Web Development Tutorial for Non-Digital Natives

Sarah Walker and Hannah Pierce-Hoffman

We are seniors at Columbia University studying Computer Science. As the final project for our COMS 6156: Topics in Software Engineering course, we designed a study to evaluate the effectiveness of free online web development resources from W3Schools, Mozilla Developer Network, and Khan Academy. We have created a tutorial made up of lessons that we found to be the most effective in teaching introductory web development to non-digital natives who did not grow up with digital technology. Below is the lesson plan, with our own intermediate commentary to fill in any gaps that we found.

One of the most consistent findings from our study is that subjects found these lessons to be most effective when paired with verbal assistance, either in-person or over the phone, from a technologically savvy helper. Therefore, for maximum positive results from this tutorial, we recommend that non-digital natives follow the tutorial with verbal assistance from a knowledgeable friend or family member. However, this is not required.

Welcome! This tutorial is designed to teach introductory web development. As you follow this tutorial, you will gradually build your own website which you will learn how to publish to the web. Start thinking about a topic for your website. It can be about you, your hobbies, or anything else that you would like to put on the internet.

Preliminary knowledge: *This section contains some background information that will help you conceptually understand how to create a website. It is okay if you do not fully understand this section yet.*

- What exactly is a website?
    - A website is a page, or collection of pages, which contain information. You will be creating some pages of your own using HTML. HTML is a markup language which will allow you to add text, images, tables, lists, and more to your webpage.
- What is hosting?

    - Hosting a web page means making it available for others on the internet to see. We will use GitHub Pages for hosting.
- What is GitHub?
    - GitHub is a website where you can view, share, and store code. It relies on a version control tool called git. You will be using GitHub because it has a feature called GitHub Pages which allows you to host a webpage just by publishing your code to GitHub.
- What is a text editor?
    - A text editor is a program, like Microsoft Word, that allows you to edit and save files containing only text. For this tutorial, we will download and use a text editor called Sublime Text.
- What does “commit” mean?
    - Version control tools like git use “commit” to mean “take a snapshot.” When code or markup has reached a milestone, developers commit it so they can keep track of that version of it. In the future if they make a mistake and need to reverse their changes, they can start with a past commit instead of starting all the way over. In GitHub Pages, you must commit before you can publish your page to the Internet.

Create a website! Follow the steps in order, at your own pace. Before clicking each lesson link, make sure to read all of the additional notes for that step.

1. **Create a GitHub Account** at https://github.com/join.
2. **Create a new GitHub repository** on your GitHub account, using the following instructions as a guide: https://help.github.com/articles/create-a-repo/
    a. IMPORTANT: The repository needs to be named "*your-username*.github.io" exactly, so that we can use it to host a webpage.
    b. For this tutorial, you do *not* need to check the "Initialize this repository with a README" box.
3. **Download and install the free Sublime Text** software at https://www.sublimetext.com/3

   a. Depending on your computer, Sublime Text may ask you to choose a location for the install. Make sure to remember the place you install it, so that you can find the program later.

4. **Download and install GitHub Desktop Client** at https://help.github.com/desktop/guides/getting-started-with-github-desktop/installing-github-desktop/
   GitHub Desktop client allows you to communicate with the GitHub site.
   a. Sign into Desktop Client with your GitHub username.
   b. **Clone** your repository in Desktop Client. Doing this will create a repository folder under Documents → GitHub → *repository name*. Make sure you can locate this folder.

5. **Follow this tutorial to get started with GitHub Pages:** https://pages.github.com/
   GitHub Pages is where your website will be hosted on the internet.
   a. Under "What git client are you using?", click on "GitHub Desktop" to see the appropriate set of instructions.
   b. At the "Create an index file" step, open Sublime Text and create a blank file. Save it under the name index.html. Save this file in your GitHub repository folder.
   c. Use the index.html file throughout the rest of this instruction set.

6. **Learn**: Visit the following page to learn the basic concepts behind HTML: https://www.w3schools.com/html/html_intro.asp

7. **Learn**: Visit the following page to learn how to create and open HTML files: https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Getting_started
   a. Read sections up through "Anatomy of a HTML document." You don't need to do any of the Active Learning exercises in the first few sections.
   b. Do the Active Learning exercise for the "Anatomy of a HTML document" section, using the index.html file you created in your GitHub repository.

8. **Learn**: Visit the following page to learn how to embed images in a webpage: https://www.w3schools.com/html/html_images.asp
   a. Read up through the "Images in another folder" section.
   b. Look at a few of the "Try it yourself" examples for additional guidance.

c. Embed an image of your choice on the index.html page you've created in your GitHub repository. Store the image in the same folder as the repository, by copying the image you want to use to the folder containing your HTML files.

d. When inserting additional lines on your HTML document, make sure to insert them between the <body> and </body> tags, so that the content you add is displayed by the browser.

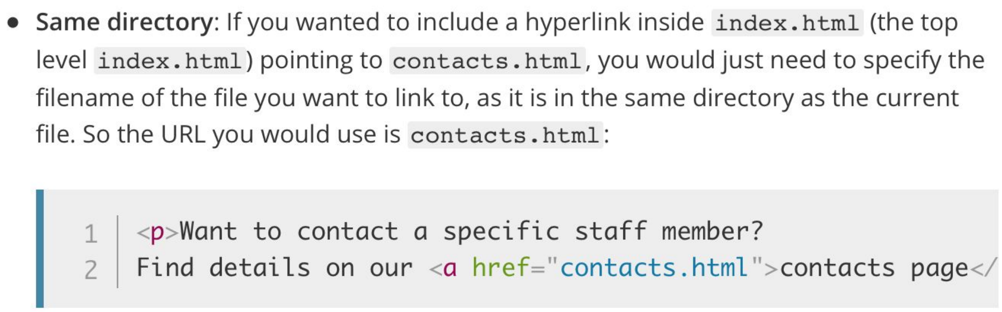
9. **Learn**: Watch the following video to learn about HTML lists: https://www.khanacademy.org/computing/computer-programming/html-css/intro-to-html/p/html-lists

   a. After watching this video, add an ordered or unordered list to your index.html page (you can choose the content of the list).

10. **Learn**: Visit the following page to learn how to link to add links to your website: https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Creating_hyperlinks

    a. Read the “Anatomy of a Link” section

    b. Read the following (which is from the same page, but provided here for easier access):



- **Same directory**: If you wanted to include a hyperlink inside `index.html` (the top level `index.html`) pointing to `contacts.html`, you would just need to specify the filename of the file you want to link to, as it is in the same directory as the current file. So the URL you would use is `contacts.html`:

```
<p>Want to contact a specific staff member?
Find details on our <a href="contacts.html">contacts page</
```

    c. In your GitHub repository folder, create another HTML file called secondpage.html. You can copy the content of the index.html file into this file to create a template.

    d. Add a link to secondpage.html from index.html. Refer to the screenshot above in step 10b. for an example of how to do this.

    e. Add a link to an external webpage on secondpage.html.

    f. Make any other edits you would like to make to secondpage.html to differentiate it from index.html.

11. **Commit your finished website and publish it to GitHub** using Desktop Client. See the GitHub Pages tutorial from step 4 for detailed instructions on how to do this.