Structure

**What is HTML?**

HTML is the language used to create the web pages you visit everyday. It provides a logical way to structure content for web pages.

Let's analyze the acronym "HTML", as it contains a lot of useful information. HTML stands for **H**yper**T**ext **M**arkup **L**anguage.

A markup language is a computer language that defines the structure and presentation of raw text. Markup languages work by surrounding raw text with information the computer can interpret, "marking it up" for processing.

HyperText is text displayed on a computer or device that provides access to other text through links, also known as “hyperlinks”. In fact, you probably clicked on many, many hyperlinks on your path to this Codecademy course!

In this course, you'll learn how to use the fundamentals of HTML to structure, present, and link content. You'll also learn how to use CSS, or Cascading Style Sheets, to style the HTML content you add to web pages.

Let's get started!

**!DOCTYPE**

A [web browser](https://en.wikipedia.org/wiki/Web_browser) must know what language a document is written in before they can process the contents of the document.

You can let web browsers know that you are using the HTML language by starting your HTML document with a *document type declaration*.

The declaration is the following:

<!DOCTYPE html>

This declaration is an instruction. It tells the browser what type of document to expect, along with what version of HTML is being used in the document. <!DOCTYPE html>*must* be the first line of code in all of your HTML documents.

**Note:** If you don't use the doctype declaration, your HTML code will likely still work, however, it's risky. Right now, the browser will correctly assume that you are using HTML5, as HTML5 is the current standard. In the future, however, a new standard will override HTML5. Future browsers may assume you're using a different, newer standard, in which case your document will be interpreted incorrectly. To make sure your document is forever interpreted correctly, always include <!DOCTYPE html> at the very beginning of your HTML documents.

# Preparing for HTML

Great! Browsers that read your code will know to expect HTML when they attempt to read your file.

The <!DOCTYPE html> declaration is only the beginning, however. It only tells the browser that you plan on using HTML in the document, it doesn't actually add any HTML structure or content.

To begin adding HTML structure and content, we must first add opening and closing <html> tags, like so:

<!DOCTYPE html>

<html>

</html>

Anything between <html> and </html> will be considered HTML code. Without these tags, it's possible that browsers could incorrectly interpret your HTML code and present HTML content in unexpected ways.

# HTML Anatomy

Before we move forward, let's standardize some vocabulary we'll use as you learn HTML. Here are some of the terms you'll see used in this course:

1. Angle brackets - In HTML, the characters < and > are known as angle brackets.
2. HTML element (or simply, element) - HTML code that lives inside of angle brackets.
3. Opening tag - the first, or opening, HTML tag used to start an HTML element.
4. Closing tag - the second, or closing, HTML tag used to end an HTML element. Closing tags have a forward slash (/) inside of them.

With the exception of a few HTML elements, most elements consist of an opening and closing tag.

In the following example, there is one paragraph element, made up of one opening tag (<p>) and one closing tag (</p>):

<p>Hello there!</p>

# The Head

So far, you've declared to the browser the type of document it can expect (an HTML document) and added the HTML element (<html>) that will contain the rest of your code. Let's also give the browser some information about the page. We can do this by adding a <head> element.

The <head> element will contain information about the page that isn't displayed directly on the actual web page (you'll see an example in the next exercise).

<!DOCTYPE html>

<html>

<head>

</head>

</html>

# Page Titles

What kind of information about the web page can the <head> element contain?

Well, if you look at the top of your browser (or at one of the tabs you may have open in this browser window), you'll notice the words Learn HTML & CSS: Part I | Codecademy, which is the title of this web page.

The browser displays the title of the page because the title can be specified directly inside of the <head> element, by using a <title> element.

<!DOCTYPE html>

<html>

<head>

<title>My Coding Journal</title>

</head>

</html>

If we were to open a file containing the HTML code in the example above, the browser would display the words My Coding Journal in the title bar (or in the tab's title).

# Where Does the Title Appear?

The web page you are currently viewing has a title of Learn HTML & CSS: Part I | Codecademy.

Because the browser used in this course does not have a title bar, you won't be able to see the title of a web page if you add one. The diagram to the right, however, illustrates specifically where a web page's title would appear on a browser.

# The Body

We've added some HTML, but still haven't seen any results in the web browser to the right. Why is that?

Before we can add content that a browser will display, we have to add a body to the HTML file. Once the file has a body, many different types of content can be added within the body, like text, images, buttons, and much more.

<!DOCTYPE html>

<html>

<head>

<title>I'm Learning To Code!</title>

</head>

<body>

</body>

</html>

All of the code above demonstrates what is sometimes referred to as ["boilerplate code."](https://en.wikipedia.org/wiki/Boilerplate_code)

The term "boilerplate code" is used to describe the basic HTML code required to begin creating a web page. Without all of the elements in the boilerplate code, you'll risk starting without the minimum requirements considered to be best practice.

**Note:** The rest of the course will use code examples like the one above. To save space, however, code examples will avoid including common elements like the declaration, head, and so on. Unless otherwise specified, you can assume that the code in the example code blocks belongs directly within the HTML file's body.