ROOT Web Pages

O.Couet - 2019 January 28

Goal of this presentation

Make the team:

- Agree that moving away from Drupal and have a static web site this is the right thing to do.
- Aware of the tool(s).
- Decide which tool to use.

Why static web site

- Simplicity of maintenance.
- No data base => only text files (markdown, html …).
- Can be maintain in Git like the rest of ROOT (PR etc …).
- Can be generated by jenkins on demand or nightlies.

What we want to keep from Drupal

Based on the installed Drupal modules for the ROOT web site

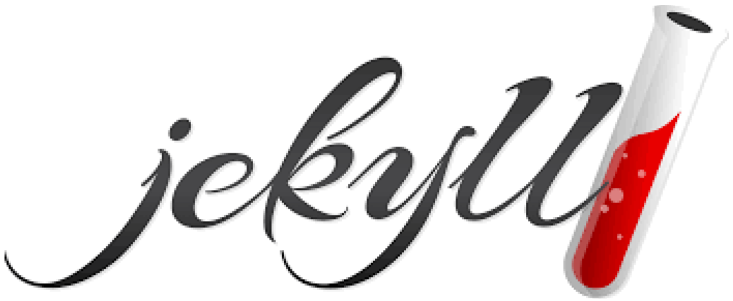
- **Links checker**.
- **Responsive site.** *(The ability to look good on desktop and mobile devices)*
- **Comments section.**

Plan for transition

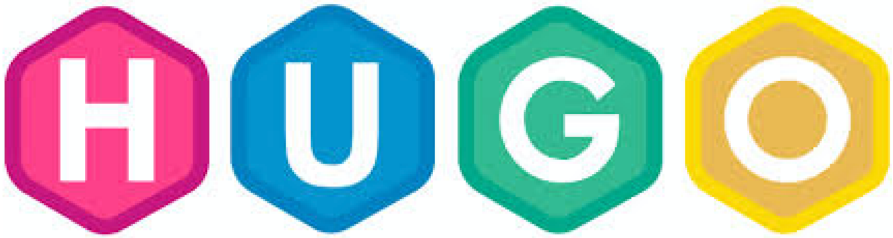
- Choose the tool today
- The Theme: by March we can decide on one.
- Make a prototype with few pages with different themes to compare (March)
- Decide which pages should be kept and which should be archived. Some page will be obviously kept. I will check with the authors for the more specific ones.
- Translate all the most relevant pages (archive the others: pdf backup). A work for all team but I will push you to have something in October to be ready for the November release.

==> Hopefully have something starting to work mid April

Tools



- Two main competitors



HUGO vs jekyll

- There is many comparison on the web between these two tools. Let see the conclusions from: https://forestry.io/blog/hugo-and-jekyll-compared/

jekyll

Pros:

- **Simple templating engine.** Jekyll’s templates will feel very familiar to the syntax found in Wordpress or Craft.
- **Rich theme library.** Jekyll has many ready-to-use themes to get started.
- **Rich plugin library.** Jekyll has dozens of plugins to add the features your site will need.
- **GitHub Pages Integration.** Setting up a site with Jekyll and GitHub pages is a breeze.
- **Very large user community.**

Cons:

- **Slow builds.** Large sites may suffer from longer build times. -> We use nightlies build anyway …. (see the speed comparison slide)
- **Lack of built-in features.** Menus, site-maps, multilingual… However, this can all be supplemented by using third-party Jekyll plugins.

HUGO

Pros:

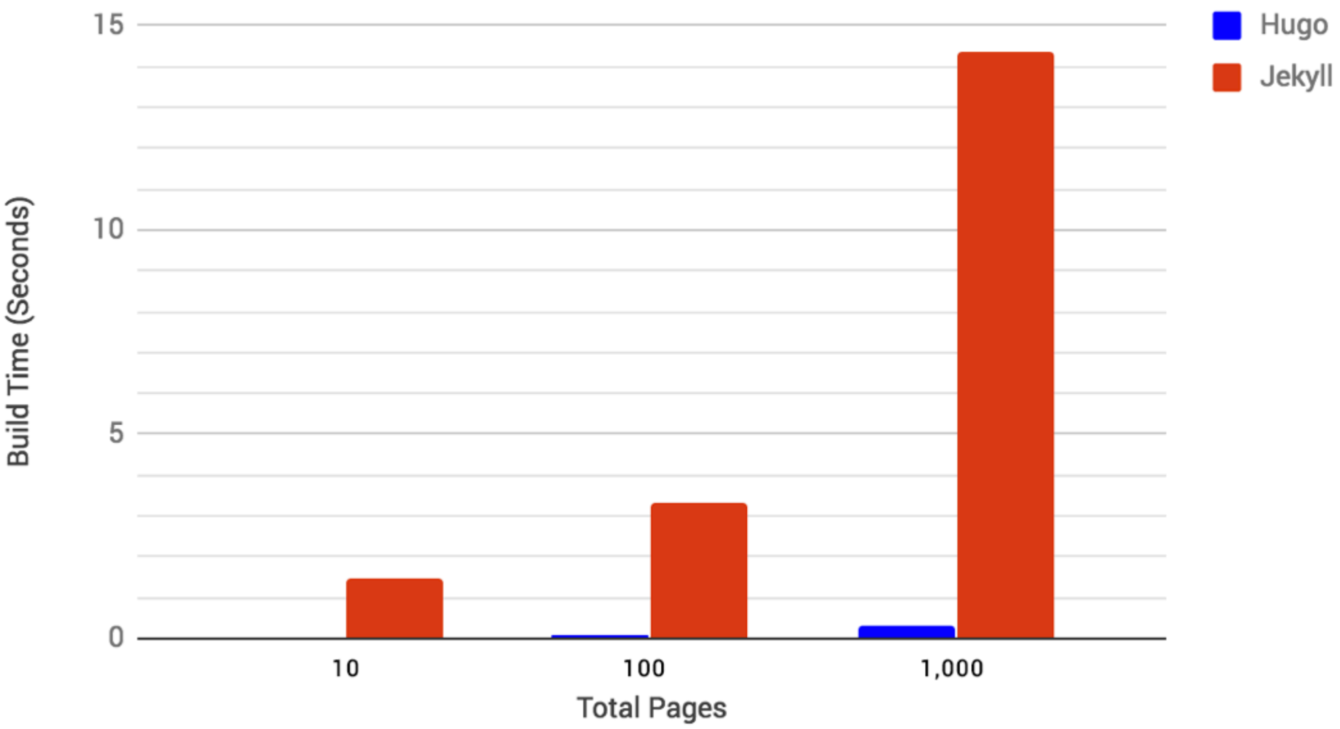
- **Extremely fast.** Build times under 1s. (see the speed comparison slide)
- **Extremely versatile.** Plenty of out the box functionality for enterprise-scale web sites.
- **Enterprise ready.** With support for multiple output types and multilingual sites.
- **Quickly growing community.**

Cons:

- **No extensions.** Hugo doesn't have plugin support, so adding highly custom functionality isn't possible.
- **Confusing template syntax.** While the template engine for Hugo is versatile, it's fairly non-standard and confusing for beginners.

HUGO vs jekyll

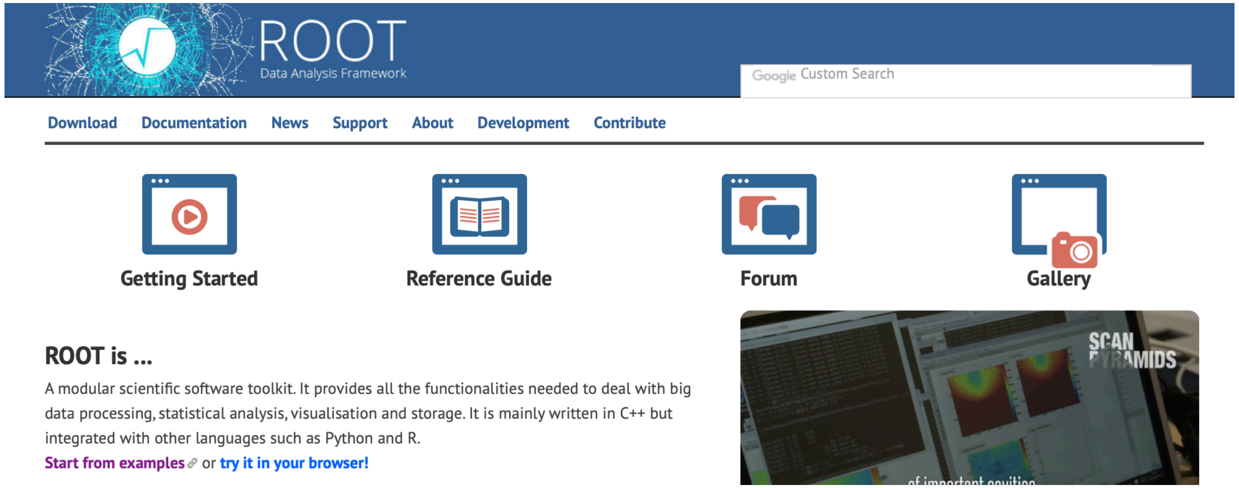
Speed comparison



Jekyll prototype

- A small prototype as been done using Jekyll. It is GitHub format in https://github.com/couet/root-web-site
- GitHub offers to serve this site via: https://couet.github.io/root-web-site/ . Git branch **`gh-pages`**

Pages examples



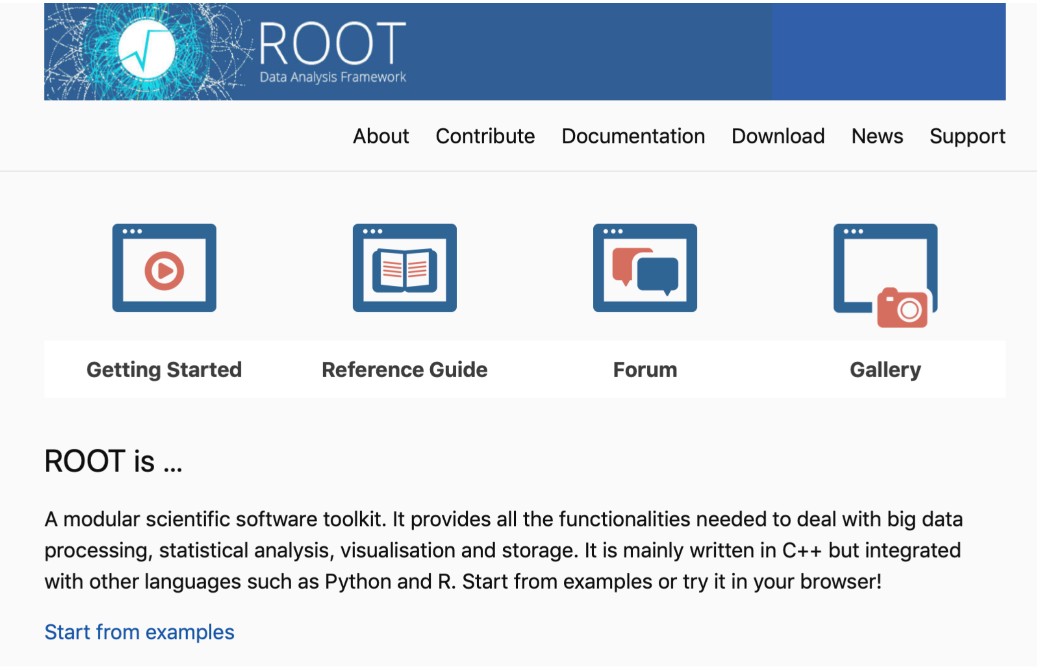
ROOT is ...

A modular scientific software toolkit. It provides all the functionalities needed to deal with big data processing, statistical analysis, visualisation and storage. It is mainly written in C++ but integrated with other languages such as Python and R.

Start from examples or try it in your browser!

Drupal

Jekyll prototype



ROOT is ...

A modular scientific software toolkit. It provides all the functionalities needed to deal with big data processing, statistical analysis, visualisation and storage. It is mainly written in C++ but integrated with other languages such as Python and R. Start from examples or try it in your browser!

Start from examples

What we want to keep from Drupal: The Jekyll way

- **Links checker**. There is Ruby gems doing that. https://github.com/endymion/link-checker
- **Responsive site**. Several Jekyll themes offer that. https://phlow.github.io/feeling-responsive/
- **Comments section**. There is several solutions. One being via Discourse: https://meta.discourse.org/t/use-discourse-as-the-comments-section-of-a-jekyll-site/68310

More on Jekyll …

- Typically you'll use **`jekyll serve`** while developing locally and **`jekyll build`** when you need to generate the site for production.
- Main site:https://jekyllrb.com
- Tutorials: https://jekyllrb.com/tutorials/video-walkthroughs/

Jekyll is perfect for blogging

- Discourse can be use to create comments pages. See: https://meta.discourse.org/t/use-discourse-as-the-comments-section-of-a-jekyll-site/68310

- Here is a nice example: https://pixls.us/blog/2019/01/libre-graphics-meeting-2019/

Web Site workflow

- Edit markdown web page files using your favorite editor.
- Optionally check locally the mods using **`jekyll serve`**
- Commit, push (possibly PR?)
- Jenkins runs on merge, generates web site. *github.io combined with **`gh-pages`** branch allows to see the result also.*
- Jenkins uploads web site to root.cern
- Done!

Choosing a theme

- Can you look for your favourite theme and let me know ? …. we*can* pay a bit for it !
- Many themes here: https://rubygems.org/search?utf8=✓&query=jekyll-theme
- Remember that responsiveness comes with the theme… https://rubygems.org/gems/jekyll-theme-feeling-responsive

Conclusion

- As you may have already guessed I would recommend to use Jekyll to build the ROOT static web site because of its:
    - Very large user community
    - Great Github integration
    - Extensibility - Rich plugins library
    - Many Themes