# This is the data for swisspedhealth

[swisspedhealth.ch](https://swisspedhealth.ch) hosted via [swisspedhealth.github.io](https://swisspedhealth.github.io)

Some installs may be required to serve this site locally for testing; e.g. at least requires: “gem install jekyll-scholar”

The repo is transformed by [Jekyll](http://github.com/mojombo/jekyll) into a static site and then pushed to GitHub. Jekyll serve runs this site locally for testing and building before pushing (jek.sh and rake commit\_deploy). This site is built from writings in Markdown and rendered as a minimalist source for the swisspedhealth project, without popups, ads, or heavy content.

The data is stored on GitHub and in private backups so that content is not lost and can be pushed from many locations.

Bibliography is set to update from my cloud master bib via shell script “update\_bib.sh”. The scholar plugin is used by: plugins/ext.rb (require ‘jekyll/scholar’). With this plugin, no information is required for the biblio in individual file headers. Within text citations are used with: {% cite name %}. The biblio is printed at end of file with: {% bibliography –cited %}.

# Rake deploy pre-build before hosting

Plugins (scholar) cannot be run on GitHub pages, therefore the site is run by using the \_site directory as the root as described by [davidensinger.com](http://davidensinger.com/2013/07/automating-jekyll-deployment-to-github-pages-with-rake/).

To allow this to work, two branches are created: \* *source* \* *master*

Normally, GH pages looks for the *master* branch. It will build the website project from the \_root directory and compile the website itself in safe-mode. It would create its own version of the \_site subdirectory, as is produced when you run Jekyll locally.

Since we run plugins that GH pages does not use (scholar citations), the build would fail. All writing is done from the *source* branch. This contains the complete website data. This branch is compiled by Jekyll locally and the resulting output is in \_site.

Then on the *master* branch, we force the \_site subdirectory to act as the project root. GH pages will host the pre-compiled site for us. If we no longer want this and prefer GH-pages to compile, we can go back to a single branch converting *source* to be *master*.

**Normal version** \* branch *master* - \_root → GH pages build → host site. - \_site → ignored by GH pages.

**Modified version** \* branch *source* - \_root → jekyll local build → \_site. \* branch *master* - \_site → GH pages no build → host site.

**Protocol**

The Rakefile contains the final version of this description. The steps are outlined here for clarity. For the inital set up, we must create the source branch:

git branch -a  
git checkout -b source

The following tasks are then automated by putting them in a rake file. Delete master branch:

git branch -D master

Check out a new master branch:

git checkout -b master

Force the \_site subdirectory to be project root:

git filter-branch --subdirectory-filter \_site/ -f

Checkout the source branch:

git checkout source

Push all branches to origin:

git push --all origin

**Rakefile**

The rake file is used as follows:

List the tasks from the Rakefile

rake -T

Run these individually

rake preview

The normal protocol is to run the tasks in order with one command

rake commit\_deploy

Make sure that sh jek.sh is run so that jekyll compiles the site and populates \_site before commiting and pushing the *master* to the live site.

# License

The following directories and their contents are Copyright Dylan Lawless. You may not reuse anything therein without my permission (although I am unlikely to complain about non-profit usage):

* \_posts/
* \_topics/
* pages/
* images/

All other directories and files are MIT Licensed. Feel free to use the HTML and CSS as you please. If you a copy the jekyll site generator, a link back to http://github.com/mojombo/jekyll would be appreciated by the devolper, but is not required. If you copy my pushlished content, a link back to https://swisspedhealth.com would be appreciated.

For git tracking, test: git config merge.conflictstyle diff3

# Cloning and keys

Since I work with others and use different accounts, machines, emails, here are some notes incase you or I need them.

To push to multiple github accounts with different keys, and different machines, these settings can be used. Instead of a global git config, local configs are used for each repo. Here is the example with two of my repos. The custom usernames for the local repo is shown (but custom email is removed to prevent spam). [Create your ssh keys as per github recommendation](https://docs.github.com/en/github/authenticating-to-github/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent). In the .ssh directory, the config file will assign the key to each git repository that you clone based on the Host that you use. i.e. custom instead of the default:

* git clone git@custom.github.com:accout/repo.git
* git clone git@github.com:accout/repo.git

# Set up the ssh config file  
cd ~/.ssh/config  
  
# set such that Host and User are custom  
# swisspedhealth repo  
Host swisspedhealth.github.com  
 HostName github.com  
 User swisspedhealth  
 PreferredAuthentications publickey  
 IdentityFile ~/.ssh/key1\_rsa  
 IdentitiesOnly yes  
  
# differentwebpage repo  
Host differentwebpage.github.com  
 HostName github.com  
 User someusername  
 PreferredAuthentications publickey  
 IdentityFile ~/.ssh/key2\_rsa  
 IdentitiesOnly yes

Then clone your repo using the custom Host instead of the default provided by github when you use button “clone/ssh/copy”.

# Clone using the correct Host as per config.  
# As shown at the end of this page, you may need to clone with submodules.  
# You can do by add the "--recursive" flag.   
git clone --recursive git@swisspedhealth.github.com:swisspedhealth/swisspedhealth.github.io.git  
  
# Set the local user here (instead of global, i.e. /Users/user/.gitconfig)  
cd "the clone repo dir"  
git config user.email personemail@addess.com  
git config user.name swisspedhealth  
  
# Clone using the correct Host as per config  
git clone git@swisspedhealth.github.com:swisspedhealth/swisspedhealth.github.io.git  
  
cd "the clone repo dir"  
git config user.email someotheremail@address.com  
git config user.name someotherusername

You should now be able to pull and push from that repo without the [“incorrect user” problems](https://stackoverflow.com/questions/4665337/git-pushing-to-remote-github-repository-as-wrong-user).

## Multiple branches

Since the website is built and deploy with source and master, you will need both and switch to source for building.

git pull --all

git branch -a

git checkout source

# Submodules

I have a fork of jekyll-reading-time as a submodule in plugins. If you did not have this submodule added in \_plugins (with reading\_time\_filter.rb), then the options set in \_layouts/topics.html and post.html would cause content to be printed 2 times, with a formatting error due to the jekyll-reading-time command “{{ content | reading\_time }}”. To add a submodule, e.g. git within the git such as in bundles, use

git submodule add https://github.com/…

## Clone submodule default

To clone use a submodule:

git clone --recursive git://github.com/foo/bar.git or if you forgot or cant use recursive, do

git submodule init and git submodule update

## Clone submodule with custom host and user SSH

And if you are using the ssh method shown above you will need to define the path such that git can make the correct request to match your .ssh/config:

To be able to clone the submodule (e.g. \_plugins/jekyll-reading-time), modify the .gitmodules url

From: url = git@github.com:swisspedhealth/jekyll-reading-time.git

To: url = git@swisspedhealth.github.com:swisspedhealth/jekyll-reading-time.git

Set git to use the updated .gitmodules:

git submodule sync

Then update the submodules

git submodule init

git submodule update

# bfg and lfs

We do not aim to have large data files on this website. However, sometimes they are used in projects and might be added to the repo. To prune out large files that are not required in the git history, bfg was used. This is not common, but to keep the repo size small files that are not required in the git history, bfg was used. This is not common but required to keep the repo size small. The jar of bfg was used as follows:

java -jar ~/Downloads/bfg-1.14.0.jar --no-blob-protection --strip-blobs-bigger-than 60M ~/web/swisspedhealth.github.io/

git reflog expire --expire=now --all && git gc --prune=now --aggressive

# rbenv

Running a consistent ruby environment. <https://github.com/rbenv/rbenv> brew install rbenv

After install, init, and adding ~/.bash\_profile variable: export PATH="$HOME/.rbenv/bin:$PATH" eval "$(rbenv init -)"

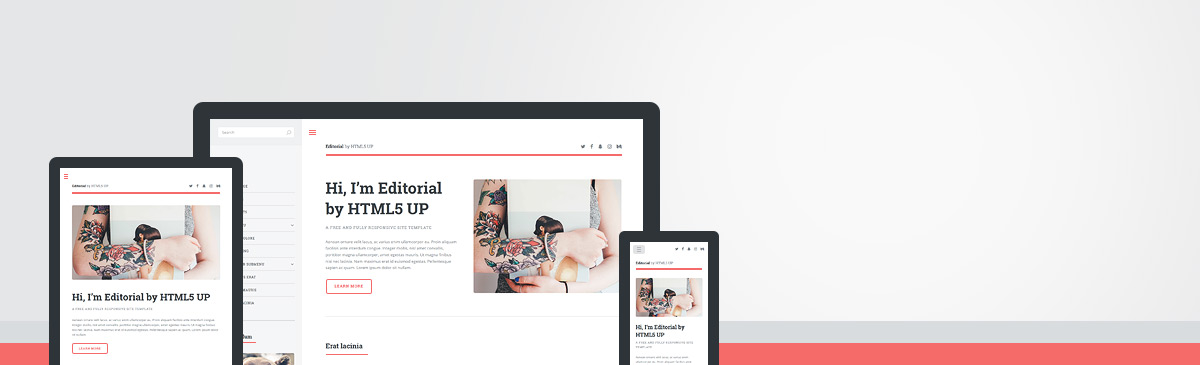
use the version specificied by the gemfile, e.g. rbenv install 2.7.0

It is critical to also set: rbenv local 2.7.0 rbenv global 2.7.0

to allow the final bundle install to work: gem install bundler bundle install

# (WIP) Editorial - Jekyll Theme

A Jekyll version of the “Editorial” theme by [HTML5 UP](https://html5up.net/).



Editorial Theme

## How to Use

For those unfamiliar with how Jekyll works, check out <https://jekyllrb.com/> for all the details, or read up on just the basics of [front matter](https://jekyllrb.com/docs/frontmatter/), [writing posts](https://jekyllrb.com/docs/posts/), and [creating pages](https://jekyllrb.com/docs/pages/).

* **GitLab**: Simply fork this repository and start editing the \_config.yml file!
* **GitHub**: Fork this reposity and create a branch named gh-pages, then start editing the \_config.yml file!

# Added Features

* Add your **social profiles** easily in \_config.yml.

# Configuration

You can use the following custom parameters in \_config.yml.

## Site

* subtitle sets the text for the lighter colored text next to your site’s title.

## Social

* 500px\_url
* facebook\_url
* github\_url
* gitlab\_url
* googleplus\_url
* instagram\_url
* linkedin\_url
* pinterest\_url
* slack\_url
* twitter\_url

## Credits

Original README from HTML5 UP:

Editorial by HTML5 UP  
html5up.net | @ajlkn  
Free for personal and commercial use under the CCA 3.0 license (html5up.net/license)  
  
  
Say hello to Editorial, a blog/magazine-ish template built around a toggleable "locking"  
sidebar (scroll down to see what I mean) and an accordion-style menu. Not the usual landing  
page/portfolio affair you'd expect to see at HTML5 UP, but I figured for my 41st (!!!)  
template I'd change it up a little. Enjoy :)  
  
Demo images\* courtesy of Unsplash, a radtastic collection of CC0 (public domain) images  
you can use for pretty much whatever.  
  
(\* = not included)  
  
AJ  
aj@lkn.io | @ajlkn  
  
  
Credits:  
  
 Demo Images:  
 Unsplash (unsplash.com)  
  
 Icons:  
 Font Awesome (fortawesome.github.com/Font-Awesome)  
  
 Other:  
 jQuery (jquery.com)  
 html5shiv.js (@afarkas @jdalton @jon\_neal @rem)  
 Misc. Sass functions (@HugoGiraudel)  
 Respond.js (j.mp/respondjs)  
 Skel (skel.io)

Repository [Jekyll logo](https://github.com/jekyll/brand) icon licensed under a [Creative Commons Attribution 4.0 International License](http://choosealicense.com/licenses/cc-by-4.0/).