



Semantic Elements

The so-called *semantic elements* were introduced to reflect important distinctions in the **meaning** of content. Few of these elements have any impact whatsoever on the content's *appearance*, but using them helps browsers, screen-readers, and other tools understand the content they contain.

On This Page

[Header](#)[Footer](#)[Main](#)[Article](#)[Section](#)[Aside](#)[Nav](#)[Figure & Figcaption](#)[Video Lesson](#)[Practice Exercise](#)

Header

Not to be confused with `head` or headings (`h1` , `h2` , etc.), the `header` element should be used to contain introductory content for a page or `article` . It might contain things such as the title, a tagline, authorship information, etcetera.

You may use more than one `header` per page. For instance, you may want a `header` for the page itself, and an additional header for an `article` contained on the page.

In spite of the name, there is no requirement that a `header` be at the top of the page or article.

Footer

The `footer` element should be used to contain content about the creator or creation of a page or `article` . It might contain things such as publication date, authorship information, links to related content, etcetera.

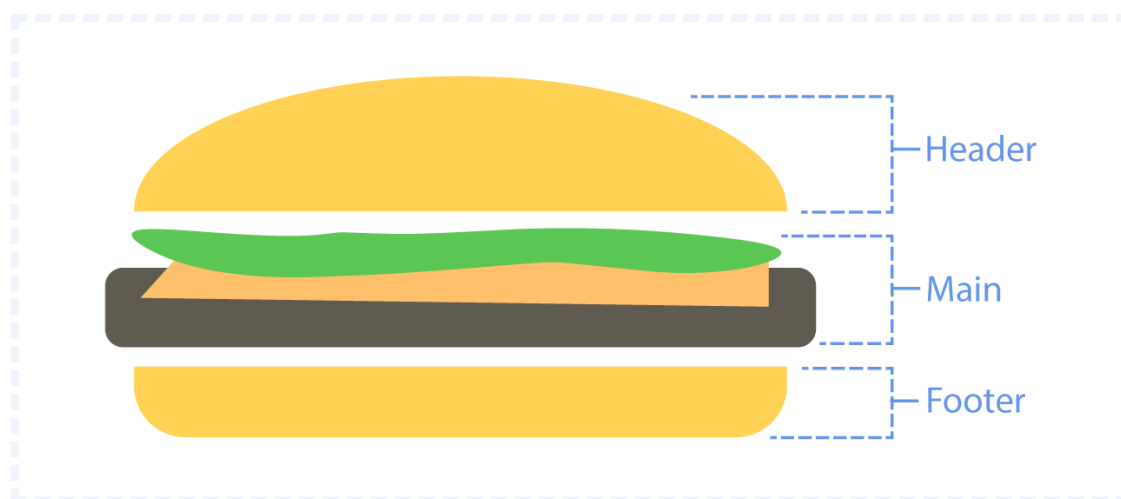
Similar to the `header` element, you may use more than one `footer` per page. For instance, you may want a `footer` for the page itself, and an additional header for an `article` contained on the page.

In spite of the name, there is no requirement that a `footer` be at the bottom of the page or article.

Main

The `main` element represents the primary content of a page. As such, **it should only be used once per page**. This is the defacto target for screen-reader users who wish to skip the sort of introductory content and navigation that might, for instance, be found in the `header` of a page.

A common pattern, though certainly *not required*, is for a site to have a common `header` on every page with things like a logo and the main `nav` , a unique `main` with the specific contents of each page, then a common `footer` on every page with things like copyright information or a company-specific navigation. I tend to think of this pattern as a sort of sandwich:



A common pattern is for pages to have a header, then a main, then a footer.

Article

The `article` element signifies content that could stand alone as a document. A good way to think of it is to ask yourself if the content would make sense if you shared it with someone who didn't know what website it came from. Examples of content that might use an article element are articles (go figure), blog posts, reviews, essays, user-submitted comments, etcetera.

Because an article acts as an independent unit of content, it may contain most of the other semantic elements, such as `header` , `footer` , `section` , `aside`, and `nav` . It could also contain nested articles, such as when the `article` element is used for post comments.

To decide if an article is appropriate, ask yourself if the content would make sense completely on its own.

The `article` element should only be used for the full and actual content it represents. In other words, it should not be used for the links in a list of articles or posts.

Section

The `section` element is the most nonspecific of the semantic elements and represents a thematically-related section of a page, application, or `article` . It should be used only in instances where other semantic elements don't make sense. Example uses might include, a chapter or subsection of an `article` or a distinct functional area of a page.

The `section` element should **always** have a natural heading demarcating its content.

If you simply want a container for styling or scripting purposes, a `div` should be used instead.

Aside

The `aside` element should contain secondary or related content to a page or `article`. If used in an `article`, the content should be related to the `article`, such as a list of related terms or a timeline. On a page as a whole, it might include a blogroll, advertising, etcetera.

You can think of the `aside` element as containing *bonus* content; something extra, but not required for understanding.

Nav

The `nav` element is relatively straightforward and should be used for sets of links that represent major navigation for a site or page. Examples might include primary site navigation, pagination links, a table of contents, or breadcrumb navigation.

`nav` is not typically used for lists or articles or posts.

Figure & Figcaption

The `figure` element is used to contain a useful image, illustration, diagram, code snippet, et cetera, that helps to explain the content in the text.

You might think of `figure` as a way of saying "for example...."

It is important to remember `figure` should not be used for purely decorative or supplemental images or content. A great example would be a chart illustrating data that is referenced in a post or an image that shows key information. Things like banner images or incidental images should not use `figure`.

It is important to note the difference between `figure` and `aside`. If the content is more generally related and not especially helpful to understanding, use `aside`. If the content is essential but its position in the flow of content isn't important, use `figure`.

It's okay to have a figure without a figcaption, but every figcaption should be inside a figure.

The `figcaption` element can be used for a caption or legend within a `figure`. It is not required for every `figure` to contain a `figcaption`. The `figcaption` may come before or after the primary content of a `figure`. Only one `figcaption` may be used per `figure`.

Video Lesson

The following video demo is fully interactive. You can pause at any time to directly edit the code, and resume playback to restore to where you left off.

[Create Account](#)

[Log In](#)

Practice Exercise

[Open exercise in a new tab](#) 

Let's test your knowledge

Enter your first name

Start the Lesson

You must complete at least 70% of steps to pass.

You may start over at any time.

Works best in the Chrome browser.

Table of Contents

Introductions

- **What is a website?**

How are websites built and what are they made of?

[A Network of Computers](#) | [Finding Websites](#) | [How Pages are Built](#) |
[Code Languages and File Types](#) | [Practice Exercise](#)

HTML: Content and Structure

- **HTML Introduction**

What is HTML? What are the rules of the language?

[A Language for Content](#) | [Elements](#) | [Semantics](#) | [Attributes](#) | [HTML Files](#) |
[Practice Exercise](#)

- **File Structure**

How to structure the folders and files of a website project

[Project Folder](#) | [Home Page](#) | [Subfolders](#) | [Parent Folders](#) |
[Practice Exercise](#)

- **Skeleton Code**

The code that should be included in every HTML document

[Overview](#) | [Doctype](#) | [HTML](#) | [Head](#) | [Meta](#) | [Title](#) | [Body](#) |
[Video Lesson](#) | [Practice Exercise](#)

- **Basic HTML Text Elements**

The most common elements for structuring text content

[Headings](#) | [Paragraph and Small](#) | [Strong and Emphasis](#) | [Lists](#) | [Breaking Elements](#) | [Practice Exercise](#)

- **Links**

The most fundamental type of interactivity in HTML

[Absolute Links](#) | [Relative Links](#) | [Anchor Links](#) | [Email Links](#) | [Optional Attributes](#) | [Video Lesson](#) | [Practice Exercise](#) | [Tool Links](#)

- **IDs & Classes**

Attributes for giving names to elements

[Usage Rules](#) | [IDs](#) | [Classes](#) | [Multiple Classes or IDs](#) | [Practice Exercise](#)

- **Semantic Elements**

A set of elements aimed at adding meaning to the parts of a web page

[Header](#) | [Footer](#) | [Main](#) | [Article](#) | [Section](#) | [Aside](#) | [Nav](#) | [Figure & Figcaption](#) | [Video Lesson](#) | [Practice Exercise](#)

- **Divs & Spans**

Meaningless elements

[Div versus Span](#) | [Practice Exercise](#)

- **Images**

Image formats, the HTML image element, and inline SVG

[Common Image Formats](#) | [The Image Element](#) | [Inline SVG](#) | [Video Lesson](#) | [Practice Exercise](#) | [Resource Links](#) | [Tool Links](#)

- **Forms**

How to accept input or information from users

[The Form Element](#) | [The Input Element](#) | [The Textarea Element](#) | [Select & Option](#) | [The Name Attribute](#) | [The Label Element](#) | [Submit Buttons](#) | [Additional Attributes](#) | [Video Lesson](#) | [Practice Exercise](#) | [Resources](#)

- **Interactive Elements**

Elements with built-in interactive open-and-close behaviors

[Details & Summary](#) | [Dialog](#) | [Video Lesson](#) | [Resource Links](#)

CSS: Style and Layout

- **Introduction to CSS**

What is the purpose of CSS? How does it work to style HTML?

[CSS Files](#) | [Syntax](#) | [The Cascade](#) | [Default Styles](#) | [Comments in CSS](#) | [Practice Exercise](#) | [Resource Links](#)

- **CSS Selectors**

Different ways CSS can select HTML elements to style

[Element Selector](#) | [Class Selector](#) | [Decendant Combinator](#) | [Selector List](#) | [Attribute Selector](#) | [Advanced Selectors](#) | [Practice Exercise](#)

- **Common Size Units**

Different ways to apply measurements in CSS

[Pixels](#) | [Rems](#) | [Percentages](#) | [Viewport Units](#) | [Ems](#) | [Practice Exercise](#) | [Resource Links](#)

- **CSS Color Definitions**

Several common ways to specify a color in CSS

[Hexadecimal](#) | [RGB & RGBA](#) | [Named Colors](#) | [Practice Exercise](#) | [Resource Links](#) | [Tool Links](#)

- **Typographic Properties**

A wide variety of ways to customize and control the appearance of text

[Font-Family](#) | [Font-Size](#) | [Font-Weight](#) | [Font-Style](#) | [Color](#) | [Line-Height](#) | [Text-Transform](#) | [Text-Decoration](#) | [Letter-Spacing](#) | [Video Lesson](#) | [Practice Exercise](#) | [Resource Links](#) | [Tool Links](#)

- **Styling Lists**

How to control the bullets, numbers, and spacing associated with list items

[List Style Type](#) | [Nested Lists](#) | [Default Padding](#) | [List Style Position](#) | [Practice Exercise](#) | [Resource Links](#)

- **The Box Model**

How block elements are sized and spaced in CSS

[Width & Height](#) | [Padding](#) | [Margin](#) | [Border](#) | [Box Sizing](#) | [Interactive Demo](#) | [Video Lesson](#) | [Practice Exercise](#) | [Resource Links](#)

- **Link States**

Different states of link interaction and how to style them

[Normal Links](#) | [Visited](#) | [Hover](#) | [Focus](#) | [Active](#) | [Order of Selectors](#) | [Full Example](#) | [Video Lesson](#) | [Practice Exercise](#)

- **Float & Clear**

How to wrap text around an image or another element

[Float](#) | [Clear](#) | [Video Lesson](#) | [Practice Exercise](#)

- **Background Colors & Images**

How to control the appearance of the area behind a page or element

[Background Color](#) | [Background Image](#) | [Background Repeat](#) | [Background Position](#) | [Background Attachment](#) | [Background Size](#) | [Video Lesson](#) | [Practice Exercise](#) | [Resource Links](#)

- **Special Effects**

How to apply unique styles such as shadows and rounded corners

[Two Kinds of Shadows](#) | [Box Shadow](#) | [Text Shadow](#) | [Rounded Corners](#) | [Video Lesson](#) | [Practice Exercise](#) | [Tool Links](#)

- **Fluid Width**

How to make elements stretch or squish depending to fit the space available

[Setting Fixed Widths](#) | [Percentage Widths](#) | [Max-Width & Min-Width](#) | [Height Properties](#) | [Video Lesson](#) | [Practice Exercise](#)

- **Responsive Design**

Using media queries to adapt a page to different screen sizes

[Mobile First](#) | [Media Queries](#) | [Video Lesson](#) | [Practice Exercise](#)

- **The Display Property**

How to control the basic layout behavior of an element

[Block](#) | [Inline](#) | [Inline-Block](#) | [None](#) | [Other Display Values](#) | [Video Lesson](#) | [Practice Exercise](#)

- **Web Fonts**

Custom typefaces for web pages

[Web Font Basics](#) | [Web Font Services](#) | [Weights & Styles](#) | [Practice Exercise](#) | [Resource Links](#) | [Tool Links](#)

- **The Position Property**

For when you want full manual control of the placement of an element

[Position Basics](#) | [Document Flow](#) | [Relative](#) | [Absolute](#) | [Fixed](#) | [Video Lesson](#) | [Practice Exercise](#)

- **CSS Transitions & Animation**

For adding motion to web pages

[Transitions](#) | [Keyframe Animation](#) | [Video Lesson](#) | [Resource Links](#) | [Tool Links](#)

- **Transform**

For scaling, rotating, skewing, or moving an element

[Basic Usage](#) | [Common Transform Functions](#)

- **Flexbox**

For laying out items across a single dimension

[Activating Flexbox](#) | [Flex Direction](#) | [Justify Content](#) | [Align Items](#) | [Video Lesson](#) | [Practice Exercise](#) | [Resource Links](#)

- **Grid**

For laying out items in a two-dimensional grid

[Terminology](#) | [Activating Grid](#) | [Grid Template](#) | [Grid Gap](#) | [Justify Items](#) | [Align Items](#) | [Positioning Grid Children](#) | [Video Lesson](#) | [Practice Exercise](#) | [Resource Links](#)

- **Advanced Typographic CSS**

Techniques for creating cohesive and responsive typography

[Modular Scales](#) | [Responsive Type](#) | [Video Lesson](#) | [Resource Links](#) | [Tool Links](#)

- **Variables**

For creating reusable CSS values

[Defining Variables](#) | [Redefining and Overriding](#) | [Using Variables](#) | [Video Lesson](#) | [Resource Links](#)

- **CSS-Generated Content**

Insert visual content with CSS

[Pseudo-Elements](#) | [Styling Inserted Content](#) | [Absolute Positioning](#)

Advanced / Other

- **[Styling a Responsive Nav](#)**

A simple example of responsive design for site navigation

[The HTML](#) | [Mobile-First Styles](#) | [Making it Responsive](#)

- **[Site Icons \(Favicons\)](#)**

An icon for the browser tab and elsewhere

[Creating a Favicon](#) | [Using a Favicon on a Site](#)

- **[Git / Github](#)**

Version control to keep track of changes

[Repositories](#) | [Commits](#) | [Workflow](#) | [Video Lesson](#) | [Resource Links](#)

- **[Going Beyond](#)**

Want to know more about front-end coding?

[General](#) | [More HTML](#) | [More CSS](#) | [Animation](#) | [Javascript](#)

Reference Resources

- **[Practical Challenges](#)**

A series of step-by-step challenges to practice what you've learned by creating a realistic website

- **[Code Style Guide](#)**

A guide for formatting and other best practices for writing your code

[HTML Guide](#) | [CSS Guide](#)

- **[Glossary](#)**

Some of the most common HTML & CSS terms and their meanings

[General Terms](#) | [HTML Terms](#) | [CSS Terms](#)

- **[Website Project Checklist](#)**

Make sure you have considered these items before completing a website project

[General](#) | [HTML-Specific](#) | [CSS-Specific](#)

- **Reference Index**

An index of all HTML elements, HTML attributes, and CSS properties discussed on this site

[HTML Elements](#) | [HTML Attributes](#) | [CSS Properties](#)

This book is made available under a [Creative Commons Attribution-NonCommercial 4.0 International License](#).