

head.sh

Write a bash script called **head.sh** that mimics the **head** utility where it will print the first few lines of a file. The script should expect one required argument (the file) and a second optional argument of the number of lines, defaulting to 3. If are no arguments, it should print a “Usage” and exit *with an error code*. Your program will expect to receive an argument in **\$1** and maybe a second in **\$2**. If the first argument is not a file, it should notify the user and exit *with an error code*. If the second argument is missing, use the value “3”. Print the number of lines requested by the user by iterating over the lines in the file and exiting the loop appropriately. Do not use the actual **head** command!

Expected Behavior

```
$ ./head.sh
Usage: head.sh FILE NUM
$ ./head.sh files/issa.txt
Selected Haiku by Issa

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

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

Testing

You have been provided a **Makefile** that will run a test suite. This is what it should look like when all tests are passing:

```
$ 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/ch02-unix-exercises/exercises
plugins: remotedata-0.3.1, openfiles-0.3.2, doctestplus-0.2.0, arraydiff-0.3
collected 3 items

test.py::test_usage_head PASSED [ 33%]
test.py::test_bad_input_head PASSED [ 66%]
```

test.py::test_head_run PASSED

[100%]

===== 3 passed in 0.06 seconds =====

The first test is that the files exist. Then they are tested with no arguments to see if they produce “usage”-type help messages. Then they are tested with bad input. Then they are tested that they produce the expected output when given good input.

This is the basis of “test-driven design” (TDD). We define a set of tests that describe what working software should do. When the tests pass, the software is done. When you are passing all the tests, you are done!