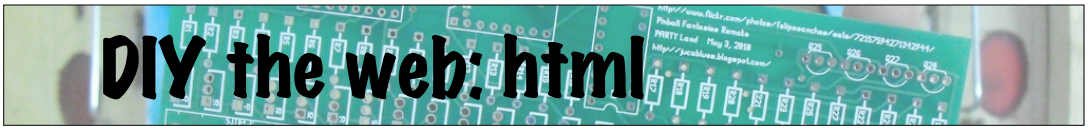

DIY the web: html

Regular people have been building webpages since the early 1990s. At the time, all websites were DIY: you had to know how to code one to make one. The mid-90s saw the introduction of WYSIWYG ("what you see is what you get") editors (like the terrrrrrible Microsoft FrontPage, and Macromedia-then-Adobe's Dreamweaver, both of which were introduced in 1997). And today, of course, there are all kinds of no-coding-needed platforms for building and/or hosting websites — such as Wix and Wordpress. However, all websites are built on html code.

html is the language of the internet. (There are other languages, too, but this is where it all starts.) It stands for "hypertext markup language" and is the name of a set of conventions for marking-up a document so that different computers can open the document and display it with formatting and pictures. html was developed in the early days of computer networking, when computer users were trying to make their documents available to more than one person at a time.

And while building your own web page might sound antiquated (given all the already-built platforms on which many live their digital lives—think Twitter and Facebook and Tumblr and Pinterest, but also LinkedIn and Academia and your institution or organization's website), your other options for being online (even building a site through Wordpress or Weebly) are dictated by others' decisions: their templates and designs and copyright and terms of service.

People make their own websites in order to shape those sites into the kinds of places they want to have online. Some people use their projects for advocacy, resource-pooling, or community-building (think Adbusters, the Organization for Transformative Works, HASTAC, Yarn Bombing Los Angeles, Craftster, and more).

Even though most contemporary websites use other coding languages in addition to html (including css, php, java script, and more), they all use html. And since that's where the web started, that's where we'll start. In its simplest uses — which are what you'll learn here — html allows very simple text formatting: you can change the size, color, and weight of typefaces (but have very limited choice over the typefaces), include pictures in your file, and link your file to other files.

An html file itself (as opposed to a webpage) is a document of text only — and of text that is completely unformatted. The text is "tagged" — marked-up with symbols — that indicate how the text should be displayed when the document is opened within a Web browser (such as Internet Explorer, Firefox, or Safari). When a browser encounters an html document, it doesn't really open the file but instead "reads" it by making a page, on the fly, that duplicates the words from the file, formats them as the markings in the file indicate, and inserts pictures and links where the file indicates to place them.

In the steps below, you'll learn how to:

- Make a very simple html file with words, a picture, and a link.
- Adding text and making basic paragraph divisions.
- Change the background color of the file.
- Change the size, weight, and color of the text in the file.
- Have a picture appear in the file when it is read in a browser.
- Add a link to the file.

what software to use

Because html documents are meant to be read across platforms, the documents can be made in the simplest of software applications. If you are on a Mac, use TextEdit; for Windows, use NotePad.

get started: make and check out the basic document

1) Open TextEdit or NotePad — the application will open with a new, untitled document you can use to make your webpage.

   If you are using TextEdit, do the following two things:

   a) From the Format menu, choose the "Make Plain Text" option.

   b) From the TextEdit menu, choose Preferences. In the dialog box that opens, click the "Open and Save"option, and make sure that the option "Ignore rich text commands in HTML files" is checked and the "Add '.txt' extension to plain text files" is NOT checked.

   Note: If you close the file and re-open it, you will need to check "Ignore rich text commands" at the bot- tom of the Open dialog box.

2) Type the following, exactly, into the blank, untitled document:

```
<html>
<head>
<title>Digital activism</title></head>
<body>
Activate!
</body>
</html>
```

   You need to type the text exactly because computers are stupid. You, a human, would understand if a character were missing or different, and would mentally replace it. Computers cannot do that. They can only take a text exactly as it is, and so will not be able to show your page properly (or, perhaps, not at all) if it is not presented exactly as the computer expects it.

note

The bits of coding within the angle-characters are called tags. <html> is a tag, as is <head>, as is <body>.

Every tag has what is called a close tag, indicated by the / character. An html document will not work if every tag does not have its corresponding close tag: <html> at the beginning of a document must be followed by </html> at the end; <body> must be followed by </body>.

The <html> at the beginning of an html document indicates to a browser (like Internet Explorer, Firefox, or Safari) that all the text that follows — up until the </html> close tag — can be interpreted and built into a webpage.

The <head> tag indicates to a browser that everything between it and the </head> tag is information not shown on the page itself; instead, the head contains information about search terms for the page and other data that we don't need to use, at this basic level — except for the <title> tag.

Any words contained between <title> and </title> will go at the very top of the browser window that contains the webpage, as you will soon see. Any words and other information like pictures or links contained between <body> and </body> are shown on the actual webpage.

3) Save your file: Click the File menu and then Save, and save the file in the place where you usually save files. It is probably also a good idea to make a folder/directory for saving this particular file — and the picture you will place in it — because doing this will help you prepare for making and publishing webpages, as you will learn. Give the file a short name like "test" or "index," and give it the extension .html; the full name of your file, that is, should be something like "test.html" or "index.html." This .html extension is absolutely necessary: a browser will not be able to open and read the file without it.

4) Open the browser you usually use.

5) From the browser's File menu, click Open and then, in the dialog box that opens, navigate to the page you made and open it. You should see a very simple webpage, titled "Digital activism" and having a white background with the word "Activate!" If you do not see this, go back to NotePad or TextEdit and make sure your text looks exactly like the text shown in Step 2 — then save your file again, and try opening it again.

add some color to your file

1) Return to your file in NotePad or TextEdit and edit the document so that the <body> tag now looks exactly like what is shown here:

```
<html>
<head><title>Digital activism</title></head>
<body bgcolor="#ff00ff">
Activate!
</body>
</html>
```

2) Save your file. You need to save your file any time you make changes in order to see those changes in the browser window.

3) Go back to the browser in which you've opened your webpage, and click the refresh button (typically, a circular-ish image with an arrow somewhere near the URL line in your browser). Your webpage should now have a bright fuschia background color. If it does not, check that your html file looks exactly like the words in Step 1.

   What's going on?

   Color in html texts is described in hexadecimal format. This is a format that allows you to use individual letters and numbers to represent more numbers than you can with a plain decimal (or ten character) system; that is, you can write one character in hexadecimal and with it you can represent the numbers from 0 to 15 instead of from 0 to 9. Hexadecimal means "sixteen," and hexadecimal format uses sixteen distinct symbols: 0-9 represent the values zero to nine, while a, b, c, d, e, f (or A through F) represent the values ten to fifteen. So the "ff00ff" of your background color means, first, 15, 15, 0, 0, 15, 15. 0 is the minimum, 15 (or f) is the maximum, and 7 is right in the middle.

   Ok, so how does all that make color?

   Color on computer screens is described as RGB: red, green, blue. Red, green, and blue are the basic colors of the light used on computer screens to build all the colors you see. 100% red + 100% green + 100% blue = white; 0% red + 0% green + 0% blue = black; 100% red + 100% green + 0% blue = yellow... and everything in between.

When you use hexadecimal coding to specify the color in an html file, the first two letters or numbers represent the red, the second two represent the green, and the third two represent the blue:

R	G	B
ff	00	77

Given what we explained above about hexadecimal numbers, what you can see is going on here is that ff equals the maximum value for red (or 100%), and 00 is the minimum value for green (or 0%), and 77 is in between for blue (or 50%) — so the color that this combination will make is a mix of full red and half blue, or a nice pink (not as saturated as fuschia!). (Note: the # is necessary because it tells a browser that the information that follows is a color.)

4) Experiment with the RGB color combinations to see what the different combinations get you. Remember that, after any change you make to your html file, you need to save the file and then refresh your browser to see the change in the webpage.

add more text to your file, use the and <br> tags

1) In your html file, type several paragraphs. You can write anything you like. Put paragraph returns (by pressing "return" on your keyboard) into the text.

2) Save the file and refresh your browser to see your changes. You'll note that none of the paragraph returns show up in the text: it will look like one big block of text. This is because, for webpages, you need to be explicit about paragraph formatting.

3) In the html file, put a tag before any text you want to be a separate paragraph, and a closing tag after it. Remember to put a tag for every tag you enter.

4) Save your file and refresh your browser. Now you should see your paragraph breaks.

5) Back in your html file, replace one or two of your tags with a
 (for "break") tag. When you put in a <br> tag, remove the tags for any tags you replaced. The <br> tag is a special tag, one of the very few that does not need a close tag.

6) Save your file and refresh your browser. Notice the differences between what a <br> tag and a tag do in a text.

format text: bold, change size, add color

1) In your html file, put a line of text at the top of your file, text that could serve as a title, and put a tag — for "bold" — before it and a closing tag after it. Also include a .

Your file should now look something like this (the fourth line is what matters here):

```
<html>
<head><title>Digital activism</title></head>
<body bgcolor="#ff00ff">
<p><b>Activate!</b></p>
<p>This is my webpage.</p>
<p>It is a great webpage.</p>
<p>It might even take down the patriarchy!</p>
</body>
</html>
```

2) Save your file and refresh your browser. The title should be bolded at the top of the page. If it is not, check that you added the closing tag, and that the brackets are in the right place, as shown in the example.

3) To change the size of your title, make the line of the title look like this:
```
<p><font size = “3”><b> Digital activism </b></font></p>
```

4) Save your file and refresh your browser. Your title should be bold and large. Experiment with other numbers for the font size to see how the font size property works.

5) To add color to your type, change the line of code for the title to read:
```
<p><font size = "3" color = "#00ff77"><b> Digital activism </b></font></p>
```

6) Save your file and refresh your browser.

7) Experiment with different color combinations (just as you did for the background color) to find a combination you like.

8) Note that, when you want to change just the color of a word or phrase, and not the size at the same time, your tags would look like this:
```
<p><font color = “#00ff77”><b> Digital activism </b></font></p>
```

add a picture

1) Find a digital picture you like, as a jpg file type. (It can be anything, for our purposes. I'd recommend getting familiar with Creative Commons on flickr.com as a place to search for images you can legally use, although images of your own are of course a good option.)

2) Copy the picture into the same folder or directory as your html file.

3) In your html file, decide where you would like the picture to appear, and add this line of code: <img src = "picturename.jpg"> (Be sure that, instead of "picturename" you type exactly the name of your file, including its exact capitalization.)

4) Save your file and refresh your browser. The picture will not appear in the file if you have not gotten its name exact, or if the picture is not in the same folder/directory as the html file. If you do not like how the picture appears, you can put a <p> or <br> tag before the picture to separate it out. You can also move the line of code to any other place in the code where you would like it. (If you want to resize the picture, it is best to open the picture in Photoshop or Gimp to resize.)

add a link

1) Find a webpage you would like to have linked to your page, and copy its URL.

2) Around the words you would like to be the link to this other webpage, add the following, pasting the URL you copied in between the quotation marks:
```
Check out <a href = "http://confusedcatsagainstfeminism.tumblr.com/">Confused Cats Against
Feminism</a>!
```

3) Save your file and refresh your browser. The words around which you put the <a> tags should be blue and underlined, to indicate that they are a link. Click the link. If it doesn't work, or does not show

up, double check that you have typed the and tags exactly, and that the URL is exact. You can include as many words or phrases in your links as you want.

putting your web pages onto a network so others can see them

Many universities have web hosting space for students and faculty, so it's worth checking into that. (And if you don't have it automatically, you might be able to get it pretty easily.) If yours does not, you can purchase hosting space through vendors like GoDaddy, Arvixe, or Bluehost, usually for a few dollars a month. If you wanted to DIY hosting, you certainly can — but you'll need to look beyond me for advice.

want to learn more?

What I've shown you here is just the beginnings of working with html. If you want to learn more, the following URLs are helpful:

- http://www.w3schools.com/HTML/default.asp
- http://www.echoecho.com/html.htm

On the W3Schools website, you'll notice that html5 is prominent. This is (as of October 2014) the current standard for html markup. Following html5 conventions will ensure that the greatest number of browsers will be able to read your files correctly.

While you've been working in TextEdit or NotePad today, there is software that can make html and other coding languages easier to read. Free options include Komodo Edit and TextWrangler for Mac and Notepad++ for Windows. There is also software available — like Dreamweaver, for example — that can do much of the work of writing html for you. But it is important to know this basic coding, because you will sometimes (or, sometimes, often) need to be able to read the code and make corrections. (And there are good reasons to keep your work "lo-fi," as Karl Stolley has argued in his "Lo-Fi Manifesto" <http://kairos.technorhetoric.net/12.3/topoi/stolley/>.)

If you want to practice *reading* html, I'd suggest opening a browser, finding a webpage that looks interesting, and viewing the source code. To do this in Internet Explorer, right-click the background or text of a webpage or frame and then select View Source Code. In Firefox, you can do that by clicking Tools, then Web Developer, then Page Source. In, you first have to go into the Safari menu, select Preferences, and then Advanced. Check "Show develop menu in menu bar" and close preferences. Finally, click Develop and then Show Page Source.

Note that html provides the *structure* of a web page, but another language — css, or cascading style sheets — now usually provides the *layout* for web pages. css makes it relatively easy to assign consistent styles across a series of pages that go together, such as on a website. For example, the page background color you specified in your html file for this introduction would likely be assigned in the style sheet if you were building a website. But never fear: in addition to html, W3Schools also covers css (specifically, css3, the current standard for that language).

acknowledgements

Thanks to Anne Wysocki for leading an html workshop at UW-Milwaukee back in September of 2009; my own approach to teaching html — and this handout itself — are largely adapted from her work.

Additional thanks go to Flickr.com user Felipe Sanches for making the image "lamp controler boards printed by BatchPCB" available to others by using an Attribution-ShareAlike 2.0 Generic (CC BY-SA 2.0) Creative Commons license <https://creativecommons.org/licenses/by-sa/2.0/>. I cropped the full-size image for use behind the title on page 1.