



DECEMBER 12, 2019 / #HTML5

# Semantic HTML5 Elements Explained

Semantic HTML elements are those that clearly describe their meaning in a human- and machine-readable way.

Elements such as `<header>`, `<footer>` and `<article>` are all considered semantic because they accurately describe the purpose of the element and the type of content that is inside them.

## What are Semantic Elements?

HTML was originally created as a markup language to describe documents on the early internet. As the internet grew and was adopted by more people, its needs changed.

Where the internet was originally intended for sharing scientific documents, now people wanted to share other things as well. Very quickly, people started wanting to make the web look nicer.

Because the web was not initially built to be designed, programmers used different hacks to get things laid out in different ways. Rather than using the `<table></table>` to describe information using a table, programmers would use them to position other elements on a page.

Learn to code — free 3,000-hour curriculum

purpose. For example, instead of `<header>` this was often written as `<div class="header">`.

As HTML5 is still relatively new, this use of non-semantic elements is still very common on websites today.

## List of new semantic elements

The semantic elements added in HTML5 are:

- `<article>`
- `<aside>`
- `<details>`
- `<figcaption>`
- `<figure>`
- `<footer>`
- `<header>`
- `<main>`
- `<mark>`
- `<nav>`
- `<section>`
- `<summary>`
- `<time>`

Elements such as `<header>`, `<nav>`, `<section>`, `<article>`, `<aside>`, and `<footer>` act more or less like `<div>` elements. They group other elements together into page sections. However where

Learn to code — free 3,000-hour curriculum

An example of semantic element layout by w3schools

## Why use semantic elements?

To look at the benefits of semantic elements, here are two pieces of HTML code. This first block of code uses semantic elements:

```
<header></header>
<section>
  <article>
    <figure>
      <img>
      <figcaption></figcaption>
    </figure>
  </article>
</section>
<footer></footer>
```

Whilst this second block of code uses non-semantic elements:

```
<div id="header"></div>
<div class="section">
  <div class="article">
    <div class="figure">
      <img>
      <div class="figcaption"></div>
    </div>
  </div>
</div>
<div id="footer"></div>
```

Learn to code — free 3,000-hour curriculum

...reading through hundreds or thousands of lines of code. The easier it is to read and understand that code, the easier it makes your job.

It has **greater accessibility**. You are not the only one that finds semantic elements easier to understand. Search engines and assistive technologies (like screen readers for users with a sight impairment) are also able to better understand the context and content of your website, meaning a better experience for your users.

Overall, semantic elements also lead to more **consistent code**. When creating a header using non-semantic elements, different programmers might write this as `<div class="header">`, `<div id="header">`, `<div class="head">`, or simply `<div>`. There are so many ways that you can create a header element, and they all depend on the personal preference of the programmer. By creating a standard semantic element, it makes it easier for everyone.

Since October 2014, HTML4 got upgraded to HTML5, along with some new “semantic” elements. To this day, some of us might still be confused as to why so many different elements that doesn’t seem to show any major changes.

`<section>` **and** `<article>`

“What’s the difference?”, you may ask. Both these elements are used for sectioning a content, and yes, they can definitely be used interchangeably. It’s a matter of in which situation. HTML4 offered only one type of container element, which is `<div>`. While this is still used in HTML5, HTML5 provided us with `<section>` and `<article>` in a way to replace `<div>`.

Learn to code — free 3,000-hour curriculum

1. An article is intended to be independently distributable or reusable.
2. A section is a thematic grouping of content.

```
<section>
  <p>Top Stories</p>
  <section>
    <p>News</p>
    <article>Story 1</article>
    <article>Story 2</article>
    <article>Story 3</article>
  </section>
  <section>
    <p>Sport</p>
    <article>Story 1</article>
    <article>Story 2</article>
    <article>Story 3</article>
  </section>
</section>
```

## <header> **and** <hgroup>

The <header> element is generally found at the top of a document, a section, or an article and usually contains the main heading and some navigation and search tools.

```
<header>
  <h1>Company A</h1>
  <ul>
    <li><a href="/home">Home</a></li>
    <li><a href="/about">About</a></li>
    <li><a href="/contact">Contact us</a></li>
  </ul>
```

Learn to code — free 3,000-hour curriculum

```
</header>
```

The `<hgroup>` element should be used where you want a main heading with one or more subheadings.

```
<hgroup>
  <h1>Heading 1</h1>
  <h2>Subheading 1</h2>
  <h2>Subheading 2</h2>
</hgroup>
```

REMEMBER, that the `<header>` element can contain any content, but the `<hgroup>` element can only contain other headers, that is `<h1>` to `<h6>` and including `<hgroup>`.

## `<aside>`

The `<aside>` element is intended for content that is not part of the flow of the text in which it appears, however still related in some way. This of `<aside>` as a sidebar to your main content.

```
<aside>
  <p>This is a sidebar, for example a terminology definition or
</aside>
```



Before HTML5, our menus were created with `<ul>` 's and `<li>` 's. Now, together with these, we can separate our menu items with a `<nav>`, for navigation between your pages. You can have any

Learn to code — free 3,000-hour curriculum

.....

```
<nav>
  <ul>
    <li><a href="/home">Home</a></li>
    <li><a href="/about">About</a></li>
    <li><a href="/contact">Contact us</a></li>
  </ul>
</nav>
```

## <footer>

If there is a <header> there must be a <footer> .A <footer> is generally found at the bottom of a document, a section, or an article. Just like the <header> the content is generally meta-information, such as author details, legal information, and/or links to related information. It is also valid to include <section> elements within a footer.

```
<footer>&copy; Company A</footer>
```

## <small>

The <small> element often appears within a <footer> or <aside> element which would usually contain copyright information or legal disclaimers, and other such fine print. However, this is not intended to make the text smaller. It is just describing its content, not prescribing presentation.

Learn to code — free 3,000-hour curriculum

## <time>

The `<time>` element allows an unambiguous ISO 8601 date to be attached to a human-readable version of that date.

```
<time datetime="2017-10-31T11:21:00+02:00">Tuesday, 31 October 2017 11:21 AM CEST</time>
```

Why bother with `<time>` ? While humans can read time that can disambiguate through context in the normal way, the computers can read the ISO 8601 date and see the date, time, and the time zone.

## <figure> **and** <figcaption>

`<figure>` is for wrapping your image content around it, and `<figcaption>` is to caption your image.

```
<figure>  
    
  <figcaption>Cover art for Middle-earth: Shadow of Mordor</figcaption>  
</figure>
```

## Learn more about the new HTML5 elements:

- [w3schools](https://www.w3schools.com/html5/semantic-html5-elements/) provides simple and clear descriptions of many of the new elements and how/where they should be used.



Learn to code — free 3,000-hour curriculum

---

If this article was helpful, [tweet it](#).

Learn to code for free. freeCodeCamp's open source curriculum has helped more than 40,000 people get jobs as developers.

[Get started](#)

freeCodeCamp is a donor-supported tax-exempt 501(c)(3) charity organization (United States Federal Tax Identification Number: 82-0779546)

Our mission: to help people learn to code for free. We accomplish this by creating thousands of videos, articles, and interactive coding lessons - all freely available to the public. We also have thousands of freeCodeCamp study groups around the world.

Donations to freeCodeCamp go toward our education initiatives, and help pay for servers, services, and staff.

**You can [make a tax-deductible donation here](#).**

### Trending Guides

Hello World in Java

Python Set

What is a 429 Error?

What is REST?

Inline Style in HTML

What is CRUD?

Python Absolute Value

%.2f in Python

Java Switch Statement

Git Reset Origin

Change Font with HTML

What is Localhost?

502 Bad Gateway Error

JS Array.includes()

[Forum](#)

[Donate](#)

[Learn to code — free 3,000-hour curriculum](#)

[Python Global Variable](#)

[Python enumerate\(\)](#)

[Create a Set in Python](#)

[What's a Data Analyst Do?](#)

[Remove a Dir in Linux](#)

[Refresh Page in JavaScript](#)

[What is Coding Used For?](#)

[Remove Underline from Link](#)

[JavaScript String Format](#)

[How to Clear an Array in JS](#)

[JavaScript String to Date](#)

[Capitalize 1st Letter in JS](#)

### **Our Charity**

[About](#) [Alumni Network](#) [Open Source](#) [Shop](#) [Support](#) [Sponsors](#) [Academic Honesty](#)

[Code of Conduct](#) [Privacy Policy](#) [Terms of Service](#) [Copyright Policy](#)