



Python Basics Cheatsheet 3.0

Importing Files

```
>>> import numpy as np
>>> import pandas as pd
```

Help

```
>>> np.info(np.ndarray.dtype)
>>> help(pd.read_csv)
```

Plain Text Files

```
Open the file for reading >>> filename = 'huck_finn.txt'
Read a file's contents >>> file = open(filename, mode='r')
Check if file is closed >>> text = file.read()
Close file >>> print(file.closed)
>>> file.close() Close file
>>> print(text)
```

USING THE CONTEXT MANAGER WITH

```
>>> with open('huck_finn.txt', 'r') as file:
    print(file.readline()) Read a single line
    print(file.readline())
    print(file.readline())
```

Table Data | Importing Flat files with NumPy

Files with one data type

```
>>> filename = 'mnist.txt'
>>> data = np.loadtxt(filename,
                        delimiter=',', String used to separate values
                        skiprows=2, Skip the first 2 lines
                        usecols=[0,2], Read the 1st and 3rd column
                        dtype=str) The type of the resulting array
```

Files with mixed data types

```
>>> filename = 'titanic.csv'
>>> data = np.genfromtxt(filename,
                        delimiter=',',
                        names=True, Look for column header
                        dtype=None)
```

Table Data | Importing Flat files with Pandas

```
i >>> filename = 'winequality-red.csv'
>>> data = pd.read_csv(filename,
                        nrows=5, Number of rows of file to read
                        header=None, Row number to use as col names
                        sep='\t', Delimiter to use
                        comment='#', Character to split comments
                        na_values=[""]) String to recognize as NA/NaN
```

Excel Spreadsheets

```
>>> file = 'urbanpop.xlsx'
>>> data = pd.ExcelFile(file)
>>> df_sheet2 = data.parse('1960-1966',
                        skiprows=[0],
                        names=['Country',
                              'AAM: War(2002)'])
>>> df_sheet1 = data.parse(0,
                        parse_cols=[0],
                        skiprows=[0],
                        names=['Country'])
```

SAS Files

```
>>> from sas7bdat import SAS7BDAT
>>> with SAS7BDAT('urbanpop.sas7bdat') as file:
    df_sas = file.to_data_frame()
```

Relational Databases

```
>>> from sqlalchemy import create_engine
>>> engine = create_engine('sqlite://Northwind.sqlite')
```

Querying Relational Databases

```
>>> con = engine.connect()
>>> rs = con.execute("SELECT * FROM Orders")
>>> df = pd.DataFrame(rs.fetchall())
>>> df.columns = rs.keys()
>>> con.close()

Using the context manager with
>>> with engine.connect() as con:
    rs = con.execute("SELECT OrderID FROM Orders")
    df = pd.DataFrame(rs.fetchmany(size=5))
    df.columns = rs.keys()
```

Querying relational databases with Pandas

```
>>> df = pd.read_sql_query("SELECT * FROM
Orders", engine)
```

Exploring your Data- NumPy Arrays

```
>>> data_array.dtype Data type of array elements
>>> data_array.shape Array dimensions
>>> len(data_array) Length of array
```

Exploring your Data- Pandas DataFrames

```
>>> df.head() Return first DataFrame rows
>>> df.tail() Return last DataFrame rows
>>> df.index Describe index
>>> df.columns Describe DataFrame
columns
>>> df.info() Info on DataFrame
>>> data_array = data.values Convert a DataFrame to an
a NumPy array
```

Pickled Files

```
>>> import pickle
>>> with open('pickled_fruit.pkl',
'rb') as file:
    pickled_data = pickle.load(file)
```

HDF5 Files

```
>>> import h5py
>>> filename = 'H-
H1_LOSC_4_v1-815411200-
4096.hdf5'
>>> data = h5py.File(filename, 'r')
```

Matlab Files

```
>>> import scipy.io
>>> filename = 'workspace.mat'
>>> mat =
scipy.io.loadmat(filename)
```

Magic Commands

Navigating Filesystem

```
!ls List directory contents of files
and directories
%cd Change current working
directory
%pwd Return the current working
directory path
```