

HTML Me Something, Part I: HTML

◀ **Back to Overview page (..)**

In Part 1, you will get comfortable with writing markup, and with separating content from design and layout.

Getting Started

Stub out `index.html` with these basic elements:

- the HTML5 doctype
- `<html>`
- `<head>`
- `<title>`
- `<body>`

...and fill each element with some appropriate content. You can start by removing the snippet of HTML that you added during setup.

Getting to Work

Your mission is to build a page that:

- Tells a story. This can be personal or impersonal, funny, serious or neither. You can do whatever you like, but generally, it should be something in the range of 3-10 paragraphs or sections. **Here is an example** (<http://education.launchcode.org/html-me-something/submissions/chrisbay/index-nocss.html>), and here are some other ideas:
 - Create a resume page that tells the story of your professional journey to-date, and where you want to go as a coder.
 - Tell the story of a trip that you took.
 - Talk about some hobby or passion of yours.
- Does each of the following:
 - Uses each of the following structural HTML5 tags: `<p>` , `<header>` , `<footer>` , `<main>` , `<article>` . If you need to brush up on what each of these tags is used for, or if any of them are new to you, check out the **HTML tag reference** (<http://www.w3schools.com/tags/default.asp>) at w3schools.
 - Uses at least one `` tag (and hopefully more). When placing images in your page, put them in a new subfolder called `images` within your `html-me-something` directory.

- Uses at least one **HTML entity** (http://www.w3schools.com/html/html_entities.asp). Hint: putting a copyright notice in your footer will afford you the opportunity to use `©` , but you should also try to get creative here.
- Is creative. Don't stop with these items or tags. Have some ideas for your page, and make it great. And dig into the **w3schools HTML reference** (<http://www.w3schools.com/tags/default.asp>) to learn more about other tags, their usage and attributes!

Notes and Tips

- Use your browser developer tools to look at **the example page** (<http://education.launchcode.org/html-me-something/submissions/chrisbay/index-nocss.html>), or to troubleshoot things that don't look right. You can mimic the document structure of this example, but **do not just copy/paste!** Use the example to learn how your HTML elements might be structured, and structure your own page according to your own content's needs.
- Don't add any CSS yet. Really, we mean it! If you think your page looks boring now, that's okay. We'll get there soon enough.
- As you make changes, you will obviously want to see the results. To do so, simply re-save the file in your text editor, and then, over in your browser window, click Refresh (or use cmd+R on a Mac, ctrl+R on Windows).
- Rely on reference sites linked on this page, and elsewhere online. We haven't taught you every detail about every tag that you may want or need to use, and you're free to use tags that haven't been explicitly introduced in class.

We've given you enough background to get started, and are intentionally leaving some of the learning up to you.

Add and Commit Your Changes on Git

Once you are finished, use Git once again to `add` and `commit` your `index.html` changes.

Why again?

The reason is that, since you have added a bunch of new HTML code, your `index.html` file is now in a very different state from how it looked the first time you committed it. You can see this by checking your status:

```
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")
```

Git is telling us: *I see that you have modified `index.html`. Use `git add` if you want this file's changes to be included in your next commit.*

At the risk of oversimplifying a bit, the Git workflow more or less comes down to this:

- Create some initial stuff
- `init`
- `add` and `commit`
- make some changes
- `add` and `commit`
- make some more changes
- `add` and `commit`
- ... etc

The general rule is that **whenever you make any changes, you must add and commit those changes to Git.**

So let's now go ahead and do that. From within your root `html-me-something/` directory:

```
$ git add index.html
$ git commit -m "Finished work on HTML page"
```

You might be wondering: *How do I know when it's time to pause working and do another commit?*

This is somewhat subjective, and ultimately up to you. The basic guiding principle is that you should pause and commit any time you have reached a natural stopping point and completed a coherent "chunk of work".

Done!

Well done! Time to **dive into some CSS** ([../part2-css/](#)).