

# Using CriticMarkup with pandoc

Kolen Cheung

Using CriticMarkup with pandoc—not a filter but a preprocessor.

## Caveats

The way this script works depends on the fact that pandoc allows RAW HTML and RAW LaTeX in the markdown source. The CriticMarkup is transformed into either a RAW HTML or RAW LaTeX representations (specified by the command line arguments).

Because of the asymmetry in the way pandoc handle RAW HTML and RAW LaTeX (namely, markdown inside RAW HTML are parsed, but not in RAW LaTeX), markdown within the CriticMarkup will not be rendered in LaTeX output. This filter might help if you want to change that: LaTeX Argument Parser.

Another caveat is that nesting CriticMarkup might have unexpected behavior, especially in LaTeX output. For example, the test.md file do not have a valid LaTeX output because of nesting CriticMarkup.

Lastly, see the caveats section in the spec of CriticMarkup.

## Definition of CriticMarkup

- Deletions: This is ~~is~~-a test.
- Additions: This is a test.
- Substitutions: This ~~isn't~~is a test.
- Highlighting: This is a **test**.
- Comments: This is a test.

What is a test  
for?

## The Scripts

These scripts are supposed to be in the same folder:

- `criticmarkup-accept.py`
- `criticmarkup-reject.py`
- `pandoc-criticmarkup.sh`

The `criticmarkup-accept.py` and `criticmarkup-reject.py` are extracted from the OS X System Services from CriticMarkup Toolkit.

## Note on LaTeX Output: Usepackage Required

Note that the latex output requires the LaTeX packages `color` and `soul`. As you can see from this markdown file, I used a hack to guarantee your pandoc standard template also use them, like this: `fontfamily: lmodern,color,soul`.

In my personal use, I have an YAML of `usepackage: [color,soul]` that in my template will added `\usepackage{color,soul}`. See `ickc/pandoc-templates` at `latex-usepackage-hyperref` for details.

## Usage

`pandoc-criticmarkup.sh [options...] [file]`

Options:

- `accept: -a`
- `reject: -r`
- `permanent: -p`
- `show diff: -d`
  - `-d html`: targeting html output using RAW HTML
  - `-d latex`: targeting LaTeX output using RAW LaTeX
  - `-d pdf`: same as above

If `permanent` is used, it will overwrite the original, if not, it will output to `stdout`. In most situation `permanent` should be used with `-a` or `-r` only, but it can be used with `-d` as well.

`-a`, `-r`, `-d` are supposed to use separately:

- If `-d` is used, the others are ignored,
- if `-r` is used, `-a` is ignored

It can be used with the pandoc commands, like these (see `build.sh` for more examples):

```
## Showing Difference and not overwriting
./pandoc-criticmarkup.sh -d html README.md | pandoc -s -o README.html
./pandoc-criticmarkup.sh -d pdf README.md | pandoc -s -o README.pdf
## accept or reject while overwriting the source
```

```
./pandoc-criticmarkup.sh -ap README-accept.md
./pandoc-criticmarkup.sh -rp README-reject.md
```

## Todo

- add a minimal CSS (especially for the HTML aside element)

## Appendix

### CSS

An optional CSS `pandoc-criticmarkup.css` make the deletions and additions more obvious in HTML output.

### Mapping for Showing Differences

critic markup	HTML	LaTeX
<code>\st{[text]}</code>	<code>&lt;del&gt;[text]&lt;/del&gt;</code>	<code>\st{[text]}</code>
<code>\underline{[text]}</code>	<code>&lt;ins&gt;[text]&lt;/ins&gt;</code>	<code>\underline{[text]}</code>
<code>\st{[text1]}\underline{[text2]}</code>	<code>&lt;del&gt;[text1]&lt;/del&gt;&lt;ins&gt;[text2]&lt;/ins&gt;</code>	<code>\st{[text1]}\underline{[text2]}</code>
<code>\hl{[text]}</code>	<code>&lt;mark&gt;[text]&lt;/mark&gt;</code>	<code>\hl{[text]}</code>
<code>\marginpar{[text]}</code>	<code>&lt;aside&gt;[text]&lt;/aside&gt;</code>	<code>\marginpar{[text]}</code>

### Test Files From MMD

test.md is from MMD-Test-Suite/Critic.text at master · fletcher/MMD-Test-Suite