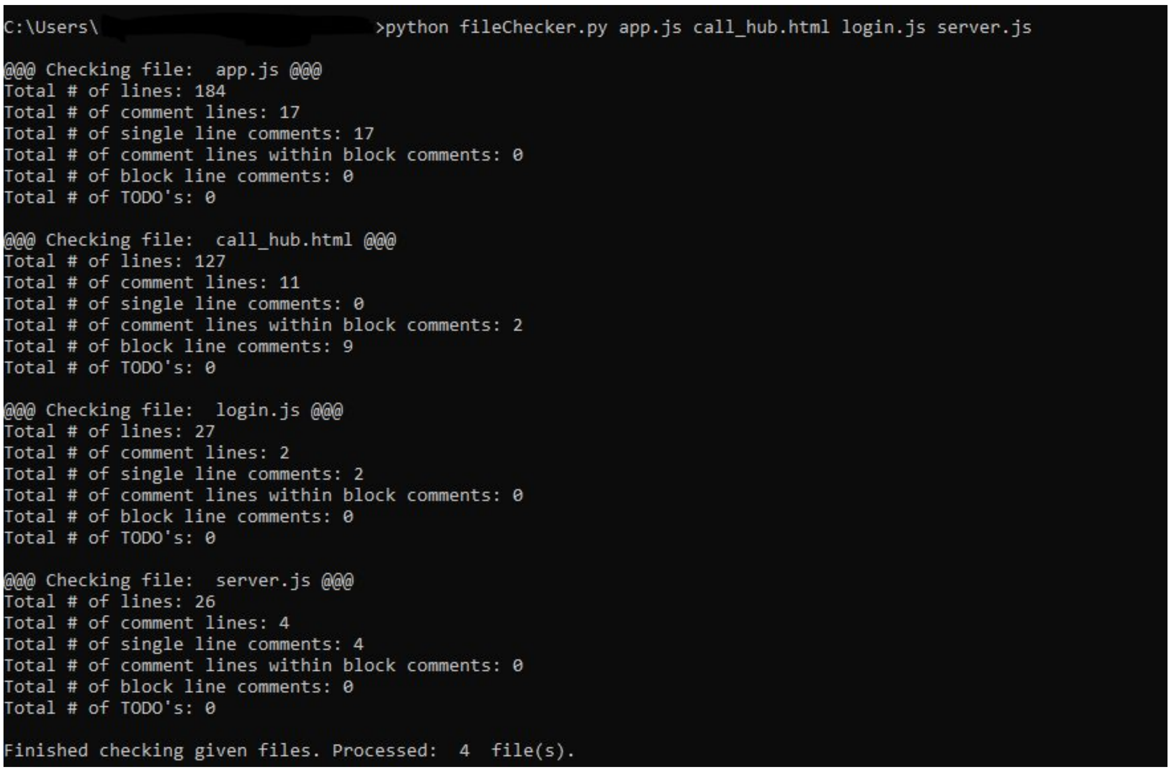
Nicholas Hebert
2019-10-01

FileChecker is written in Python v3.7.4 and gives comment statistics on given files. Python was selected for its ease of use since this will be tested by a foreign computer. To run the program simply type:

```
python fileChecker.py <file1> <file2> ...
```

Any number of files can be processed at once by the program. The output will be titled by filename and will be presented in the order inputted.

The following is a screenshot of a use case:



```
C:\Users\[redacted]>python fileChecker.py app.js call_hub.html login.js server.js

@@@ Checking file:  app.js @@@
Total # of lines: 184
Total # of comment lines: 17
Total # of single line comments: 17
Total # of comment lines within block comments: 0
Total # of block line comments: 0
Total # of TODO's: 0

@@@ Checking file:  call_hub.html @@@
Total # of lines: 127
Total # of comment lines: 11
Total # of single line comments: 0
Total # of comment lines within block comments: 2
Total # of block line comments: 9
Total # of TODO's: 0

@@@ Checking file:  login.js @@@
Total # of lines: 27
Total # of comment lines: 2
Total # of single line comments: 2
Total # of comment lines within block comments: 0
Total # of block line comments: 0
Total # of TODO's: 0

@@@ Checking file:  server.js @@@
Total # of lines: 26
Total # of comment lines: 4
Total # of single line comments: 4
Total # of comment lines within block comments: 0
Total # of block line comments: 0
Total # of TODO's: 0

Finished checking given files. Processed:  4  file(s).
```

If no input is given the program will stop. If a given file has no filetype, it will be skipped.
The program supports the following filetypes:

- Assembly
- C, C++, C#
- CSS
- Fortran (1990, 1995, 2003)
- HTML
- Javascript
- Java
- Pascal
- Perl
- PHP
- Ruby On Rails
- Shell
- SQL
- Swift
- Visual Basic

Other languages will default to the same commenting convention from C, C++, and C#. My assumption is the total # of comment lines is the sum of all comments. Comment lines within the block are the comments between the start and end comment block symbols. Block line comments are the start and end lines for the comment blocks. My last assumption is that users are not actively trying to alter the readings, the "" symbol after a single line comment cannot be read by the program due to limiting lookbehind functions with regular expressions. Also adding the block start or end comment symbol within a string will be read by the program as a start or end and can skew comment counting. Future steps would be to find a new solution to substitute regular expressions and use that for both single-line and block comments searching.