

## head.py

Create a Python program called `head.py` that expects one or two arguments. If there are no arguments, print a “Usage” statement. The first argument is required and must be a regular file; if it is not, print “ is not a file” and exit *with an error code*. The second argument is optional. If given, it must be a positive number (non-zero); if it is not, then print “lines () must be a positive number”. If no argument is provided, use a default value of 3. You can expect that the test will only give you a value that can be safely converted to a number using the `int` function. If given good input, it should act like the normal `head` utility and print the expected number of lines from the given file.

## Expected behavior:

```
$ ./head.py
Usage: head.py FILE [NUM_LINES]
$ ./head.py foo
foo is not a file
$ ./head.py files/issa.txt
Selected Haiku by Issa
```

```
Don't worry, spiders,
$ ./head.py files/issa.txt 5
Selected Haiku by Issa
```

```
Don't worry, spiders,
I keep house
casually.
```

## Test Suite

A passing test suite looks like the following:

```
$ make test
python3 -m 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/python
cachedir: .pytest_cache
rootdir: /Users/kyclark/work/python/practical_python_for_data_science/ch05-python-strings-1
plugins: remotedata-0.3.1, openfiles-0.3.2, doctestplus-0.2.0, arraydiff-0.3
collected 4 items

test.py::test_usage PASSED [ 25%]
```

```
test.py::test_bad_number PASSED [ 50%]  
test.py::test_bad_input PASSED [ 75%]  
test.py::test_runs PASSED [100%]  
  
===== 4 passed in 0.54 seconds =====
```