CONTENTS 1

What remains to be designed after deciding on the tools?

What remains to be designed? The description with in the abstract will be followed up with the specifics.

August 4, 2023

Contents

content as markdown text

content as markdown text

Pandoc allows a metadata block in YAML before the text written in markdown[1].

The design must fix the information to include in the metadata block and decide on the markdown extension to be included.

A decision on the editor is not required; it is recommended that the editor used should include spell checking tools for multiple languages. I think that spell checking is part of editing the source and not part of the baking of a site.

I wish tools to systematically process text from input on an US-keyboard where input of accented characters and similar (umlaut, "ñ", "¿" etc.) is difficult.

It should be possible to collected several shorter markdown texts and produce a booklet.

[1] Pandoc markdown has many extensions. There is an effort underway to specify markdown more precisely as `commonmark`. The extensions must include at least - footnotes - references to be included from bibtex - images - hyperrefs - table of content

Build Site Generator around Pandoc

Build Site Generator around Pandoc

Pandoc translates many different text file formats into `html`, it can handle BibTex references in text and produce publication lists (with pandoc-citeproc now with citeproc.

It works well with doctemplates, which is a small template system[2]

[2] similar to `moustache`

with just conditionals and loops. It is produced by the same author as `pandoc`.

Pandoc works with conversion of files into value. Files can include metadata as YAML blocks in the text source.

It is possible to produce `pdf` files from the blog entries, which gives a printable format (better than printing the `html` document).

Pandoc uses internally JSON, which in Haskell means using aeson.

Directory structure

Directory structure

The design fixes the file structure: theme and content (dough) is separated from the baked site. The source for the web pages(`dough`), the layout and appearances (`theme`) are stored in a directory. The produced web pages go into a different directory (`baked` site).

The resulting `html` files served are stored elsewhere. (Storage of Site Data)

Description of site source layout in a text file

Description of site source layout in a text file

The layout of the source directories and the target directory can be set out in a YAML file (`settings.yml`). In the same file, other general parameters of the web site can be included as well.

Principle: only one settings file in structured text format (YAML).