

How to do some things in [Jekyll](#)

As of Feb 2015 I discovered Jekyll and this has rocked my world. Beautifully done. No more Word-Press for me !!!

Install on OSX

```
sudo gem install jekyll
```

Making posts and pages

To make a blog post, the blog-post.md file needs to start with this yml [front matter](#)

```
---
layout: page
title: "example user file"
category: post
date: 2015-01-30 22:01:06
---
```

To make a new page new-page.md the starting yml front matter looks like

```
---
layout: page
title: "Opening and Closing Igor Pro"
category: gettingstarted
date: 2015-01-30 22:38:12
order: 3
---
```

Making external hypertext links

Reference links look like this

```
This links to [Jekyll front-matter page][1]
This links to [Daring Fireball][2]

[1]: http://jekyllrb.com/docs/frontmatter/ "Jekyll Front-Matter"
[2]: https://daringfireball.net/projects/markdown/basics "Markdown Bascis"
```

Internal Links

I edited [_config.yml](#) like this

```
# Dates are not included in permalinks
#permalink: none
permalink: /:title
```

I do not need to add anything special to my yml frontmatter!

And now this works as link to [stack](#) this works as link to [stack](#)

```
this also works, [Link to stack page][3]
[4]: {{site.baseurl}}/stack/ "stack"
```

this also works, [Link to stack page](#)

On the mac

I am keeping my local Jekyll sites in /Users/cudmore/Sites/Vascular-Analysis

To run Jekyll on a Mac,

```
cd /Users/cudmore/Sites/Vascular-Analysis
jekyll serve --watch
```

View the site locally with

```
http://localhost:4000/Vascular-Analysis/
```

View the site on Github

```
http://cudmore.github.io/Vascular-Analysis/
```

Convert markdown to pdf

```
cd /Users/cudmore/Sites/Vascular-Analysis/_posts
//with TOC
gimli -y -m -s ../css/main.css -w '--toc --footer-right "[page]/[toPage]" -o pdf
//without toc, -w is passing commands to backend wkhtmltopdf
gimli -y -m -s ../css/main.css -w '--footer-right "[page]/[toPage]" -o pdf
```

```
gimli -y -m -s ../css/main.css -w '--footer-right "[page]/[toPage]" -o ../pdf
gimli -y -m -s ../css/main.css -w '--header-left "[webpage]" --header-right "[page]/[toPage]" -o ../pdf
```

```
gimli -y -f ./2015-02-21-simple-stack-db.md -s ../css/main.css -w '--page-size "letter" --footer-right "[page]/[toPage]" -o ../pdf/
```

-y to remove yml

-f to specify a file -s to specify a .css file

Check page load speed

<https://developers.google.com/speed/pagespeed/insights/>

Started using imageoptim-cli

<http://www.smashingmagazine.com/2013/12/17/imageoptim-cli-batch-compression-tool/>

This doesn't work because jpegmini is not configured properly

```
imageoptim --jpeg-mini --image-alpha --quit --no-color --directory /Users/cudmore/Sites/Vascular-Analysis/images
```

This does work

```
imageoptim --image-alpha --quit --no-color --directory /Users/cudmore/Sites/Vascular-Analysis/images
```

Adding a tag cloud

This worked

<http://blog.meinside.pe.kr/Adding-tag-cloud-and-archives-page-to-Jekyll/>

And is inspired on this:

<http://kalapun.com/posts/liquid-tag-management-for-jekyll/>

1. make and fill in tags.html in /Vascular-Analysis/
2. create and fill in _includes/post-tags.html
3. include it in this file _layouts/post.html with

```
{% include post-tags.html %}  
remove the .'s
```

1. edit and add to styles.css
2. include some tags in each post with

```
layout: post  
title: This is an example.  
tags:  
- jeekyll  
- example  
published: true
```

Now get rid of all this in /Vascular-Analysis/, it is redundant with the left TOC.

And add it to /blog/. Then rewrite post-tags.html as i like and include it in a sidebar like i did in post.html