```
##     ## ######## ##     ## ######## ######## ##     ##
##   ##  ##       ##  ##   ##    ##    ##       ###    ##
##  ##   ##       ##  ##   ##    ##    ##       ####    ##
#####    ######## ##     ##    ##    ######   ## ## ##
##  ##   ##   ##    ##     ##    ##    ##       ##  ####
##   ##  ##    ##    ##     ##    ##    ##       ##   ###
##     ## ##     ##    ##     ##    ######## ##     ##
> That's easy for you to say, Mr David, you're a human
```

What is Kryten?

Kryten is a tool that makes terminal-based presentations a snap to write, present, and share.

Kryten takes it's name from Kryten of the British television series Red Dwarf.

You can see an PyCon presentation using Kryten on YouTube.

Why Kryten?

Kryten allows presenters to focus on the presentation, not the code. With Kryten, you can present live code demos from prewritten presentation files. Write the presentation beforehand, and play it, step-by-step with Kryten.

Installing Kryten

```
pip install kryten
```

or download kryten.py from this repo.

Usage

Writing a presentation

Kryten works with .play files.

Syntax

Start comments with a #

Mark file edits with `@@UPDATE@@ path.to/file.ext`, followed by the new file contents, ending with `@@END@@`.

A simple example

This example presentation presents a couple Python loops.

`python.play`

```
# a simple loop
for i in range(5):
    print i

# another loop
for i in range(5):
    for j in range(3):
        print i+j

# print something
print 'hello world'

quit
```

Example with file edits

Kryten uses diff comparison to update files on the system. Use `@@UPDATE@@ path.to/file` to start a file edit block. Write how the file should look post-edit and end with `@@END@@`.

`file_edit.play`

```
# create a file

@@UPDATE@@ file_edit.txt
a sample presentation
@@END@@

# update the file

@@UPDATE@@ file_edit.txt
a sample presentation
with another line
@@END@@

quit
```

Playing a presentation

To load a presentation, use the -p[lay] argument to pass in a .play file.

```
kryten | ⇒ ./kryten.py -p plays/python.play

##    ## ####### ##    ## ####### ####### ##    ##
##  ##  ##   ## ##  ## ##    ##   ##     ###   ##
## ##   ##   ## ####   ##    ##   ##     #### ##
####    ####### ##     ##    #####   ## ## ##
## ##   ##   ## ##     ##    ##     ##   ###
##  ##  ##   ## ##     ##    ##     ##    ###
##    ## ## ##    ##    ##    ####### ##    ##
> That's easy for you to say, Mr David, you're a human

....0> # a simple loop
....1> for i in range(5):
....1>     print i

     | 0
     | 1
     | 2
     | 3
     | 4
....2> # another loop
....3> for i in range(5):
....3>     for j in range(3):
....3>         print i+j

     | 0
     | 1
     | 2
     | 1
     | 2
     | 3
     | 2
     | 3
     | 4
     | 3
     | 4
     | 5
     | 4
     | 5
     | 6

....4> # print something
....5> print 'hello world'
     | hello world

....6> quit
kryten | ⇒
```

Commands

Commands in shell:

SPACE execute and move to next line

q quit

p previous

n next without execue

x quickly execute and move to next line

Commands in editor:
SPACE next line
UP, DOWN for moving highlited line
q quit
b previous
n next
s save partial
x quickly execute end exit
(i intrecative mode - not yet suported sorry)

Advanced arguments

In addition to the standard -p[lay] argument, there a few more advanced arguemnts to be used during the initialization of a presentation.

-n[onblock]
-d[elay]
-i[nput]
-o[utput]
-s[peak]
-d[ebug]
-m[arkmin]

Notes

Comments that are too long to print given the active terminal width will be skipped.

License