EEB-5100 Homework due Sep. 6

Paul Lewis

A good goal for the Sep. 6 meeting of the Preparing for an EEB Career (EEB 5100) class would be for everyone to get started on creating a professional web site for themselves. Some of you have already done this, and those that have not should (for this class) create a new GitHub or Weebly site (see instructions below). During class, you will be expected to show us what you have created, even if it is still only a minimal page. Note that, if you already have a web site, *don't* create a new one: just show us your current site next week (but perhaps use this as an opportunity to freshen up your site if it's gone a bit stale). We can troubleshoot problems, trade ideas and advice, and hopefully, by the time we meet on Sep. 6, everyone will either have a site they are happy about or know enough to get there soon.

Feel free to contact me (paul.lewis@uconn.edu) if you have problems along the way: I'm happy to zoom with you or meet in person to help you get unstuck.

Before you start, visit https://eeb.uconn.edu/graduate-students/ and view the web pages other grads have constructed. Here are some web page systems being used by current grads:

Weebly (Apgar, Bizzarri, Clifton, Dalui, Deguire, Lafountain, Lin, Milici, Myers, Terlova)
GitHub (Barreto, Grames, McEvoy, Stukel, Vaziri, Vuruputoor)
WiX (Muscavitch)
Wordpress (Colby)
Squarespace (Lech)

None of the systems above are tied to UConn, which is good! When you leave UConn in a few years, the last thing you want to do is create a new web site, so it is good to use a system that is *portable* (independent of UConn) so that transitioning to a job or a postdoc involves only changing your contact information.

I provide instructions for creating a minimal web site for each of two online web hosting systems below: GitHub and Weebly. I encourage using GitHub for the reasons outlined below, but the most important thing is to get a web site up and running, so, if you are more comfortable with graphical drag-and-drop applications, then feel free to use Weebly for this exercise.

One advantage of GitHub is that, in the process of creating your web page, you will learn how to use a version control system (git), which will be an important skill for your future scientific career. Version control was formerly only used by software developers, but today it is often used to maintain the integrity of any information that can be stored in a text file (manuscripts, datasets, web sites, etc.). Git (https://git-scm.com) is currently the state-of-the-art in version control, and GitHub.com is the place where most git users store their primary repositories. In GitHub, a web site is created by editing text files using a shorthand called markdown that makes it easy to create and, more importantly, easily maintain, a professional web site. When you edit a file (either in a working copy on your local laptop or using the web interface provided by GitHub), you commit the modified files to your GitHub repository, and the changes automatically appear (nicely formatted) on your web site a few seconds later.

Weebly is an alternative way to create a web site. You drag and drop photos or widgets onto a page to build it. With Weebly, you pretty much have to be happy with the tools provided, but it is possible to get a pretty nice looking web site up and running really fast.

Choose either Weebly or GitHub and use the appropriate section below to get started. You can, of course, create web pages under *both* Weeby and GitHub and delete the one you like least afterwards. If you do this, keep both until after our meeting.

GitHub Pages

Steps:

(Note: if you ever find that you want to just start over from scratch, it is possible to delete your GitHub account, or delete just one particular repository, using the Settings link. Just remember there is no undo option when deleting an account or a repository!)

1. Sign up for an account at github.com.

   You'll need to supply a user name, an email address (to maintain portability, use your personal email address, not your UConn email address), and a password. Your ultimate web site will have the web address xxxx.github.io, where xxxx is your user name, so choose the user name carefully!

2. Click the Create Account button.

   Leave everything set to the defaults (check unlimited public repositories for free, and don't check any checkboxes). Press the Continue button.

3. Tailor your experience.

   Provide whatever information you want in this section, then press the Submit button.

4. Click "Start a Project”.

   You'll be asked to confirm your email address before continuing further. This will involve clicking a link in an email that GitHub sends to the email address you supplied when you created your account.

5. Choose a repository name.

   To create a web site, GitHub demands that you use a specific name for your repository: specify xxxx.github.io, where xxxx is *exactly* the same as the username you chose in step 1.

   Keep "Public" checked (checking private will require you to pay money).

   Be sure to check "Initialize this repository with a README".

   You need not add a .gitignore file at this time. It is easy to add one later if you need it.

   Choose a license for your website. You can click the little "i" next to the drop-down list box to read about how to choose an open source license. If in doubt, you can avoid making a choice, but it is probably better to choose any of these options than to leave it set to None.

   Press "Create Repository" to continue.

6. Your README.md already contains some text. Edit it by clicking on it and pressing the button with the pencil icon. Erase everything that’s there and replace it with the text below, substituting your own information everywhere you see REPLACE_WITH_XXXX placeholders.

   ```
   ![Image of REPLACE_WITH_YOUR_NAME](images/headshot.png
   "REPLACE_WITH_SHORT_DESCRIPTION")

   ## About Me
   I am a REPLACE_WITH_MS_OR_PHD student in the UConn EEB department
   interested in REPLACE_WITH_YOUR_PROFESSIONAL_INTERESTS.

   [My CV](PDFs/cv.pdf)

   [Contact Info](contact-info.html)
   ```

   The text above contains links to folders (images, PDFs) and files (headshot.png, cv.pdf, contact-info.html) you have not yet created. Don't worry, we'll fix that momentarily.

   Be sure to save your changes. You do this by typing a brief message (e.g. "added information about me to README file") in the first text box in the "Commit" section, then pressing the "Commit changes" button.

7. Activate your web site.

   Click on the "Settings" link at the top and click on the "Pages" tab on the left. Choose a theme using the button provided. Don't agonize over the theme, it is very easy to switch themes later. I recommend starting with the Primer theme because it provides a simple, clean design.

8. Point your browser to https://xxxx.github.io, replacing xxxx with your username, and view your web page! Note that it may take a minute or two to update. You will have some broken things on your web page - not to worry, we'll fix those things in the next few steps.

9. Adding an image.

   First, create an image of yourself and name the file "headshot.jpg". I recommend resizing the image to 300x300 pixels in whatever software you use for playing with images (e.g. Photoshop, Preview, etc.). While it is not necessary to resize the image, smaller images will load faster for folks visiting your web site. We're going to put this image in a directory named "images". GitHub does not let you create empty directories, so we'll sneakily create a dummy file in the images directory and then later delete the dummy file once we've uploaded the image file.

   Click on the "Add file" button and then the "Create new file" button and enter "images/dummy.txt". Add some dummy text and commit the file. Now enter the images directory and click the "Upload files" button. Find your "headshot.jpg" file and upload it. You'll need to commit the change or else GitHub will ignore the upload.

   You've already added the appropriate line in your README.md file to load this image, so in a minute or two your website should display your headshot image.

10. Adding a link to your CV.

    Create a new dummy file named "PDFs/dummy.txt" in order to create the PDFs directory. Upload a PDF version of your CV (named "cv.pdf") into that directory. Once you have at least one other file in a directory, you can click on "dummy.txt" and then click the trash can button to delete the "dummy.txt" file. Be sure to always commit changes or else they will not take effect.

    Once you've successfully uploaded your CV, edit "README.md" and, if necessary, modify "PDFs/cv.pdf" to reflect the actual name of the PDF file you uploaded if you used a name other than "cv.pdf".

11. Adding a contact info page.

    Your "README.md" has a line that provides a link named "Contact Info" that points to the file "contact-info.html". Let's create that "contact-info.html" file so that this link is no longer broken.

    Click the "Add file" and "Create new file" buttons and type "contact-info.md" into the box. Yes, that's right: type "contact-info.md", **not** "contact-info.html". We are creating a Markdown file that GitHub Pages will automatically convert to an html file. Once the page is created, and before committing it, type in the following content:

    ```
    ---
    title: Contact Info
    layout: default
    ---

    Feel free to contact me at <REPLACE_WITH_YOUR_UCONN_EMAIL_ADDRESS>.

    Postal address:

        REPLACE_WITH_YOUR_FULL_NAME
        Department of Ecology and Evolutionary Biology
        University of Connecticut
        Storrs, CT 06269-3043
        U.S.A.

    [Back to Home](https://xxxx.github.io/)
    ```

    Don't forget to replace xxxx with your user name on that last line, and replace the other REPLACE_WITH_xxxx templates with appropriate text. Be sure to put a blank line between "Postal address:" and the first line of your address. Also, indent each line of your address using exactly 4 spaces (not a tab character). The angled brackets surrounding your email address are also important. That creates a link to your email address that is encrypted so that hackers and spammers cannot glean email addresses from your web page. To read more of what you can do in a Markdown file, see http://assemble.io/docs/Cheatsheet-Markdown.html or similar pages.

That's it! Your minimal web site is now complete and should work properly within a browser within a minute or two. If you want to see the example I created in the process of creating these instructions, check out https://uconneeb.github.io/eebgrad/.

Feel free to read more about GitHub pages at https://pages.github.com and take some courses in generating a GitHub pages web site at https://lab.github.com.

Weebly

Steps:

12. Sign up for an account at weebly.com.

13. Once you have a Weebly account, click the + button to create a new site.

14. When asked "What type of website are you creating?" choose "Personal"

15. Fill in the name of your site (could be your actual name).

16. Select a theme (don't worry about the words or images that are inserted by default, those can be easily replaced with your own)

17. Start editing. Feel free to be bold and daring, deleting and adding pictures and text with abandon! No one will be able to see your site until you click the Publish button.

Tips:

- You do not need to publish your page before class; only publish it if you are satisfied with it and ready for anyone and everyone on the internet to see it.

- Clicking on the Pages tab will show you the separate pages that the theme generated for you automatically - you may want to delete some (and certainly will want to modify all) of these before pushing the Publish button.

- Each page that appears when you select Pages will be a main menu item.

- Click the Build tab from the main menu to work on a page you have selected from the Pages tab.

- Clicking an X will generally delete an element.

- Under the Build tab, there should be a palette on the left of items you can add to your page - just drag and drop (but don't drop too fast; move your mouse around - with the mouse button still held down - to see various options for positioning your item before dropping it).

- To replace an image, click on it to get a dialog box that allows you to choose another image to replace the current one.

- If you want paragraphs of text separated by space, note that you can drag a separate Text widget onto your page (might be more visually appealing than adding blank spaces).

- Visit the support page under the Help tab on the main menu - there are some tutorials for beginners that might be worth watching if you get stuck.

Your web site should contain at least the following things:

- An image (either of you or of a place or an organism central to your research).
- Your contact information (can be just your UConn email address).
- A link to www.eeb.uconn.edu (the Department might as well get a small increase in its Google search status while we are at it).

See http://eeb-grad.weebly.com/ for a silly example of a page I created using only the free features of weebly. This is obviously not the acme of web site design! It is instead designed to show you an example of the *minimum* expected!