Painless CSS

Build Beautiful, Intuitive Websites

BILL MEI

This is a free sample of the book
Painless CSS: Build Beautiful, Intuitive Websites
by Bill Mei

Learn more at https://www.painlesscss.com/

Chapter 1:
HTML: Hypertext Markup Language

CSS can't exist without HTML. Therefore, before we can talk about CSS, we must talk about HTML.

HTML stands for "Hypertext Markup Language". What do these words mean? Let's break them down one by one.

What is "Language"?

In computer terms, "language" just means "programming language".

What is "Markup Language"?

A programming language that defines the *look* or *structure* of text.

Imagine you are handwriting an essay, and you want to emphasize a word in a sentence. What can you do? You can use your pen to underline the word or use a highlighter to highlight it. You are "marking up" the page with your highlighter by adding "marks" that were not part of the original text.

Markup is additional meaning that is layered on top of the plain words you are writing. Sometimes this layering is critical—depending on what word you choose to emphasize in a sentence, it can change the meaning of the sentence entirely:

EMPHASIS	MEANING
"***I*** never said she stole my money"	Someone else said it, but I didn't.
"I ***never*** said she stole my money"	I simply didn't ever say it.
"I never ***said*** she stole my money"	I might have implied it in some way, but I never explicitly said it.

"I never said ***she*** stole my money"	I said someone took it; I didn't say it was she.
"I never said she ***stole*** my money"	I said she probably borrowed it.
"I never said she stole ***my*** money"	I said she stole someone else's money.
"I never said she stole my ***money***"	I said she stole something of mine, but not my money.

When you write words down, you can emphasize it in many ways. You could make it **bold**, or you could make it *italics*. You could change the color, make it bigger, or even *change the font*.

HTML Tags

To create this emphasis, HTML uses something called ***tags***. For example, to create the above sentence, this is what you would write in the HTML language:

```
<p>You could make it <strong>bold</strong>, or you could make it <
em>italics</em>.</p>
```

`<p>`, `<strong>`, and `<em>` are the names of the tags. `p` means "paragraph", `strong` means "strong emphasis" (by default, it becomes bold), and `em` means "emphasis" (by default, it becomes italics).

A reference of common tag and their meanings can be found in *Appendix 1*. If I talk about a tag that you haven't seen before, you can look it up in the Appendix or search Google to learn more.

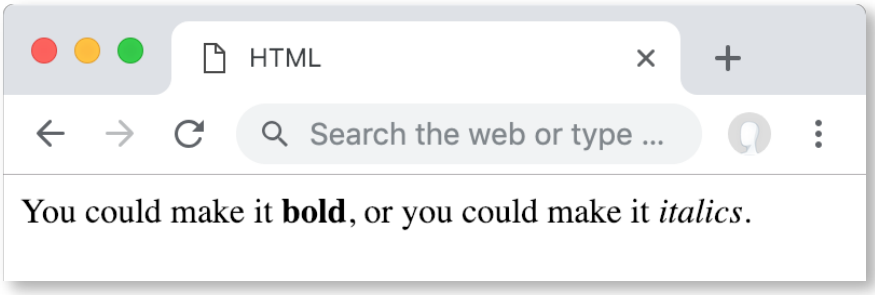
To emphasize a word, you wrap the word with the name of the tag along with the special angle bracket `<>` characters. For example, we use `<strong>` to specify the start of the bolded word, and `</strong>` to specify the end of the bolded word.

`<strong>` is the ***opening tag*** and `</strong>` is the ***closing tag*** (note the `/` slash character right after the `<` in `</strong>`). The opening tag says, "make everything from here until the closing tag bold". Therefore, all opening tags must be

accompanied with a closing tag, or else the computer won't know when to stop making everything bold!

The opening and closing tags together plus everything in between them is also known as a ***node***, or an ***HTML Element***, or just ***element***.

Loading this HTML code in our browser gives us the following result:



For brevity, whenever I refer to a tag, I'll usually omit referring to its closing tag because the closing tag is implied. For example: "Use the `<nav>` element to define a navigation menu for your website".

Semantic vs. Not Semantic

If you are a writer, you don't always care *how* a word gets emphasized, only that it does. You can leave the design decision up to your designer. No matter if the word is bold, or highlighted red, or larger in size, we can tell it has a special meaning because its design is different from the words around it.

However, if you make the *entire document* a different font, a different color, or a different size, you are not emphasizing anything. You are simply changing the design because you think the new design looks better than the old one.

This distinction between changing the design because the design creates a special meaning versus changing the design simply because you think the newer design looks pretty, is captured by the terms "Semantic" and "Not Semantic" (or "Unsemantic").

- If you change the design to create a *special meaning*, then your change is ***Semantic***.
- If you change the design just to *make it look nice*, then your change is ***Not Semantic***.

Incidentally, this is also the difference between HTML and CSS:

- **HTML** is the language used to give your text a *special meaning*.
- **CSS** is the language used to *make it look nice*.

In addition to HTML and CSS, websites usually also contain **JavaScript** (or **JS** for short), which is a programming language used to handle user interaction, as well as more advanced things like animations and dynamic content. We'll skip talking about JavaScript for now because this book is focused on CSS.

Some HTML tags have a semantic meaning, while other tags do not have a semantic meaning. In the above example, we used `<em>` to mean "emphasize this text by making it italics"; however, we could have also used the `<i>` tag to get the same visual result. The difference is `<i>` simply means "make this text italics", and it does not imply any sort of *emphasis*; it is only a design change. Thus:

- `<em>` is used to emphasize some text, so it is a ***semantic tag***.
- `<i>` is used merely to change the design, so it is an ***unsemantic tag***.

We'll talk more about how to choose between semantic and unsemantic tags in *Bonus 1: Tags for Search Engine Optimization*.

What is "Hypertext"?

Text: Just text.

Hypertext: Text that can reference (link to) other text.

Teleporting

Let's say you're a politician and you hate dogs because you are a terrible person. You are writing a new law to ban people from walking dogs in public. But you want to make an exception for service dogs, which is defined previously in some other part of the law. Here's an example of what you might write:

> *No dogs are allowed on public property, except for service dogs, as defined in Section 4(b).*

Any reader who wants to read this sentence properly must find Section 4(b), by tediously going through the following steps:

1. Remember the current page number so you can come back to it later
2. Flip to the table of contents
3. Find Section 4
4. Find Subsection (b)
5. Note the page number that the table of contents says Section 4(b) is located under
6. Go to that page number
7. Read what is contained in Section 4(b), and remember it
8. Go back to the original page number memorized from step 1
9. Now you have enough information to understand the whole sentence

Before computers, most documents behaved this way. Laws, academic papers, contracts, non-fiction books, and scientific dissertations all required you to spend most of your time cross-referencing before you could fully understand the text.

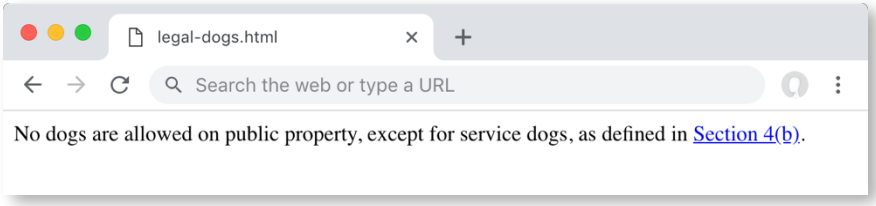
With computers, you don't have to go through this whole song and dance. You can use a ***link***, which allows us to simply click on the words "Section 4(b)" and we will teleport directly to that paragraph. We can then use the browser's "Back" button to go back to where we were previously.

Links are used like so:

```
Source Code: chapter01/legal-dogs.html
```

```
<p>No dogs are allowed on public property, except for service dogs
, as defined in <a href="https://example.com/section4b">Section 4(
b)</a>.</p>
```

Which gives the following result:



The `<a>` tag is called an "anchor" tag and it defines a link. Its `href` property defines where you should be teleported to once you click on the link. We'll talk more about what a "property" is in *Chapter 2: CSS Styles*.

Notice also that you can include tags inside other tags—this is also known as ***nesting***. We'll talk more about what exactly happens when you nest tags in *Chapter 4: The DOM*.

Anyway, teleporting around this way almost feels like you're jumping through a wormhole in the fourth dimension, doesn't it? In science fiction, the fourth dimension is often called "hyperspace". Thus, these links are known as ***hyperlinks***.

Therefore, if you have a text document that contains hyperlinks, we can call it a ***hypertext*** document, to distinguish it from a regular text document that does not have hyperlinks.

Finally, if we put together what we learned about each term in *Hypertext Markup Language*, what we get is:

Hypertext Markup Language (HTML):

A *programming language* that defines the *look or structure* of *hypertext*.

The Hyperlink Legacy

One of the reasons why the Internet got so huge and important, is that it turns out the action of taking a hyperlink is so powerful that you can build an incredible number of applications on top of this relatively simple action.

This happened because people realized you can use a hyperlink to teleport between things that are not just text. For example, in a video player application, you can think of clicking on the "play/pause" button as taking a hyperlink through a wormhole between a universe where the video is paused, and a parallel universe where the video is playing.

This conceptual breakthrough is what leads to the powerful applications we have on the Internet today. Today's Internet consists of hyperlinks that access different states of text, images, music, videos, animated drawings, 3D models, maps, and many more media that would have been unimaginable in the early days of the Internet.

Because hypertext is not only text anymore, it would be more accurate to call it ***hypermedia***, but this term is not widely used, and programmers still call it "hypertext" because that's just what we've always called it.

Throughout this book, I'll still say "text" or "document" instead of "media", because the name sticks around from the original idea of text documents, and "document" is still the standard terminology. However, from now on in the book whenever I say "text" or "document", you can assume that what I really mean is "media".