

Org HTML Export

Leon Rische

[2019-12-27 Fri 16:33]

Contents

1	Date Metadata Advice	1
2	Publishing Configuration	2
3	SVG Images	2

This website is generated by exporting orgmode files to html. A small python script then adds headers, footers and a backlink section to each page.

Install `htmlize` for export to html and require / configure a few packages.

```
language=Lisp,label= ,caption= ,captionpos=b,numbers=none (use-package
htmlize)
```

```
(require 'ox-publish) (require 'ox-html) (setq org-export-htmlize-output-
type 'css) (setq org-html-htmlize-output-type 'css)
```

```
(setq org-html-doctype "html5") (setq blog-base-path "/home/leon/org/website/")
```

1 Date Metadata Advice

The HTML exporter adds `<meta>` tags for the file's author and keyword. Add an advice around it to also include the date.

```
language=Lisp,label= ,caption= ,captionpos=b,numbers=none (defad-
vice org-html-build-meta-info (after my-html-build-meta-info) (let* ((info
(ad-get-arg 0)) (date (plist-get info :date)) (protect-string (lambda (str)
(replace-regexp-in-string "\"" "quot;" (org-html-encode-plain-text str)))))) (pp
date) (if date (setq ad-return-value (concat ad-return-value (org-html-close-
tag "meta" (format "name=datecontent=(funcall protect-string (if (stringp
(car date)) (car date) (org-element-property :raw-value (car date)))))) info
"))))))
(ad-activate 'org-html-build-meta-info)
```

2 Publishing Configuration

Using the inheritance pattern from the org-publish html tutorial.

```
language=Lisp,label=,caption=,captionpos=b,numbers=none (setq org-
publish-project-alist '(("website-static" :base-directory " /org/static/" :re-
cursive t :base-extension "css
|js
|woff" :publishing-directory " /org/websitehtml/static/" :publishing-functionorg-
publish-attachment)("website-images" :base-directory " /org/website/" :
recursivet :base-extension"png
|svg
|jpg
|jpeg
|gif" :publishing-directory" /org/websitehtml/" :publishing-functionorg-
publish-attachment)("website-files" :base-directory" /org/website/files/" :
recursivet :base-extension"dsk" :publishing-directory" /org/websitehtml/files/" :
publishing-functionorg-publish-attachment)("website-org" :base-
directory" /org/website/" :base-extension"org" :publishing-directory" /org/websitehtml/" :
recursivet;; :html-headnil :publishing-functionorg-html-publish-
to-html :headline-levels4 :html-extension"html" :html-preamblenil :
html-postamblenil
:section-numbers nil :with-toc nil :with-title nil :with-date t
;; Sitemap generation ;; disable sitemap for now ;; :auto-sitemap t ;;
:sitemap-filename "index.org" ;; :sitemap-title "Index" ;; :sitemap-sort-files
anti-chronologically ;; :sitemap-style list :makeindex t) ("website-org-txt"
:base-directory " /org/website/pages/" :base-extension "org" :publishing-
directory " /org/websitehtml/pages/" :publishing-functionorg-ascii-
publish-to-ascii :headline-levels4 :section-numberst :with-
toct)("website-org-pdf" :base-directory" /org/website/pages/" :base-
extension"org" :publishing-directory" /org/websitehtml/pages/" :publishing-
functionorg-latex-publish-to-pdf :headline-levels4 :section-
numberst :with-toct)("website" :components("website-images""website-
files""website-org""website-org-txt""website-org-pdf""website-
static"))))
```

3 SVG Images

By default, the org html export wraps .svg images in a <object> tag.

I just redefine the function to use tags for all images.

```
language=Lisp,label= ,caption= ,captionpos=b,numbers=none (defun
org-html-format-image (source attributes info) "Return img tag with given
SOURCE and ATTRIBUTES. SOURCE is a string specifying the location
of the image. ATTRIBUTES is a plist, as returned by 'org-export-read-
attribute'. INFO is a plist used as a communication channel." (org-html-
close-tag "img" (org-html-make-attribute-string (org-combine-plists (list :src
source :alt (if (string-match-p "!txpng/" source)(org-html-encode-plain-
text(org-find-text-property-in-string'org-latex-srcsource))(file-
name - nondirectorysource)))attributes))info))
```