

Semantic HTML

by Devmountain Tutorials



WHAT'S COVERED

This section will explore semantic HTML by discussing:

1. WHY WRITE SEMANTIC HTML?
2. SEMANTIC ELEMENTS

1. WHY WRITE SEMANTIC HTML?

In a previous section, you learned that it's possible to make any HTML element look and behave differently using a combination of CSS and JavaScript. For example, in **Challenge 1.3 – Tools of the Trade**, we made a paragraph look like a button with CSS and used JavaScript to make the paragraph clickable.

We've included this interactive example again. When you click on **Hello, friend!** an alert box displays asking, "What's your name?" If you type in your name and click OK, you'll see the button update to say hello to you.

As you continue learning about HTML, you'll find many resources and articles about the importance of using **semantic HTML**. This means using HTML correctly by staying true to an element's original, intended purpose. In other words, even though it's possible to make a paragraph look and act like a button, it's bad practice and you shouldn't do it.

Semantic HTML doesn't just help make content more readable for humans, it also helps computer programs. Programs like assistive technology software rely on HTML to make sense of a web page's contents. For example, severely visually impaired people might not read web pages very often — they listen to them using programs called screen readers instead. Screen readers don't read out the entire contents of a web page from top to bottom. Instead, they might first read out the outline of the document using the page's headings and subheadings. That way, users can quickly find the information they care to hear more about. Without proper headings and subheadings, users will be forced to listen to the whole document read out loud. Other programs might use headings to determine the important keywords of a page. For example, Google looks at a page's headings to influence a page's search rankings.

Another way to understand the importance of writing semantic HTML is to compare and contrast how developers wrote HTML before semantics became a widespread concern. The next section will provide some historical context around HTML and semantics, starting with a look at earlier versions of HTML and their limited semantics.



Semantic HTML

To write semantic HTML means to use HTML correctly by staying true to an element's original, intended purpose.

2. SEMANTIC ELEMENTS

Early HTML had limited semantics, especially when it came to grouping related content together. Developers only had one element they could use to group content — the `div` element. However, since the `div` element does not suggest any meaning about its content to a browser, it only helped humans understand how content is grouped. Consider this navigation menu:

```
<div class="navigation">
  <a href="/">Home</a>
  <a href="/about">About</a>
  <a href="/sign-in">Sign In</a>
</div>
```

A screen reader wouldn't be able to tell that the `div` above makes up a navigational menu, even though it has `class="navigation"`.

In the latest version of HTML — HTML5 — new elements, called **semantic elements**, were introduced and gave developers a more diverse set of elements to choose from when defining the structure of a document. Below are the names of a few of the new HTML5 elements with descriptions of how they might be used.

Code Element	Element Name	Element Description
<code><<nav></nav>></code>	nav	A section with navigation links that appear often on a site.
<code><html></html></code>	html	It wraps all the content on the page and is also known as the root element.
<code><article></article></code>	article	A piece of content that is self-contained like a blog post, comment, or widget.
<code><section></section></code>	section	A section of a document. For example, a video and its play/pause buttons should each be put in a separate section (one for the video and one for the play/pause buttons).
<code><aside></aside></code>	aside	A section that is related to some content but doesn't belong in its main flow. One example that you're familiar with is the Term to Know box.
<code><header></header></code>	header	A section that contains a heading. For example, many web pages have headers that contain the website's logo, title, and navigation menu. It can also be used inside article or section . For example, an article's header might have a title and a byline with the name of the author. Headers do not have to appear at the beginning of a page or section.
<code><footer></footer></code>	footer	A page footer. This might have a copyright or links. Footers do not have to appear at the end of a page or section.

We can rewrite the example navigation menu above using the semantic element, **nav**:

```
<nav>
  <a href="/">Home</a>
  <a href="/about">About</a>
  <a href="/sign-in">Sign In</a>
</nav>
```

You've learned a lot in this challenge! You can recognize the different parts of HTML, combine elements to create a web page, and identify different semantic elements. In the next challenge we'll explore how to style HTML by using CSS.



TERMS TO KNOW

Semantic HTML

To write semantic HTML means to use HTML correctly by staying true to an element's original, intended purpose.