Semantic Markup with the M4 macro processor

rohieb

33c3, 27.–30.12.2016

M4

- Kernighan & Ritchie, 1977
- Part of almost every Unix/Linux distribution
- Heavily used in GNU `autotools`
- Simple mechanism:
  - define macros
  - when macro token is found in input, replace token with definition
  - write input to output

Example: Hello(?) World

```
$ cat 01-helloworld.m4
define(HELLO, Ohai)dnl
HELLO world.

$ m4 01-helloworld.m4
Ohai world.
```

Notes:

- `dnl`: Discard to Next Line
  - "comments"
  - prevents unwanted whitespace in output

Parametric Macros

```
$ cat 02-params.m4
define(`HEADING', `<h1 id="$1">$2</h1>')dnl
define(`PARA',    `<p id="$1">$2</p>')dnl
define(`ITALIC',  `<i>$1</i>')dnl
dnl
HEADING(title, `This is the title')
PARA(first_para, This is rendered as text. Parameters can even span
multiple lines. If you include commas`,' be sure to include quotes.)
PARA(second_para, This is normal text. ITALIC(This is italic.))

$ m4 02-params.m4
<h1 id="title">This is the title</h1>
<p id="first_para">This is rendered as text. Parameters can even span
multiple lines. If you include commas, be sure to include quotes.</p>
<p id="second_para">This is normal text. <i>This is italic.</i></p>
```

Semantic Markdown

- Write documents in Markdown, using a “public macro interface” for e.g.:
  - votes
  - resolutions
  - TODOs
- Write an M4 library for each output format
  - define the “public macro interface” accordingly
- Use the markup with multiple M4 “libraries” to convert your minutes to:
  - human-readable formats like Markdown
    - with consistent formatting
    - via pandoc: HTML, ReST, LaTeX, PDF, ...
  - machine-readable formats like JSON, CSV, XML, ...

See: `https://github.com/rohieb/minutes-m4rkup`

example.m4rkdown

```
MEETING_MINUTES(Board meeting, 2016-09-01, 20:00,
    [Adam Ant, Henrietta Horse, Beatrice Bull], [Ephraim Elephant], [Adam Ant])

We talked about things for a while.

* RESOLUTION(Resolution 2016-00, VOTE_ADOPTED(3,0,0), World Domination)
* TODO(T2016-03, Adam Ant, Build Doomsday Machine)
* Recent TODOs:
    * TODO(T2016-02, , Try out LAZZZORs from Evil Corp., [
        * They make nice, cheap LAZZZORS
        * LAZZZORS are needed for Doomsday Machine])
    * DONE(T2016-01, Beatrice Bull, Buy some space rockets)
END(21:00)
```

example.markdown

```
Board meeting 2016-09-01
========================

* Starting time: 20:00
* Attending: Adam Ant, Henrietta Horse, Beatrice Bull
* Absent: Ephraim Elephant
* Minute taker: Adam Ant

We talked about things for a while.

* __Resolution 2016-00  (3|0|0):__ World Domination
* __TODO Adam Ant:__ Build Doomsday Machine
* Recent TODOs:
    * __TODO:__ Try out LAZZZORs from Evil Corp..
        * They make nice, cheap LAZZZORS
        * LAZZZORS are needed for Doomsday Machine
    * ~~__DONE Beatrice Bull:__ Buy some space rockets~~
* Ending at 21:00
```

example.json

```
{ "metadata": { "type": "Board meeting","date": "2016-09-01", "start_time": "20:00",
    "attendants": "Adam Ant, Henrietta Horse, Beatrice Bull",
    "absentees": "Ephraim Elephant", "keeper_of_minutes": "Adam Ant" },
  "content": [
    { "type": "resolution", "public": true, "ref": "Resolution 2016-00",
      "vote": { "approved": true, "pro": 3, "contra": 0, "abstain": 0 },
      "text": "World Domination", "allocated_money": "", "notes": "" },
    { "type": "todo", "public": true, "done": false, "ref": "T2016-03",
      "assigned": "Adam Ant", "text": "Build Doomsday Machine",
      "notes": "" },
    { "type": "todo", "public": true, "done": false, "ref": "T2016-02",
      "assigned": "", "text": "Try out LAZZZORs from Evil Corp.", "notes":"
  * They make nice, cheap LAZZZORS
  * LAZZZORS are needed for Doomsday Machine" },
    { "type": "todo", "public": true, "done": true, "ref": "T2016-01",
      "assigned": "Beatrice Bull", "text": "Buy some space rockets", "notes": "" }
  ],
  "end_time": "21:00"
}
```