs: `for (key, value) in foo.items()`

Test if <key> in dict: `<key> in foo`

1.2 Sorting

With sortable-thing name `foo`:

Sort (modify in place): `foo.sort()`

Sort (make a copy): `foo.sorted()`

Specify field to sort by (here by 2nd element of tuple): `foo.sorted(key=lambda x: x[1])`

Sort in reverse order: `foo.sorted(reverse=True)`

1.3 Data Processing

Read data from json file: `with open('data.json', 'r') as f: data = json.load(f)`

Write data to json file: `with open('data.json', 'w') as f: data = json.dump(f)`

Regex processing of text, re package: look at <https://docs.python.org/3/library/re.html>

1.4 Misc

Return integer representing Unicode <char>: `ord(<char>)`

1.5 re

2 Numpy

Using import numpy as np:

2.1 Data Processing

Import data from csv file: `np.genfromtxt('filename', delimiter=',')`

2.2 Number generation

Constant matrix of <val>: `np.full(shape, <val>)`

Matrix of ones/zeros: `np.ones(shape), np.zeros(shape)`

Id matrix: `np.eye(dim)`

Uniform dist on (low,high): `np.random.uniform(low, high, numsamps or shape)`

Uniform dist on (0,1) with given dims: `np.random.rand(d1, d2, ..)`

Normal dist: `np.random.normal(mean, stddev, numsamps)`

Normal dist on with given dims: `np.random.randn(d1, d2, ..)`

Multivariate normal: `np.random.multivariate_normal(..args)`

Random permutation of elements in ndarray: `np.random.permutation(NDArray)`

Permute elements of (range or ndarray) *in place*: `np.random.shuffle(int or NDArray)`

Integers over specified range: `np.arange(start, stop)`

Evenly spaced numbers over range w/ int stepsize: `np.arange(beg, end, stepsize)`

numvals Evenly spaced numbers over range: `np.linspace(beg, end, numvals)`

2.3 NDArray handling

- NDArrays are naturally *row vectors*, and of shape $(m,)$.

Reshape array: `np.reshape(NDArray, tuple of shape)`

2.4 NDArray ops

max/min element of array: `np.max(NDArray), np.min(NDArray)`

index of max/min element of array: `np.argmax(NDArray), np.argmin(NDArray)`

fill diagonal of sq matrix: `np.fill_diagonal(NDArray, val)`

round elements to nearest int: `np.rint(NDArray)`

return bin counts in histogram: `np.histogram(NDArray, binboundaries)`

<n>th difference of array: ¹ `np.diff(NDArray, <n>)`

2.5 Linear Algebra

Inverse matrix: `np.linalg.inv(square NDArray)`

Transpose matrix: `np.linalg.transpose(NDArray)`

evals and right evecs: `np.linalg.eig(square NDArray)`

¹Think transforming array of tick prices into array of tick prices *changes*

3 Matplotlib

Using `import matplotlib as mpl, import matplotlib.pyplot as plt:`

show image (if not in inline mode): `plt.show()`

plot image: `plt.imshow(NDArray)`

set axis bounds: `plt.axis([xmin, xmax, ymin, ymax])`

set x,y axis label: `plt.xlabel(name), plt.ylabel(name)`

set plot title: `plt.title(name)`

show plot legend: `plt.legend()`

visualize matrix vals as heat map: `plt.matshow(NDArray)`

pan/zoomable plots in PyCharm: `import matplotlib.pyplot as plt`
`mpl.use('Qt5agg')` before `import matplotlib.pyplot`
`as plt`

4 Pandas

5 Scikit-learn

Cross Validation: `sklearn.model_selection.cross_val_score`

6 packages to try

text from PDFs: `PyPDF2`