Peer-review of assignment 6 for *INF3331-Shayan*

Reviewer 1, Git-repo INF3331-AndreasLoland, andrelt@student.matnat.uio.no
Reviewer 2, Git-repo INF3331-KimMadsen, kimmp@mail.uio.no

Deadline: Tuesday, 8. December 2015, 23:59:59.

1 Introduction

The code where tested on the following systems:

- OS X 10.11, Red Hat Enterprise Linux Workstation 7.2, Python 2.7.10

2 Review

The code is generally well documented and easy to understand, however, the sphinx-style is only used some places. It would be nice to have documented consistently across all the code, especially to be able to use the magic command ”?” to get rich documentation for functions in these programs. You could also be a little more consistent with the comments: since some are placed over code and some next to it, it makes it a little less readable.

Good naming of variables and functions. You could be more consistent with the use of single and double quotation-marks (’/”).

Part from this we think this is a well executed assignment.

Assignment 6.1: Feedline

In your Readme.md file you explain how to use feedline.py quite well. So the issue where you have to make the lineNumber and namespace variables applies only to running feedline-function alone (which doesn’t matter).

```
python
lineNumber=0
namespace=vars().copy()
from feedline import feedline
print feedline("print ’Hello World!’",namespace)
```

This use of accessing and manipulating ”global” variable lineNumber through the local namespace is quite nice!

Assignment 6.2 Create basic interface

The code in the main-function is very, very long. Of course, all the necessary checks leads to long code, but if you had separated the functionality in functions it would be a lot more readable. Also, we think the assignment asked for main to just call a prompt-function, this is not implemented.

Assignment 6.3 Error handling

Works as expected. Handles what we have tried.

Assignment 6.4 Control sequence

Ctrl-d works as expected. If you hit ctrl-d with a non empty prompt/buffer, it is cleared with a ”KeyboardInterupt”-message. Otherwise the system exits with a thanks message.

Assignment 6.5 Up and down characters and command history

This implementation is done in mypython.py. Cool that you have bothered to implement left/right-key!

Up and down chars are handled as described in the assignment. To be a bit picky, we would have preferred if the prompt was emptied if the down-key is pressed again when the latest used command is displayed. At the moment you have to remove the line to be able to write anything else than what you wrote last time.

Great implementation of TAB-completion, even though this is not part of the 3331-assignment which we assume you are part of based on the repo-name.

Assignment 6.6 Magic commands

The magic-commands works for both mypython.py and my_webserver.py. This is partly because they are implemented in separate files, which feels a little excessive since there is so little code. Separate functions in feedline.py could work just as fine.

The %save-command does not work, we get NameError and then the program crashes. This is because the ***commandHistory-list*** is declared in **mypython.py**, but **feedline()** tries to split and send it to **savecommand.py**.

The other two functions works partly as expected. The output is not handled correctly. The output of !- and ?-functions should not go to out[ ], but rather be printed to screen as shown in the assignment-example.

The help-command works well. An improvement could be to also apply the functionality to the users own variables within the iPython-clone. **Help()** takes either a string or an object as parameter. Since you only give it a string (which checks for names of modules, types etc.) it is not possible to for example do this:

```
In [1]: x = 1
In [2]: x?
Out[2]: no Python documentation found for 'x'
```

If you had called help() with x in the namespace as an object, the user would get the documentation for integers.

Assignment 6.7 Webinterface

The webinteface is working as expected.

We miss some commenting in both html.file and your my_webserver.py though!