Script Style

An example: What is this code doing?

```
# -*- coding: utf-8 -*-
"""
Created on Wed Jan 30 16:00:31 2019
@author: rosm
"""

from pandas import *

df = read_csv('https://raw.githubusercontent.com/Michael-E-Rose/HasselbackFacul

df = df.dropna( subset = [ 'U2001_dep' ] )
df = df.drop(df.loc[df['U2001_rank']=='Retir'].index)
df.drop(df.loc[df['U2001_rank']=='Deces'].index, inplace=True)
df['U2001_dep2'] = df['U2001_dep'].str.upper()

df3 = DataFrame(df.U2001_dep2.value_counts().reset_index())
df3.columns = ['University', '# of profs at dep in 2001']
```

...and this one?

```
#!/usr/bin/env python
# Author:   Michael E. Rose <michael.ernst.rose@gmail.com>
"""Computes faculty size using Hasselback Faculty Rooster."""

import pandas as pd

DEPARTMENT_FILE = 'https://raw.githubusercontent.com/Michael-E-Rose/'\
    'HasselbackFacultyRoster/master/hasselback.csv'
_drops = ("Retir", "Deces")  # Faculty status we don't need

# Read in
sizedep = pd.read_csv(DEPARTMENT_FILE)
sizedep = sizedep.dropna(subset=['U2001_dep'])

sizedep = sizedep[~sizedep['U2001_rank'].isin(_drops)]

# Dataframe of # of people working at the Department
sizedep['U2001_dep'] = sizedep['U2001_dep'].str.upper()
sizedep2 = sizedep['U2001_dep'].value_counts().reset_index()
sizedep2.columns = ['University', '# of profs at dep in 2001']
```

The Zen of Python

```
1 import this
```

The Zen of Python

```
1 import this
```

"Code is read much more often than it is written" *(*Guido van Rossum*)*

Python Enhancement Proposal No. 8

- Use descriptive names for your objects
- Never use `l`, `O`, or `I` as variable names
- Surround top-level functions and classes with two blank lines
- Try to have each line end after 79 characters
- Use 4 consecutive spaces to indicate indentation
- Document code with complete plain English sentences
- Do not document obvious things
- Start module and functions with docstrings (surrounded by """)
- Add 1 whitespace around binary operators (+, −, =)
- Do not add whitespace around = assigning parameters
- Avoid trailing whitespaces

Script layout

```
1  #!/usr/bin/env python3
2  # Author:   Michael E. Rose <michael.ernst.rose@gmail.com>
3  """Docstring describing what that module does in 3rd person."""
4
5  import re
6  from collections import Counter, defaultdict
7
8  import pandas as pd
9  from numpy import nan
10
11 from script import read_file
12
13 CONSTANT1 = "./some_relative/path/file.csv"
14
15
16 def first_function(arg1):
17     """Docstring describing what the function does in imperative.
18
19     Parameters:
20     -----------
21     arg1 : str
22         What is arg about?
23     """
```

PEP8: Naming convention

Object	naming convention	example
Function	lowercase separated by underscores	`parse_patents()`
Variable	lowercase separated by underscores	`patent_text`
Class	camel case	`LogEstimator()`
Method	lowercase separated by underscores	`.fit_model()`
Constant	uppercase	`URL, CONFIG_FILE`
Module	lowercase separated by underscore	`patent_parser.py`

PEP8: Naming convention

Object	naming convention	example
Function	lowercase separated by underscores	`parse_patents()`
Variable	lowercase separated by underscores	`patent_text`
Class	camel case	`LogEstimator()`
Method	lowercase separated by underscores	`.fit_model()`
Constant	uppercase	`URL, CONFIG_FILE`
Module	lowercase separated by underscore	`patent_parser.py`

Full proposal at
https://www.python.org/dev/peps/pep-0008/

PEP8 checker

```
$ pip install pycodestyle
$ pycodestyle my_script.py
my_script.py:2:21: E231 missing whitespace after ','
my_script.py:6:19: W293 blank line contains whitespace
```

PEP8 checker

```
$ pip install pycodestyle
$ pycodestyle my_script.py
my_script.py:2:21: E231 missing whitespace after ','
my_script.py:6:19: W293 blank line contains whitespace

$ pip install black
$ black my_script.py
reformatted my_script.py
All done!
```

Relative paths

- Relative path: starts from some given working directory (i.e. where the script is executed from)
- Absolute path: points to the same location in a file system, regardless of the current working directory

Relative paths

- Relative path: starts from some given working directory (i.e. where the script is executed from)
- Absolute path: points to the same location in a file system, regardless of the current working directory

```
1  import pandas as pd
2
3  FNAME = "C:\Users\rosm\Dropbox\science_project\input_file.csv"
4
5  df = pd.read_csv(FNAME)
```

- Will the code run on my coauthor's computer?
- How should I best write it?

The __main__ function

- Magic function to define top-level and sub-level namespace
- Relevant if you want to import from this file
- Good coding practice to always include it and have main code (but not functions in it

```
1  def read_in(fname):
2      """Read filename."""
3      ...
4
5
6  def main():
7      ...
8
9
10 if __name__ == '__main__':
11     main()
```