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 |
|-----|-----|
| >>> | Х |
| 5 | |

Calculations With Variables

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

Types and Type Conversion

| | <u> </u> | |
|---------|---------------------|-----------------------|
| 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
```

Lists

```
>>> 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 o

Also see NumPy Arrays

Subset

| >>> | my_ | _list[1] |
|------|-----|----------|
| >>> | my_ | list[-3] |
| Slic | e ¯ | _ |

- >>> my list[1:3] >>> my list[1:] >>> my list[:3] >>> my list[:]
- **Subset Lists of Lists** >>> my list2[1][0]
- >>> my list2[1][:2]

Select item at index 1 Select 3rd last item

Select items at index 1 and 2 Select items after index o Select items before index 3 Copy my list

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
```

List Methods

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

String Operations

Index starts at o

```
>>> 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 Selective import

>>> from math import pi





Machine learning

```
NumPy
                     * matplotlib
Scientific computing
                      2D plotting
```

Install Python



Leading open data science platform powered by Python



Free IDE that is included with Anaconda



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 o

```
Subset
>>> my array[1]
```

Slice

```
>>> my array[0:2]
  array([1, 2])
Subset 2D Numpy arrays
>>> my 2darray[:,0]
  array([1, 4])
```

Select item at index 1

Select items at index 0 and 1

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
```

Python For Data Science Cheat Sheet

NumPy Basics

Learn Python for Data Science Interactively at www.DataCamp.com



NumPy

The **NumPy** library is the core library for scientific computing in Python. It provides a high-performance multidimensional array object, and tools for working with these arrays.

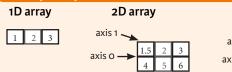
Use the following import convention: >>> import numpy as np



3D array

axis 2

NumPy Arrays



Creating Arrays

```
>>> a = np.array([1,2,3])
>>> b = np.array([(1.5,2,3), (4,5,6)], dtype = float)
>>> c = np.array([[(1.5,2,3), (4,5,6)], [(3,2,1), (4,5,6)]],
                 dtype = float)
```

Initial Placeholders

| >>> np.zeros((3,4)) >>> np.ones((2,3,4),dtype=np.int16) >>> d = np.arange(10,25,5) | Create an array of zeros Create an array of ones Create an array of evenly |
|--|--|
| >>> np.linspace(0,2,9) | spaced values (step value) Create an array of evenly |
| >>> e = np.full((2,2),7) >>> f = np.eye(2) | spaced values (number of samples) Create a constant array Create a 2X2 identity matrix |
| >>> np.random.random((2,2)) >>> np.empty((3,2)) | Create an array with random values Create an empty array |

1/0

Saving & Loading On Disk

```
>>> np.save('my array', a)
>>> np.savez('array.npz', a, b)
>>> np.load('my array.npy')
```

Saving & Loading Text Files

| >>> | np.loadtxt("myfile.txt") |
|-----|---|
| >>> | np.genfromtxt("my file.csv", delimiter=',') |
| >>> | np.savetxt("mvarrav.txt", a, delimiter=" ") |

Data Types

| >>> np.int64 | Signed 64-bit integer types |
|-----------------|--|
| >>> np.float32 | Standard double-precision floating point |
| >>> np.complex | Complex numbers represented by 128 floats |
| >>> np.bool | Boolean type storing TRUE and FALSE values |
| >>> np.object | Python object type |
| >>> np.string_ | Fixed-length string type |
| >>> np.unicode_ | Fixed-length unicode type |

Inspecting Your Array

| >>> | a.shape | Array dimensions |
|-----|---------------|--------------------------------------|
| >>> | len(a) | Length of array |
| >>> | b.ndim | Number of array dimensions |
| >>> | e.size | Number of array elements |
| >>> | b.dtype | Data type of array elements |
| >>> | b.dtype.name | Name of data type |
| >>> | b.astype(int) | Convert an array to a different type |

Asking For Help

>>> np.info(np.ndarray.dtype)

Array Mathematics

Arithmetic Operations

| >>> g = a - b | Subtraction |
|-------------------------------------|--------------------------------|
| array([[-0.5, 0. , 0.], | |
| [-3., -3., -3.]]) | Culturation |
| >>> np.subtract(a,b) | Subtraction |
| >>> b + a array([[2.5, 4., 6.], | Addition |
| [5., 7., 9.]]) | |
| >>> np.add(b,a) | Addition |
| >>> a / b | Division |
| array([[0.66666667, 1. , 1.], | 211.5.6.1 |
| | |
| >>> np.divide(a,b) | Division |
| >>> a * b array([[1.5, 4., 9.], | Multiplication |
| [4., 10., 18.]]) | |
| >>> np.multiply(a,b) | Multiplication |
| >>> np.exp(b) | Exponentiation |
| >>> np.sgrt(b) | Square root |
| >>> np.sin(a) | Print sines of an array |
| >>> np.cos(b) | Element-wise cosine |
| >>> np.log(a) | Element-wise natural logarithr |
| >>> e.dot(f) | Dot product |
| array([[7., 7.], | |
| [7., 7.]]) | |

Comparison

| >>> a == b array([[False, True, True], | Element-wise comparison |
|--|-------------------------|
| <pre>[False, False, False]], dtype=bool) >>> a < 2 array([True, False, False], dtype=bool)</pre> | Element-wise comparison |
| 4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 | Array-wise comparison |

Aggregate Functions

| >>> a.sum() | Array-wise sum |
|----------------------|--------------------------------|
| >>> a.min() | Array-wise minimum value |
| >>> b.max(axis=0) | Maximum value of an array row |
| >>> b.cumsum(axis=1) | Cumulative sum of the elements |
| >>> a.mean() | Mean |
| >>> b.median() | Median |
| >>> a.corrcoef() | Correlation coefficient |
| >>> np.std(b) | Standard deviation |

Copying Arrays

| >>> h = a.view() | Create a view of the array with the same data |
|------------------|---|
| >>> np.copy(a) | Create a copy of the array |
| >>> h = a.copy() | Create a deep copy of the array |

Sorting Arrays

| г | | |
|---|--------------------|--------------------------------------|
| | | Sort an array |
| | >>> c.sort(axis=0) | Sort the elements of an array's axis |

Subsetting, Slicing, Indexing

1 2 3

Subsetting

>>> a[2]

>>> b[1,2]

>>> a[0:2]

>>> b[:1]

Boolean Indexing

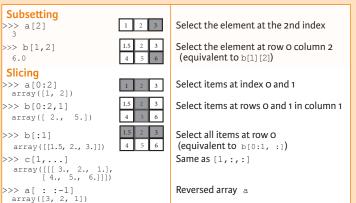
>>> a[a<2]

array([1])

Fancy Indexing

6.0 Slicing

Also see Lists



and columns

Array Manipulation

>>> b[[1, 0, 1, 0], [0, 1, 2, 0]]

array([4. , 2. , 6. , 1.5]) >>> b[[1, 0, 1, 0]][:,[0,1,2,0]]

| Tra | ın | sp | osing Array | |
|-----|----|-----|-----------------|--|
| >>> | i | = | np.transpose(b) | |
| >>> | i | . Т | | |

Changing Array Shape >>> b.ravel()

| ŀ | >>> g.reshape(3,-2) |
|---|---------------------------------|
| | Adding/Removing Elements |

| | >>> | h.resize((2,6)) |
|--|-----|--------------------|
| | >>> | np.append(h,g) |
| | >>> | np.insert(a, 1, 5) |
| | >>> | np.delete(a,[1]) |

Combining Arrays >>> np.concatenate((a,d),axis=0)

```
array([ 1, 2, 3, 10, 15, 20])
>>> np.vstack((a,b))
 array([[ 1., 2., 3.], [ 1.5, 2., 3.], [ 4., 5., 6.]])
>>> np.r [e,f]
>>> np.hstack((e,f))
array([[ 7., 7., 1., 0.],
         [ 7., 7., 0., 1.]])
>>> np.column stack((a,d))
 array([[ 1, 10],
           2, 15],
          [ 3, 20]])
>>> np.c [a,d]
```

Splitting Arrays

| | >>> np.hsplit(a,3) |
|--|---|
| | [array([1]),array([2]),array([3] |
| | >>> np.vsplit(c,2) |
| | [array([[[1.5, 2. , 1.], |
| | [4. , 5. , 6.]]]), array([[[3., 2., 3.], |
| | [4., 5., 6.]]])] |

Permute array dimensions Permute array dimensions

Select elements from a less than 2

Select elements (1,0), (0,1), (1,2) and (0,0)

Select a subset of the matrix's rows

Flatten the array Reshape, but don't change data

Return a new array with shape (2,6) Append items to an array Insert items in an array Delete items from an array

Concatenate arrays

Stack arrays vertically (row-wise)

Stack arrays vertically (row-wise) Stack arrays horizontally (column-wise)

Create stacked column-wise arrays

Create stacked column-wise arrays

Split the array horizontally at the 3rd

Split the array vertically at the 2nd index