

## Averager

Write a Python program called `averager.py` that takes one or more file names as positional arguments, finds all the numbers (integers or floats) in each document, and prints the average (using `{:10.02f}`) and the document's basename. Note that you will need to capture both positive and negative numbers. If an argument is not a file, print '“argument” is not a file' to *STDERR* and skip to the next argument.

## Expected Behavior

```
$ ./averager.py
usage: averager.py [-h] FILE [FILE ...]
averager.py: error: the following arguments are required: FILE
$ ./averager.py -h
usage: averager.py [-h] FILE [FILE ...]
```

Average all the numbers in a document

positional arguments:  
FILE Input file(s)

optional arguments:  
-h, --help show this help message and exit

```
$ ./averager.py flkdj
"flkdj" is not a file
$ ./averager.py inputs/const.txt
6.46: const.txt
$ ./averager.py inputs/*.txt
6.46: const.txt
-2.43: negative.txt
0.00: no_numbers.txt
890.00: usdeclar.txt
0.00: zero.txt
```

## Test Suite

A passing test suite looks like this:

```
$ make test
pytest -v test.py
===== test session starts =====
platform darwin -- Python 3.6.8, pytest-4.2.0, py-1.7.0, pluggy-0.8.1 -- /anaconda3/bin/pyth
```

```
cachedir: .pytest_cache
rootdir: /Users/kyclark/work/worked_examples/2019_spring_finals/averager_b, inifile:
plugins: remotedata-0.3.1, openfiles-0.3.2, doctestplus-0.2.0, arraydiff-0.3
collected 7 items
```

```
test.py::test_usage PASSED [ 14%]
test.py::test_bad_input PASSED [ 28%]
test.py::test_no_numbers PASSED [ 42%]
test.py::test_const PASSED [ 57%]
test.py::test_zero PASSED [ 71%]
test.py::test_all PASSED [ 85%]
test.py::test_all_and_bad PASSED [100%]
```

```
===== 7 passed in 0.49 seconds =====
```