

Workset #1: Getting online

We're going to start off by getting you a presence on the web from which you will post your blog posts.

Task One: make yourself a webpage.

This week, I want you create a web page using HTML for yourself, with a blog, that we'll use for class communications through the semester.

There are several ways to do this. Choose **one**.

Since we'll be aggregating posts of each others to read, it's vitally important that you use a blogging engine that has an feed of some sort. Both the options I'm going to give here under 2, 3, and 4 should do that automatically. If you wing it, check with me to make sure that I'm able to access your stuff.

Mode 1: Keep using something you already like.

If you already have a website and are willing to use it to post your reading reflections, go ahead and do that. Just make sure it has an atom or rss feed.

Mode 2: Markdown/Jekyll via Github

The cool thing, at least in 2015 or so, was to use a so-called "static site generator." In this scheme, you create your pages in a plain text format, and a program automatically formats them as HTML for you.

Github pages supports templates using something called jekyll which lets you write in "markdown", a language a little more intuitive than HTML and then posts things to the Internet. (This is similar to the course website, although I use something different from jekyll to get from Markdown to HTML.

There are a few ways to do this.

- There's a guide to this [here](#). This will produce a much more modern-looking page for you, at the expense of a little more work. Still, it may be a better web presence.
- There's an *academic-specific* Jekyll template for github at <https://github.com/academicpages/academicpages.github.io>. I've tried to make this even clearer to use at <http://github.com/bmschmidt/bmschmidt.github.io>. But it may still be a little complicated, although you'll end up with something free, modern-looking, and extensible for a few years.
- Here's something worth knowing. If *one* of you gets one of these github templates running nicely with more grad-student appropriate menus, then *anyone else* could copy the same page and fix it to their information. I highly encourage this; the ethics of reusing computer code are somewhat similar to ordinary attribution (i.e, you should attribute where possible), but usually academics of all stripes err on the side of borrowing too little code from others, rather than too much.

Mode 3: Wordpress via Reclaim Hosting.

Finally, you could use one of the more elaborate blogging engines, like Wordpress or Drupal. These tend to be prettier, in terms of existing templates, for all of us who don't do front-end web design; they are also a little more expensive and less sustainable. (For instance, if you use your name as your password on a Wordpress site, you tend to

If you do this, I'm going to suggest that you do it for real; by getting an honest-to-goodness website of your own, with your own domain. I would try using a service called "Reclaim Hosting" for this.

It can host Omeka and Neatline, or Wordpress for a personal site. It will cost \$25 a year, including the registration of your domain name. That's not very much—and I'm not making you buy a lot of books for this class!

The wordpress instructions are good: <https://portal.reclaimhosting.com/knowledgebase.php?action=displayarticle&id=2>. And academics like it.

I have not done this myself, but am happy to help you figure out the various steps on their web site if you hit any snags in doing so.

Mode 4: Punt and use one of the big blogging sites

If this all sounds hopelessly boring and pointless, you can also punt and use an established site like Wordpress.org or blogger.com to just create a simple old blog that isn't really a personal site.

Look into the options above first, though.

Task 2: Do a blog post on your new site!

Post something on your new site, including a brief description of who you are, and **some image from the Digital Public Library of America** that is in some way characteristic of you (with an explanation of how).

Task 3: Minor prep for next week

Install on your laptop a text editor with full *regular expression* capabilities. A text editor looks like a word processor, but focuses less on formatting and more on letting you see digital files as they are.

For Macs, I recommend [TextWrangler](#); for Windows, I recommend [notepad++](#).