📖 tef / **remarkable**

let's just not go here. it is a silly document markup format.

⭐ **26** stars   🍴 **1** fork

⭐ Star	🔔 Notifications

`<> Code`   ⊙ Issues   ⭑↑ Pull requests 1   ▶ Actions   ▥ Projects   📖 Wiki   ⊘ Secur

🎋 master ▾   Go to file
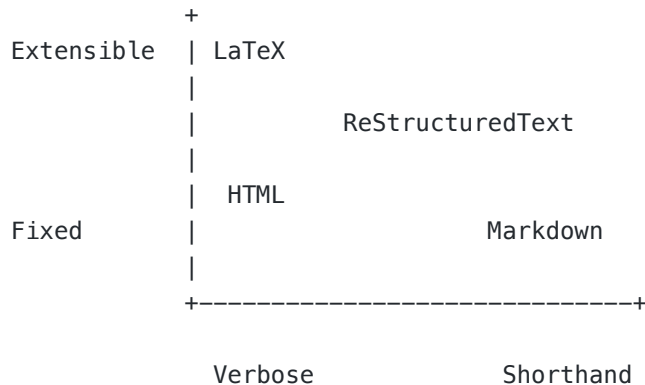
tef Proposal 9   …   on Feb 10, 2016   ⏱ 35

View code

README.txt

```
"Why I hate your document markup format"
---------------------------------------------------------------------------

I have to write worksheets. Friends have to write technical documentation. I
would like a pleasant way to write and manage documents that can be translated
into web ready, ebook, and print formats. Without crying.

And yes, I have seen that XKCD. So let's look at the options:


                      +
    Extensible  | LaTeX
                      |
                      |          ReStructuredText
                      |
                      |   HTML
    Fixed          |                      Markdown
                      |
                      +------------------------------+

             Verbose              Shorthand

    Markdown is great for small comments, blog posts. LaTeX is good for writing
    a book on Mathematics. RST is good for python documentation, and HTML is good
    for webpages.

    Shorthand is good for blog posts, and making markup "look just like text", but
    it comes at a price.
```

 – Some require lining up characters and precision formatting. i.e
   exact underlines. This turns into drawing ASCII art.

 – Nesting becomes awful. To put lists within lists often involves
   extra formatting or indentation to get it to work.

 – Implicit markup leads to Accidental commands. Sentences which start with
   a letter or number. Including a document can change header heirarchy.
   It becomes harder to see what is being parsed, and much harder to
   just get it to work.

 – Semantic whitespace can be useful, but exact numbers of whitespace,
   or trailing whitespace are a pain in the butt.

 – New commands all look different, and it can be hard to recall each
   special parsing case, or introduce new forms, without creating
   incompatible variants.

What I really desire is something with a handful of shorthand for very common
cases, and a uniform syntax for commands, rather than different syntax
for inline, block, and other document markup.

Existing Solutions

Markdown:

        – Considerable amounts of ASCII art.
        – Everyone does it differently.
        – It's a HTML superset. Generating non HTML is painful.
        – Whitespace sensitive to the point of tears. Indents must be 4 spaces
          in some variants. Line continuations are 2 trailing spaces.
        – Zero tests or specs or grammars.
        – Great for blog comments. Awful for books.

LaTeX:

        – Turing Complete. Syntax is mostly Quirks.
        – How do I escape symbols? OH GOD HELP ME.
        – Weirdo error messages. You can't do that in horizontal mode. Overful
          hbox.
        – Hard to recall edge cases of syntax.
        – Won't do EPUB, HTML, pleasantly. Tied to print output.
        – Possible to extend, but only in weirdo arcane macro language.
        – It's not a document format bob, it's a turing complete
          macro system. Did you just tell me to go fuck myself?
        – Unicode.

HTML:

        – Clunky at times, some formatting built in.
        – Won't generate references, indices, tables of contents.
        – Manual Demarcation of Paragraphs, Zero Shorthand.

```
ReStructuredText:

        - Some ASCII art (header titles, etc)
        - Headers are defined implicitly, so including a document
          can change what level headers are.
        - Extensible blocks but hard to add inline markup, without using
          substitutions, or roles.
        - Table syntax, Header names involves shunting characters around
        - Reliance on shorthand makes it hard to remember all the options and
          rules.

Scribble:

        http://docs.racket-lang.org/scribble/
        - Tied to racket.



Winner?

        ReStructuredText is the closest to what I desire, but the reliance
        on ascii markup causes headaches in longer documents.

        Although some shorthand is useful, being able to unambiguously
        markup text is useful too.

        What I desire is something similar in scope to ReStructuredText
        but with a standard mechanism for markup directives.



Things I will have to do before this is close to usable
-------------------------------------------------------------------------------

[ ] Define core grammar
[ ] Write parser, tests
[ ] Define core commands for markup
[ ] Define shorthand
[ ] Build HTML, PDF, epub output
[ ] Pandoc support
```

Releases

No releases published

Packages

No packages published