Palpatine Documentation

Release 0.1.0

Tzu-ping Chung

March 01, 2015

Contents

Contents:

CHAPTER 1

Palpatine

Manipulate your console screen like an emperor.

- Free software: BSD license
- Documentation: https://palpatine.readthedocs.org.

1.1 Features

- TODO

CHAPTER 2

Installation

At the command line:

```
$ pip install palpatine
```

CHAPTER 3

Usage

To use Palpatine in a project:

```
import palpatine
palpatine.init()
```

Clear the screen:

```
palpatine.screen.clear()
```

Move the cursor around and do things:

```
palpatine.cursor.move(5, 3)
print('Hello world!')
print('A second line.')
```

Move the cursor relatively:

```
palpatine.cursor.up(bol=True)
```

Clear a line:

```
palpatine.line.clear()
```

CHAPTER 4

palpatine package

4.1 Submodules

4.2 palpatine.const module

This module defines several constants to use with functions that require options to use.

`palpatine.const.`**`AFTER = 0`**
Used with clear functions.

If supplied, the clear function will clear everything *after* the cursor.

Seealso `screen.clear()`, `line.clear()`

`palpatine.const.`**`ALL = 2`**
Used with clear functions.

If supplied, the clear function will clear the entire context (e.g. either the current line, or the whole screen).

Seealso `screen.clear()`, `line.clear()`

`palpatine.const.`**`BEFORE = 1`**
Used with clear functions.

If supplied, the clear function will clear everything *before* the cursor.

Seealso `screen.clear()`, `line.clear()`

4.3 palpatine.cursor module

`palpatine.cursor.`**`down`**(*y=1*, *bol=False*)
Move the cursor down y lines.

If `bol` is set to `True`, the cursor will be put to the beginning of line. Otherwise it will maintain its current x position.

`palpatine.cursor.`**`hide`**()
Hide the cursor from screen.

The cursor is still available, just not drawn.

Seealso `show()`, `set_visible()`

palpatine.cursor.**hori**(*value*, *rel=True*)
Convinience function to move the cursor horizontally.

If `value` is positive, the cursor will be moved to the right; negative values move the cursor to the left. If `value` is `0`, this is a no-op. If `rel` is set to `False`, this function moves the cursor to the absolute x position of the current line.

palpatine.cursor.**left**(*x=1*)
Move the cursor x columns to the left.

palpatine.cursor.**move**(*x*, *y*)
Move the cursor the absolute position (x, y) on screen.

palpatine.cursor.**reset**()
Move the cursor to the top-left position on screen.

palpatine.cursor.**right**(*x=1*)
Move the cursor x columns to the right.

palpatine.cursor.**set_visible**(*visible*)
Set the cursor's visibility on screen.

The cursor is always available, even if it is not visible.

Seealso `show()`, `hide()`

palpatine.cursor.**show**()
Show the cursor from screen.

Seealso `hide()`, `set_visible()`

palpatine.cursor.**up**(*y=1*, *bol=False*)
Move the cursor up y lines.

If `bol` is set to `True`, the cursor will be put to the beginning of line. Otherwise it will maintain its current x position.

palpatine.cursor.**vert**(*value*)
Convinience function to move the cursor vertically.

If `value` is positive, the cursor will be moved down; negative values move the cursor up. If `value` is `0`, this is a no-op.

4.4 palpatine.line module

palpatine.line.**clear**(*clear_type=2*)
Clear part of, or all characters in the current line.

The cursor is *not* moved after the line is cleared. You will need to call cursor-moving functions (see `cursor`) to move it manually.

Parameters **clear_type** – What part of the current line should be cleared. See `const` for a list of possible choices.

4.5 palpatine.screen module

palpatine.screen.**clear**(*clear_type=2*)
Clear a part, or all of the screen.

The cursor is *not* moved after the screen is cleared. You will need to call cursor-moving functions (see `cursor`) to move it manually.

Parameters **clear_type** – What part of the screen should be cleared. See `const` for a list of possible choices.

`palpatine.screen.`**`scroll`**`(`*value*`)`
Scroll the screen.

Scroll the screen by `value` lines. If `value` is positive, the screen if scrolled down; negative values scroll the screen up. If `value` is `0`, this is a no-op.

4.6 palpatine.utils module

`palpatine.utils.`**`init`**`()`
Initialize Palpatine.

Call this at the start of your program to initialize Palpatine. Some monkey-patching will happen on certain systems (e.g. Windows) to make Palpatine work.

Seealso `deinit()`.

`palpatine.utils.`**`deinit`**`()`
De-initialization Palpatine.

Call this to disable Palpatine. This does not always disable Palpatine completely, but only restores some monkey-patched elements. Use it only if you run into problems with those monkey-patched things.

Seealso `init()`, `reinit()`.

`palpatine.utils.`**`out`**`(`**params*, ***options*`)`
Outputs a ASCII escape command to console.

Palpatine works by sending ASCII escape sequences to your terminal. If you want to send your own sequences, maybe to achieve some advanced operations not covered by Palpatine, this is the function for you.

Seealso https://en.wikipedia.org/wiki/ANSI_escape_code

4.7 Module contents

CHAPTER 5

Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

5.1 Types of Contributions

5.1.1 Report Bugs

Report bugs at https://github.com/uranusjr/palpatine/issues.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

5.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with "bug" is open to whoever wants to implement it.

5.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with "feature" is open to whoever wants to implement it.

5.1.4 Write Documentation

Palpatine could always use more documentation, whether as part of the official Palpatine docs, in docstrings, or even on the web in blog posts, articles, and such.

5.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/uranusjr/palpatine/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

5.2 Get Started!

Ready to contribute? Here's how to set up *palpatine* for local development.

1. Fork the *palpatine* repo on GitHub.
2. Clone your fork locally:

   ```
   $ git clone git@github.com:your_name_here/palpatine.git
   ```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

   ```
   $ mkvirtualenv palpatine
   $ cd palpatine/
   $ python setup.py develop
   ```

4. Create a branch for local development:

   ```
   $ git checkout -b name-of-your-bugfix-or-feature
   ```

   Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

   ```
   $ flake8 palpatine tests
   $ python setup.py test
   $ tox
   ```

   To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

   ```
   $ git add .
   $ git commit -m "Your detailed description of your changes."
   $ git push origin name-of-your-bugfix-or-feature
   ```

7. Submit a pull request through the GitHub website.

5.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.6, 2.7, 3.3, and 3.4, and for PyPy. Check https://travis-ci.org/uranusjr/palpatine/pull_requests and make sure that the tests pass for all supported Python versions.

CHAPTER 6

Credits

6.1 Contributors

- Tzu-ping Chung <uranusjr@gmail.com>

CHAPTER 7

History

CHAPTER 8

0.1.0 (2015-03-01)

- First release on PyPI.

CHAPTER 9

Indices and tables

- *genindex*
- *modindex*
- *search*

p