Python For Data Science *Cheat Sheet*

Python Basics

Learn More Python for Data Science Interactively at www.datacamp.com

Variables and Data Types

Variable Assignment

```
>>> x=5
>>> x
 5
```

Calculations With Variables

```
>>> x+2
 7
```
Sum of two variables
```
>>> x-2
 3
```
Subtraction of two variables
```
>>> x*2
 10
```
Multiplication of two variables
```
>>> x**2
 25
```
Exponentiation of a variable
```
>>> x%2
 1
```
Remainder of a variable
```
>>> x/float(2)
 2.5
```
Division of a variable

Types and Type Conversion

`str()`	`'5'`, `'3.45'`, `'True'`	Variables to strings
`int()`	`5, 3, 1`	Variables to integers
`float()`	`5.0, 1.0`	Variables to floats
`bool()`	`True`, `True`, `True`	Variables to booleans

Asking For Help

```
>>> help(str)
```

Strings

```
>>> my_string = 'thisStringIsAwesome'
>>> my_string
'thisStringIsAwesome'
```

String Operations

```
>>> my_string * 2
'thisStringIsAwesomethisStringIsAwesome'
>>> my_string + 'Innit'
'thisStringIsAwesomeInnit'
>>> 'm' in my_string
 True
```

Lists

Also see NumPy Arrays

```
>>> a = 'is'
>>> b = 'nice'
>>> my_list = ['my', 'list', a, b]
>>> my_list2 = [[4,5,6,7], [3,4,5,6]]
```

Selecting List Elements

Index starts at 0

Subset
```
>>> my_list[1]
```
Select item at index 1
```
>>> my_list[-3]
```
Select 3rd last item

Slice
```
>>> my_list[1:3]
```
Select items at index 1 and 2
```
>>> my_list[1:]
```
Select items after index 0
```
>>> my_list[:3]
```
Select items before index 3
```
>>> my_list[:]
```
Copy my_list

Subset Lists of Lists
```
>>> my_list2[1][0]
>>> my_list2[1][:2]
```
my_list[list][itemOfList]

List Operations

```
>>> my_list + my_list
['my', 'list', 'is', 'nice', 'my', 'list', 'is', 'nice']
>>> my_list * 2
['my', 'list', 'is', 'nice', 'my', 'list', 'is', 'nice']
>>> my_list2 > 4
True
```

List Methods

```
>>> my_list.index(a)
```
Get the index of an item
```
>>> my_list.count(a)
```
Count an item
```
>>> my_list.append('!')
```
Append an item at a time
```
>>> my_list.remove('!')
```
Remove an item
```
>>> del(my_list[0:1])
```
Remove an item
```
>>> my_list.reverse()
```
Reverse the list
```
>>> my_list.extend('!')
```
Append an item
```
>>> my_list.pop(-1)
```
Remove an item
```
>>> my_list.insert(0,'!')
```
Insert an item
```
>>> my_list.sort()
```
Sort the list

String Operations
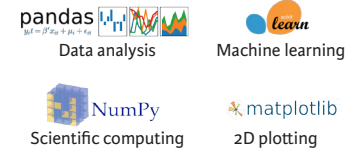
Index starts at 0

```
>>> my_string[3]
>>> my_string[4:9]
```

String Methods

```
>>> my_string.upper()
```
String to uppercase
```
>>> my_string.lower()
```
String to lowercase
```
>>> my_string.count('w')
```
Count String elements
```
>>> my_string.replace('e', 'i')
```
Replace String elements
```
>>> my_string.strip()
```
Strip whitespaces

Libraries

Import libraries
```
>>> import numpy
>>> import numpy as np
```
pandas — Data analysis
scikit-learn — Machine learning

Selective import
```
>>> from math import pi
```
NumPy — Scientific computing
matplotlib — 2D plotting

Install Python

ANACONDA — Leading open data science platform powered by Python

spyder — Free IDE that is included with Anaconda

jupyter — Create and share documents with live code, visualizations, text, ...

Numpy Arrays

Also see Lists

```
>>> my_list = [1, 2, 3, 4]
>>> my_array = np.array(my_list)
>>> my_2darray = np.array([[1,2,3],[4,5,6]])
```

Selecting Numpy Array Elements

Index starts at 0

Subset
```
>>> my_array[1]
 2
```
Select item at index 1

Slice
```
>>> my_array[0:2]
 array([1, 2])
```
Select items at index 0 and 1

Subset 2D Numpy arrays
```
>>> my_2darray[:,0]
 array([1, 4])
```
my_2darray[rows, columns]

Numpy Array Operations

```
>>> my_array > 3
 array([False, False, False,  True], dtype=bool)
>>> my_array * 2
 array([2, 4, 6, 8])
>>> my_array + np.array([5, 6, 7, 8])
 array([6, 8, 10, 12])
```

Numpy Array Functions

```
>>> my_array.shape
```
Get the dimensions of the array
```
>>> np.append(other_array)
```
Append items to an array
```
>>> np.insert(my_array, 1, 5)
```
Insert items in an array
```
>>> np.delete(my_array, [1])
```
Delete items in an array
```
>>> np.mean(my_array)
```
Mean of the array
```
>>> np.median(my_array)
```
Median of the array
```
>>> my_array.corrcoef()
```
Correlation coefficient
```
>>> np.std(my_array)
```
Standard deviation