

1.1.4 Add a Little HTML

HTML (Hypertext Markup Language) is the underlying structure of any webpage. Every piece of the Run Buddy mock-up—the header, sign-up form, trainer images—will be contained in an HTML element. The `index.html` file that we created in the previous step will hold all of this content and code.

We could open this single file in our code editor (VS Code), but we should start thinking about Run Buddy as a project that will eventually have multiple files and folders. So let's open the entire `run-buddy` folder in VS Code.

You can do this in a number of ways: by dragging the folder into VS Code, using VS Code's "File > Open Folder" menu option, or opening it from the command line.

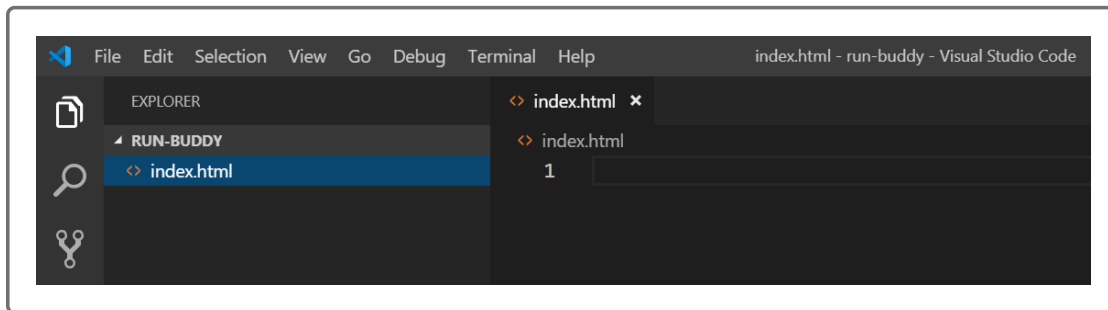
To open VS Code from the command line, in the `run-buddy` directory, run the command `code .` (the `.` means "this directory").

IMPORTANT

macOS users might need to enable the `code` command first. See [this article on about launching VS Code from the command line](#)

(https://code.visualstudio.com/docs/setup/mac#_launching-from-the-command-line).

Click on the `index.html` file in the Explorer on the left to load it in the main panel on the right. You should see something like the following image in VS Code:

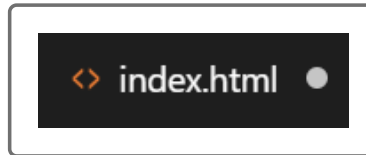


SHOW PRO TIP

Now we're ready to code! Type (don't cut and paste!) the following code in the right-hand panel in VS Code:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>Run Buddy</title>
  </head>
  <body>
    <h1>RUN BUDDY</h1>
    What We Do
  </body>
</html>
```

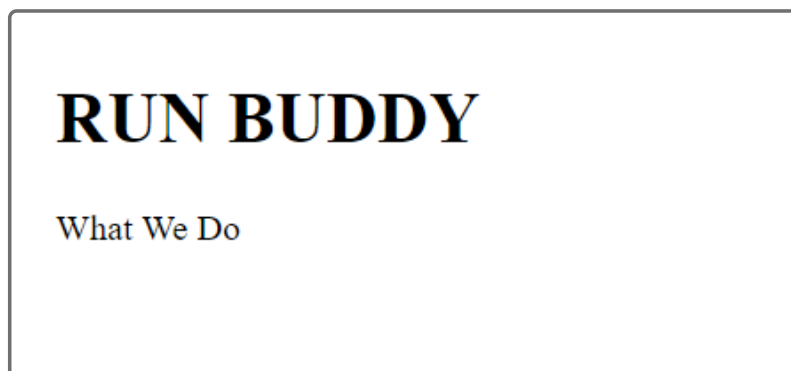
Awesome! You just created your first webpage! We'll go over what this code means later. Before you can view it in a browser, you need to save the file. You'll know the file hasn't been saved if a white dot appears in the tab next to the filename, like this image:



SHOW PRO TIP

Now open this HTML file in Google Chrome. If you have the [Open in Browser VS Code extension](https://marketplace.visualstudio.com/items?itemName=techer.open-in-browser) (<https://marketplace.visualstudio.com/items?itemName=techer.open-in-browser>) installed, you can simply right-click anywhere in the HTML file and select Open in Default Browser.

The webpage should look like this image:



So it's not the most exciting webpage in the world, but you gotta start somewhere!

Introducing HTML Elements

Let's back up and explain in more detail what we just wrote. Notice that a lot of the code was composed of **elements** that are made up of an opening tag (e.g., `<title>`) and a closing tag (e.g., `</title>`). Whatever is in between these tags is affected by what that element signifies.

The `<h1>` Element

The `<h1>` element is a "level 1" heading element, so it makes the text inside it big and bold. If we have other less important headings on the page, we can use the heading elements `<h2>`, `<h3>`, and so on.

DEEP DIVE ▼

Try changing `<h1>RUN BUDDY</h1>` to `<h2>RUN BUDDY</h2>`. Save the file and refresh the tab in the browser. Notice how the text shrunk slightly.

SHOW PRO TIP

Also make a note that this change only affected the "RUN BUDDY" text. Why do you think that is? It's because the element was closed. If you remove the closing `</h2>` tag, the text that follows "RUN BUDDY" would also be big and bold because the browser wouldn't know where the `<h2>` content ends.

Now change it to an `<h3>` and observe again how the text shrinks.

The `<html>`, `<head>`, `<title>`, and `<meta charset>` Elements

With this new understanding of HTML elements, let's look at the others we used in `index.html`. We're writing an HTML page, so it makes sense that

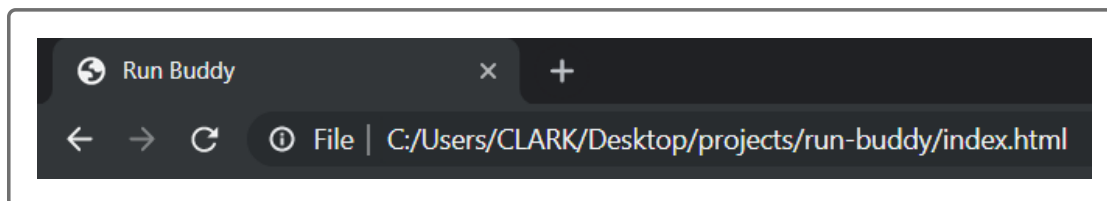
everything would need to be contained in an `<html>` element. The only thing that didn't go inside this element was the `<!DOCTYPE html>`. This is an extra line that tells the browser how to interpret your HTML code.

DEEP DIVE ▼

Inside the `<html>` element, our webpage is further divided into two other elements: `<head>` and `<body>`. The `<body>` is where all of the content should go. Basically, everything in our mock-up will go here. The `<head>`, on the other hand, is where we can provide anything else the browser needs to know about our page.

For example, the browser wants to know what to call this webpage. The `<title>` element inside `<head>` tells it to call it "Run Buddy." Whatever is in the

`<title>` element will become the name of the browser tab or window, as you can see here in this image:



Try changing the title in the HTML file. Save the file and then reload the page in the browser to see the change.

To help reinforce some things we've learned so far, let's take a moment and take a small assessment. Don't worry about how you do on it, the assessments are here to help solidify your knowledge and get you to recall upon things you have learned previously.



Please Wait...

There are two remaining pieces we haven't covered yet. The opening `<html>` tag was actually written as `<html lang="en">`. This is how we specify the default language for our page. In this case, we're letting browsers know that this page is meant to be read in English.

Lastly, there's an element in the `<head>` that looks like this: `<meta charset="UTF-8" />`. This is an example of a **meta tag**, of which there are many. Like `<title>` elements, meta tags give the browser extra information about the page, sometimes for display purposes and sometimes for search engine optimization. We'll discuss these more in future modules.

DEEP DIVE ▼

The `charset` meta tag is important to include because it specifies the range of characters (letters, numbers, symbols, etc.) that can be used. UTF-8 accommodates just about any character, from foreign language symbols to emojis.

Try this out for fun: Copy this emoji (🏃) and paste it somewhere in the `<body>` of `index.html`. Note that it shows up just fine in the browser. But if you change the meta tag to `<meta charset="ASCII" />`, it no longer works. Why do you think that is? It's because the browser then limits the

available characters to those of the ASCII character set, which does not include emojis. Best to stick with UTF-8!

DEEP DIVE ▼

The good news is that much of this is boilerplate, meaning every webpage is going to start with this same structure. Every page needs a DOCTYPE, an `<html>` element, a `<head>` element, etc. It's worth typing out again and again just to reiterate their importance, but it can also be copy/pasted into future projects without harm.

SHOW PRO TIP

© 2020 Trilogy Education Services, a 2U, Inc. brand. All Rights Reserved.