

Lesson

Monday

Introduction to Programming

(/introduction-to-programming)

/ Git, HTML and CSS (/introduction-to-programming/git-html-and-css)

/ HTML Block Elements: <p> Tags, Headings, List Elements, and More

Text

Cheat sheet

We're going to start our HTML writing by working with **block elements**. These are HTML elements that when displayed start on a line of their own, in effect making a block on the page.

HTML Block Elements

Let's create a project directory and initialize a Git repository for a new project called `my-first-webpage`. We'll use this project to practice HTML block Elements.

Create `my-first-webpage`

As a review, here are the steps:

- Open **Terminal**
- Make a project directory called `my-first-webpage` in your *Desktop* directory. (It is important that when you begin a new project you create a new project folder that is *not* nested inside of an existing project folder.)
- Change into your project directory.

- If you type `pwd` now, you should see that you are in your project directory at `/Users/Guest/Desktop/my-first-webpage`.
- Within your project directory, initialize your git repository: `$ git init`
- Create an empty file: `$ touch my-first-webpage.html`.
- Open your project directory in Visual Studio Code: `$ code .` (The command `code .` opens the whole directory you are currently in because the `.` refers to *current directory*. If you only want to open the single file, you could instead type: `code my-first-webpage.html`).

<p> Tags

Once `my-first-webpage.html` is open in your text editor, type the following code (**no more copying and pasting!**):

my-first-webpage.html

```
<p>This is my first web page!</p>
<p>Isn't it nice?</p>
```

The instructions for formatting around the text are called **tags**. The `<p>` is called a **p tag**. It means that everything that follows is a paragraph, until `</p>` is reached. The `</p>` is called a **closing tag** or **end tag** and indicates the paragraph is at its end. The `<p>` is a block element because it is written on its own line and creates a block of content on the displayed page.

One thing to note before we move on — with multiple lines of text, we'll sometimes see `<p>` tags indented like this:

```
<p>
```

```
    This is a lot of text which means it spans multiple line  
s. When text spans multiple lines like this, it's much easi  
er to read our code if we indent the multiple lines of text  
within our paragraph tags. In other words, we add a p tag,  
have a line break and indent, and then add our text.
```

```
</p>
```

Both ways are fine — ultimately, we want our code to be as readable as possible both for ourselves and other developers looking at our code.

Now, if we open the file in a web browser, we have two paragraphs of text. We just wrote our first HTML code! Let's commit it!

```
$ git add my-first-webpage.html  
$ git commit -m "add initial paragraph block elements to th  
e page"
```

Verify that you are on the `main` branch and that the commit has been made by running `$ git status`.

HTML Headings

Let's try some more tags:

my-first-webpage.html

```
<h1>My first web page</h1>  
<h2>Written with the guidance of Epicodus</h2>  
  
<p>This is my first web page!</p>  
<p>Isn't it nice?</p>
```

Here, we've added a main **heading** (<h1>) and a subheading (<h2>). You can actually add up to six different levels of headings using <h1> through <h6> . Like the paragraph tags, we have an opening tag that indicates where the heading begins (<h1>) and a closing tag that indicates where it ends (</h1>) . Closing tags always match the heading but have a forward slash in front.

Notice the blank line between the <h2> and the <p> tags. HTML ignores empty spaces and blank lines, which is called **whitespace**. That's why the extra line we might add between our headings and paragraphs doesn't do anything when it is displayed. We could take it out, or make it three spaces, and it would look exactly the same in the browser.

Let's commit again!

```
$ git add my-first-webpage.html  
$ git commit -m "add heading block elements"
```

List Elements: , , and

Let's keep going with some list elements:

my-first-webpage.html

```
<h1>My first web page</h1>
<h2>Written with the guidance of Epicodus</h2>

<p>This is my first web page!</p>
<p>Isn't it nice?</p>

<p>Here are some things I'm going to learn about coding:</p>
>
<ul>
  <li>HTML</li>
  <li>CSS</li>
  <li>JavaScript</li>
  <li>And a lot more!</li>
</ul>
```

Here we've added an **unordered list** (``) with four **list items** (``). If we wanted to make an **ordered list** that uses numbers and letters instead of bullets, we would use an `` tag instead of a `` .

Notice how I've indented the `` s within the `` s. This makes it much easier to read which tags are nested, and makes it easy to see where the closing tag goes when it's not on the same line. It's a good practice to indent using two spaces (you can use the tab button on your keyboard, just make sure that it's set to two spaces in your VS Code settings). It's also a good idea to write your closing tag just after you write your opening tag, and then go back and fill in whatever comes in the middle. This way, you won't forget to close a tag after you've opened it.

Let's commit again!

```
$ git add my-first-webpage.html
$ git commit -m "add list elements"
```

<!DOCTYPE>, <html>, <head>, and <body> tags

We have some final additions to make for this to be a valid HTML web page:

my-first-webpage.html

```
<!DOCTYPE html>
<html lang="en-US">
<head>
  <title>Web page #1</title>
</head>
<body>
  <h1>My first web page</h1>
  <h2>Written with the guidance of Epicodus</h2>

  <p>This is my first web page!</p>
  <p>Isn't it nice?</p>

  <p>Here are some things I'm going to learn about coding:
</p>
  <ul>
    <li>HTML</li>
    <li>CSS</li>
    <li>JavaScript</li>
    <li>And a lot more!</li>
  </ul>
</body>
</html>
```

The <!DOCTYPE html> tag tells the browser that this document contains HTML, and specifically that it contains the newest version of HTML, HTML5. (An example of a doctype for an older version of HTML is <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">.)

The `<html>` tag wraps all of the content on the entire page. It is sometimes called the root element, because all other tags (excluding the `<!DOCTYPE>` declaration) are nested inside of it.

The `<head>` tag contains information about the page, which for now is just the `<title>` tag. The `<title>` tag sets the title for the web page — if you look in your browser, you can see that the title of the browser tab is now *Web page #1!*. We'll learn about other tags that live in the `<head>` tag soon. For now, keep in mind that the information in the `<head>` tags does not get displayed directly on the webpage.

Finally, the actual content of the page is now wrapped in a `<body>` tag. `<body>` contains all of the content that will actually be displayed on the page.

Now go ahead and commit your final changes:

```
$ git add my-first-webpage.html
$ git commit -m "add final, official HTML elements DOCTYPE,
head, title, body"
```

Finish Up: Push Your Project to GitHub

We're done for the moment. It's a good time to push the project to GitHub. As a reminder, here are the steps (which should be done by both you and your pair, since you each want a copy of the repository on your GitHub):

- Create `my-first-webpage` repository on GitHub
- `$ git remote add [remote nickname] [remote url]` (replacing `[remote nickname]` with your initials or another good nickname and `[remote url]` with the URL to the GitHub repository you just created)
- `$ git push [remote nickname] main` (replacing `[remote nickname]` with the same nickname you used when you added

the remote link)

- Check on GitHub that your code uploaded successfully

Previous (/introduction-to-programming/git-html-and-css/practice-indentation-and-spacing)

Next (/introduction-to-programming/git-html-and-css/html-block-elements-part-2)

Lesson 16 of 64

Last updated March 24, 2023

disable dark mode



© 2023 Epicodus (http://www.epicodus.com/), Inc.