RSSKISS: A RSS feed reader in 100 lines of code

rsskiss is a quick-and-dirty replacement for Google Reader.

READ THE HYPERTEXT VERSION HERE:
http://www.pabr.org/rsskiss/rsskiss.en.html

Revision History		
1.0	2014-04-12	Initial release.

Table of Contents

1. Motivations

I welcomed Google's termination of its Reader service in July 2013 as an opportunity to curb my addiction to online news. Unsurprisingly, after a brief period of improved productivity, I found myself in search of a new RSS reader. These were the alternatives:

- **Cloud-based services .** Several providers have jumped in to fill the void left by Google, but I did not want to deal with end-of-support issues ever again. If Google cannot provide a long-lived feed aggregation service for free, then who can ?
- **Browser add-ons .** Since I use completely different browsers on many devices, synchronization would have been a major problem.
- **Self-hosted server .** There are plenty of open-source RSS reader projects with a web interface, but it looks like they all require Java/PHP/Perl and a full-featured transactional database, a.k.a. bloatware and dependency hell.

Eventually I wrote a quick-and-dirty, command-line-friendly feed parser which addresses 80 % of my requirements with only 100 lines of shell script and XSL rules.

2. Requirements

- A generic UNIX/Linux server
- **xsltproc [http://xmlsoft.org/XSLT/xsltproc.html]** (from libxslt). Regular expressions would work reasonably well, but now that XML has infiltrated everything, **xsltproc** comes for free, so why not use it ?
- **wget**.
- Some way to export HTML output to multiple devices: a self-hosted web server (preferred), or a web hosting account, or simply sendmail.
- A HTML viewer.

3. How it works

- Feed subscriptions are stored in a plain text table (example [rsskiss-1.0/test.org]). Note the emacs Org-mode-friendly format.

  The last column specifies how each feed should be displayed:

  - **rawhtml**: Copy HTML from feed entries (**dangerous**).
  - **noscript**: Mangle <script> tags.
  - **notags**: Hide all HTML tags.
  - (default): Display the title only.
- rss2table.xsl [rsskiss-1.0/rss2table.xsl] and **xsltproc** transform XML-formatted RSS and Atom feeds into plain text with one item per line.
- rsskiss_poll.sh [rsskiss-1.0/rsskiss_poll.sh] reads the subscriptions file, retrieves feeds, and formats unread items into a HTML document on standard output.
- (optional) rsskiss_poll.cgi [rsskiss-1.0/rsskiss_poll.cgi] is a simple wrapper for **rsskiss_poll.sh**. It can run either as a **cron** job or as a CGI script to refresh feeds manually.

- (optional) rsskiss_mark_all_read.cgi [rsskiss-1.0/rsskiss_mark_all_read.cgi] lets me mark articles as read from within a web browser. Alternatively I could trigger it with an email reply, or with **cron**, or on the command-line.

4. Quick start

```
wget http://www.pabr.org/rsskiss/rsskiss-1.0/rsskiss_poll.sh [rsskiss-1.0/rsskiss_poll.s
wget http://www.pabr.org/rsskiss/rsskiss-1.0/rss2table.xsl [rsskiss-1.0/rss2table.xsl]
wget http://www.pabr.org/rsskiss/rsskiss-1.0/test.org [rsskiss-1.0/test.org]
sh ./rsskiss_poll.sh test.org  > new.html
firefox new.html
```

5. Warnings

- **rss2table.xsl** has been tested with a few dozen feeds from various content-management systems, but it is not guaranteed to work with every variant of RSS and Atom.
- Displaying third-party HTML code from a trusted HTTP server may be a security concern, depending on how the server and the browser are configured to handle security domains. **rsskiss** does try to sanitize untrusted data, but it probably does not catch every HTML escaping trick.