

WEB SITE NOTES

Andrew Q. Philips (GitHub section written by Hannah Paul), 8/8/18

Having a personal website is very important. It is how others find you online, see your published work/working papers, find your CV, and access replication files and contact information. There is no need to make one yet, although you probably should have one starting in your second or third year. There are a few different ways to make a website, each with pros and cons:

1. Departmental website: some universities have a departmental website for each faculty member and grad student
 - a. Pros: Not a lot of things to do besides updating your bio and CV
 - b. Cons: Can't take it with you if you go to another university. Websites aren't updated often. Pretty basic.
2. Wordpress/Weebly/Wix/any free website design website:
 - a. Pros: Point and click website, relatively easy to design. Basic ones are free
 - b. Cons: Unclear who owns your website, tracking and cookies.
3. Google sites:
 - a. Pros: Simple, free, clean looking website. Free
 - b. Cons: Lots of file storage uses Google's (e.g., Sheets, Docs...). Design features still limited compared to fully building your website. To the best of my knowledge can't add custom domain name
4. Github: Create a website by hand that you host on github
 - a. Pros: Extremely flexible; you can design the website however you want. You are in control of your data. Easy to make changes to your website from anywhere using Github. Free. Can add things like Javascript. Can add custom domain name.
 - b. Cons: Have to learn CSS/HTML and be comfortable in Github. Takes a lot of work to make a nice-looking website

Ultimately, the way you make a website is up to you. The two best ones are probably Google sites and being hardcore and making your own website that you host on Github.

There are lots of tutorials on creating your own Google site:

<https://gsuite.google.com/learning-center/products/sites/get-started/#/>
<http://depts.washington.edu/trio/trioquest/resources/web/google.php>
<https://libguides.library.kent.edu/smsweb/smsgoogle/sites>

Github is more difficult. If you are interested, below is a step-by-step approach to getting started on your own website to host on Github. Also check out these tutorials:

<https://pages.github.com/>
<https://medium.com/@svinkle/publish-and-share-your-own-website-for-free-with-github-2eff049a1cb5>
<https://codeburst.io/how-to-get-your-website-online-easily-with-github-44ea5ce2997d>

(1) Setting up a website using GitHub

1. Sign up for GitHub
2. Download GitHub Desktop: <https://desktop.github.com/>
3. Select 'create new repository'
4. Name: username.github.io
5. Leave "Initialize this repository with a README" unchecked
6. Git Ignore — None
7. License — None
8. Create Repository
9. Public repository in top right part of menu, uncheck keep code private, publish
10. Go to GitHub web site and select icon, select profile
11. Find web site theme — search "free bootstrap themes" or <https://startbootstrap.com/>
12. Select theme and download, and drag into web site folder
13. Double click on index.html to get live preview of website, refresh to see changes

(2) Editing after initial set up

TIP: every time you work on code, start by “pulling origin” when you open up GitHub Desktop” and then push the changes back to online when you are done so that everything syncs (especially important if you use different computers). When done with edits, go to GitHub Desktop and fill in “Summary (required)” and then hit “Commit to master” and the Push in the top menu

TIP: Keep an old version of everything in a separate file if you are worried about messing stuff up

TIP: Helpful for figuring out css and html code: <https://www.w3schools.com/css/>

TIP: Right click on a web page, select ‘View Page Source’ to see html code

TIP: Start with homepage: index, then can start new html pages like CV, etc.

- > Every web site has an index.html
- > CSS file — gives you the theme for a web site (recommendation: bootstrap template)
- > Bootstrap is nice because it auto formats for mobile
- > If you have a mac, download TextMate and work through that software.
- > Can change the font in the CSS file or the index.html
- > Doing most of edits in the CSS file
- > Do NOT touch the css.min file
- > Comment out original code that you change so that you can go back if you want to
- > To comment in html: `<!-- text -->`
- > To comment in css: use `*` like in STATA
- > Might need to delete the special p class in the index
- > Go to google fonts for fonts, fancier fonts take longer to load: <https://fonts.google.com/?selection.family=Source+Sans+Pro>
- > Don't touch most of the NavBar stuff
- > In the NavBar section, the # is where the button / link takes you
- > Delete sections you don't need, and change NavBar to reference sections that you make
- > Can add a Google Analytics section
- > Safari and Firefox don't cache so can take a while

(3) Miscellaneous

- > Below a: focus... line 278, write h1-h6 with a comma after each, then insert brackets