Python for Data Science
Basics
Cheat Sheet

f616 adapted from datacamp.com

Introductory Note

This document is an adaption of the original datacamp.org cheat sheet.

- https://www.datacamp.com/resources/cheat-sheets/python-for-data-science-cheat-sheet-for-beginners
- https://github.com/f616/Python-Basics-Cheat-Sheet

1   Variables and Data Types

Variable Assignment

```
1  x=5
2  x
3  >>> 5
```

Calculations With Variables

```
1  x+2    #Sum of two variables
2  >>> 7
3
4  x-2    #Subtraction of two variables
5  >>> 3
6
7  x*2    #Multiplication of two variables
8  >>> 10
9
10 x**2   #Exponentiation of a variable
11 >>> 25
12
13 x%2    #Remainder of a variable
14 >>> 1
15
16 x/float(2)   #Division of a variable
17 >>> 2.5
```

Types and Type Conversion

```
1  str()
2  '5', '3.45', 'True'   #Variables to strings
3
4  int()
5  5, 3, 1   #Variables to integers
6
7  float()
8  5.0, 1.0   #Variables to floats
9
10 bool()
11 True, True, True   #Variables to booleans
```

2   Libraries

pandas: Data analysis
NumPy: Scientific computing
matplotlib: 2D plotting
scikit-learn: Machine learning

Import Libraries

```
1  import numpy
2  import numpy as np
```

Selective import

```
1  from math import pi
```

3   Strings

```
1  my_string = 'thisStringIsAwesome'
2  my_string
3  >>> 'thisStringIsAwesome'
```

String Operations

```
1  my_string * 2
2  >>> 'thisStringIsAwesomethisStringIsAwesome'
3
4  my_string + 'Innit'
5  >>> 'thisStringIsAwesomeInnit'
6
7  'm' in my_string
8  >>> True
```

`Index starts at 0`

String Indexing

```
1  my_string[3]
2  my_string[4:9]
```

String Methods

```
1  my_string.upper()    #String to uppercase
2  my_string.lower()    #String to lowercase
3  my_string.count('w')   #Count String elements
4  my_string.replace('e', 'i')   #Replace String elements
5  y_string.strip()   #Strip whitespaces
```

4   NumPy Arrays

Also see Lists

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

Selecting Numpy Array Elements

`Index starts at 0`

Subset

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

Slice

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

Subset 2D Numpy arrays

```
1  my_2darray[:,0]    #my_2darray[rows, columns]
2  >>> array([1, 4])
```

Numpy Array Operations

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

Numpy Array Functions

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

5 Lists

Also see NumPy Arrays

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

Selecting List Elements

Index starts at 0

Subset

```
1 my_list[1]   #Select item at index 1
2 my_list[-3]   #Select 3rd last item
```

Slice

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

Subset Lists of Lists

```
1 my_list2[1][0]   #my_list[list][itemOfList]
2 my_list2[1][:2]
```

List Operations

```
1 my_list + my_list
2 >>> ['my', 'list', 'is', 'nice', 'my', 'list', 'is', 'nice']
3
4 my_list * 2
5 >>> ['my', 'list', 'is', 'nice', 'my', 'list', 'is', 'nice']
6
7 my_list2 > 4
8 >>> True
```

List Methods

```
1  my_list.index(a)   #Get the index of an item
2  my_list.count(a)   #Count an item
3  my_list.append('!')   #Append an item at a time
4  my_list.remove('!')   #Remove an item
5  del(my_list[0:1])   #Remove an item
6  my_list.reverse()   #Reverse the list
7  my_list.extend('!')   #Append an item
8  my_list.pop(-1)   #Remove an item
9  my_list.insert(0,'!')   #Insert an item
10 my_list.sort()   #Sort the list
```

6 Asking For Help

```
1 help(str)
```