

posting this post

Jan 4, 2016

hi there! i hope this posts finds you well. this is my first of (hopefully) several posts keeping track of what i'm learning. i'm going to keep this particular site code-specific (i.e. you won't see any posts about how to break nail-biting habits or how to resurrect my dead plants). in this post, i'd like to touch on the following:

1. what relevent things/knowledge did i already have before setting out to put up this blog?
2. what new things did i need to install/get acquainted with to get this site up?
3. what are my next steps?

1. what did i already have?

ok. first thing i had before getting started on this project was an interest in getting a blog up to document and motivate my learnings. i've started and abandoned blogs in the past, and knew that i want as much control as possible to help keep me on my toes in terms of learning and intrigue. i didn't really want to fill out a field for each blog post that someone else had made for me, just because that's something that i'd like to know how to do, myself. i was in the middle of putting up a personal site, with a link to "posts," but that link didn't go anywhere. i was a little confused about how i was going to timestamp my posts, whether people would be able to comment on posts, and whether posts would each be an individual file.

some other things i had were:

- a computer and internet browser and internet connection
- an intro understanding of the terminal command line (currently using a mac laptop). the commands i feel cool about are ls, cd, touch, mkdir, cp, mv, open, rm, rm -r, and then... what's it called... how to use executibles (like "vim [open this please]" and "git [do this please]")
- a basic understanding of the vim text editor (like, how to move my cursor around a little, how to edit text, how to search, how to save/quit. lots to learn and keep forgetting how to scroll).
- a github account and a basic understanding of the git executable (am i using that word right? am i spelling it right? i haven't really looked into vim-my spell checks just yet but i'm guessing it's something simple like... ":-spellcheck"). more or less, my git go-to's were (and pretty much still

are) the following:

- `git clone [url]`
- `git add .`
- `git status`
- `git commit -m "this is probably another test"`
- `git push origin master`
- a basic understanding of html
- a basic understanding of css files, scss files, and using sass
- an awareness of the ruby language and some very basic understanding of it

2. what new things did I learn?

oh my gosh, the first thing that pops into my head is that i learned that i can delete repos on github! i somehow didn't realize that each repo had a settings tab. my other learnings include:

- about the existence of [octopress](#), which is "a blogging framework for hackers." oh man! i just went to the octopress site and it looks like the guy who put out octopress, [brandon mathis](#), *just* put up a [post](#) announcing version 3.0, which is going to be a full re-write. for example, I hadn't heard of liquid html before (is that something to do with the I keep seeing?), I'd never knowingly worked with yml before, and I thought that markdown files were only used as readmes.

3. what's next?

- makeover! one of the hurdles in putting up this site is that right off the bat, i don't know how to modify these the files to best fit my needs; I'm not even 100% positive what I'd like my final-er product to look like. i can't get caught up on appearance, though, if i want to get anywhere, so this guy's going up in the lovely simplistic default jekyll layout, with a couple small changes. i'm including a screenshot of what this site looks like right now, in the hopes that one day i'll have it decked out uniquely to me. it should be fun to compare this original to different versions as things move along.
- what is xml? i need to finish reading [this article](#) and explore `/jekyll/feed.xml` a little more. what's populating the fields in this document? who's taking the text i'm inputting here, in `posts/posting-this-post-01-04-2016.markdown`,
- drawing on the above question, can i map out what's happening in the different files? at the very least, i'd like to list them out for myself and remember what lives where.
- ccc

```
def print_hi(name)
  puts "Hi, #{name}"
end
print_hi('Tom')
```



```
#=> prints 'Hi, Tom' to STDOUT.
```



Check out the [Jekyll docs](#) for more info on how to get the most out of Jekyll. File all bugs/feature requests at [Jekyll's GitHub repo](#). If you have questions, you can ask them on [Jekyll Talk](#).

learning stuff

learning stuff
sadiemparker@gmail.com

 [spacerest](#)
 [sad1eparker](#)

Where I'll be documenting my progress in studying web development and learning how to have fun with documents. Current status: still learning.