



HTML and CSS

8th Edition

Chapter 1: Webpage Building Blocks



Objectives

- Identify the basic building blocks of HTML webpages.
- Distinguish the different parts of HTML markup.
- Define best practices for building effective websites.
- Explain best practices for writing HTML.
- Mark up content with semantic HTML.
- Improve site accessibility and efficiency using semantic HTML.
- Explain CSS's role in HTML presentation.
- Create proper file and folder names.



Webpage Building Blocks

- A webpage has three main components:
 - **Text content:** The bare text that appears on the page.
 - **References to other files:** Loads images, video, and audio files, as well as CSS style sheets to control page's appearance, and JavaScript files that add special behavior to your page. Also links to other HTML pages and assets.
 - **Markup:** The HTML elements that describe your text content and make the references work.
- All three components are text-based, saved in text-only format.
- Page also includes HTML information about the page itself for browsers and [search engines](#).



Thinking in HTML

- When writing HTML, you describe your webpage's content using pre-defined element tags.
- Examples of predefined elements:
 - The `p` element is for paragraphs.
 - The `abbr` element is for abbreviations.
 - The `li` element is for list items.
- HTML elements describe what your content is, not how it looks. CSS controls your content's appearance.



A Basic HTML Page

- Foundations of a page:
 - Every webpage begins with the [DOCTYPE](#) declaration.
 - It also includes the html, head, and body elements.
 - Language code assigned to lang and text between <title> and </title> are both customizable parts of page shell.
 - Equivalent of a blank sheet of paper.
- HTML tag structure:
 - Uses < and > to enclose each HTML tag.
 - **Start tag:** <head> marks the beginning of an element.
 - **End tag:** </head> marks its end.
 - A few elements, like meta, don't have an end tag.



The Top and Head of a Webpage

- Everything above the `<body>` start tag is instructional information for browsers and search engines:
 - **`<!DOCTYPE html>`** tells browsers that it is an HTML5 page. Always make it the first line.
 - **html element** encloses the rest of the page between `<html lang="en">` and the `</html>` end tag.
 - **`lang="en"`** indicates English is the page's default language.
 - **Document head** is the area between `<head>` and `</head>`.
 - **Title** is the text between `<title>` and `</title>`.
 - Only part above the body visible to users, displaying on the browser tab, at the very top of the browser window, and typically as the default name of a browser bookmark.



The Body of a Webpage: Your Content

- Markup surrounding text content doesn't appear when you view the page in a browser.
- HTML element tags are essential because they describe your content.
- Six of the most common HTML element tags: a, article, em, h1, img, and p.
- Each has its own meaning; for instance, h1 is a heading, a is a link, and p is a paragraph.



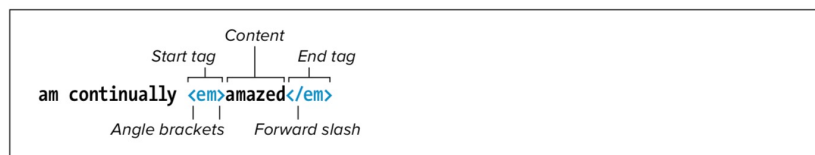
Indenting Your HTML

- The code's indentation has absolutely no bearing on how the content displays in the browser
- The `pre` element is the one exception.
- However, it's customary to indent some elements when they are contained in another element. It makes it easier to view the hierarchy of elements as you read through and work with your code.



Markup: Elements

- Markup includes: elements, attributes, and values.
- **Elements** describe different parts of a webpage.
 - Most elements can contain both text and other elements.
 - Elements consist of a **start tag**, the **content**, and an **end tag**.



- **Empty elements (void elements)** do not surround any text content. They have a single tag that serves to both open and close the element (Ex. Meta element)
- Customary to type element names in all lowercase. Uppercase is considered a dated practice.



Markup: Attributes and Values

- Attributes contain additional information about an element.
 - Attributes are located inside an element's start tag.
 - Type attribute names in lowercase and enclose attribute's value in quotation marks.
 - Some attributes only accept specific values.
- **Predefined (or enumerated) values:** You must select a value from a standard list of choices.
 - Be sure to write enumerated values in all lowercase letters.



Markup: Attributes and Values

- **Numeric value:** A few attributes require a number value, particularly those describing size and length.
 - Numeric value never includes a unit type, just the number.
 - In video or image size, values are understood as pixels.
- **URL values:** Some attributes reference other files and must contain values in the form of a URL.
- **Boolean attribute:** A special kind of attribute with optional value. If present, it evaluates to true.
 - If you include a value, set it to the name of the attribute itself.
 - Boolean attributes are predefined.



Markup: Parents and Children

- Family tree structure is a key feature of HTML code:
 - When one element contains another, it is considered the **parent element**.
 - The enclosed element is considered the **child element**.
 - Any elements contained in the child element are considered **descendants** of the outer, parent element or **ancestor**.
 - When elements contain other elements, each element must be properly **nested**—fully contained within its parent.
 - When using an end tag, it should correspond to the previous unclosed start tag.
 - This structure facilitates both styling elements with CSS and applying JavaScript behavior.



A Webpage's Text Content

- A page's text content is anything besides the markup.
- HTML text has some differences from regular text:
 - When viewing the page in a browser, the extra returns and spaces are ignored and the character reference is replaced by the corresponding [symbol](#) (©).
- HTML used to be restricted to ASCII characters.
 - Accented characters and symbols were created with special character references.
 - Set the charset value as UTF-8 or utf-8 to mitigate issues with special characters.



Links, Images, and Other Non-Text Content

- Non-text assets—links, images, and videos—are saved as independent files, referenced in the HTML page, and then displayed by the browser.
- Browsers can't handle every kind of file, so **plug-ins** are installed to provide missing features.
- **Flash** is a widespread plug-in for video, but it demands a lot of computing power.
- Modern browsers have built-in media players, and you can still provide a Flash player as a fallback for older browsers.



File and Folder Names

- **Use lowercase names:** Prevents typos.
 - Mixing upper- and lowercase letters makes it harder to type the proper address.
- **Use the proper extension:** .html is customary, but .htm also works.
 - If the page has some other extension, such as .txt, the browser will treat it as text and display the code to the visitor.
- **Separate words with a dash:** Never include spaces between words in your file and folder names.
 - Underscores work too, but dashes are preferred by search engines



URLs

- **URL:** Each file on the Internet has a unique URL, the web address, that describes a file's location and what a browser should do with it.
- A URL has a scheme, a host, and a path:
 - **Scheme:** Tells the browser how to deal with the file. The scheme is followed by a colon and two forward slashes.
 - **HTTP:** The most common scheme; used to access webpages.
 - **HTTPS:** Used for secure webpages, such as e-commerce sites.
 - **Host (host name):** Where the file is located.
 - **Path:** Includes directory names that lead to a file and the file name itself, all of which are optional.



Absolute URLs

- **Absolute URL:** Contains all the information that points to a directory or file, including the **scheme**, the **host**, and the **path**. Ex.:
`src="https://www.w3schools.com/images/img_girl.jpg".`
- This means you can reference an absolute URL from any webpage on any host with the same URL every time.
- When referencing a file located on someone else's web server, always use an absolute URL.
- Also use absolute URLs for FTP sites or any kind of URL that doesn't use an HTTP protocol.



Relative URLs

- **Relative URL:** Describes location of file in relation to the file that contains the URL reference.
- Linking to:
 - **File in the same directory:** Use file name and extension (Ex. index.html).
 - **File in subdirectory:** Create URL by typing the name of the subdirectory, followed by a forward slash, and then the name and extension of the desired file (Ex. main/index.html).
 - **File in higher directory:** Use two periods and a forward slash for each directory level above the current file (Ex. ../)
 - **Root relative URLs:** If files are on a web server, jump straight to the site's root and drill down to the targeted file.



HTML: Markup with Meaning

- **Semantics:** The meaning of a webpage's content.
- **Semantic HTML:** Refers to content that is marked up with the HTML elements that best describe it and without regard for how the content should look.
- Writing semantic HTML is a cornerstone of an effective website.



The Semantics of Our Basic HTML Page

- **Article element:** Holds all content for a self-contained composition.
- **Heading element:** HTML provides six heading levels to structure page content.
 - h1 is most important, h2 is a subheading of h1, h3 is a subheading of h2, and so on.
- **Img element:** Primary way to display an image.
- **Alt attribute:** Text that displays if the browser can't load the image or if the page is opened in a text-only browser.



The Semantics of Our Basic HTML Page

- **p element:** Marks a paragraph. A paragraph can contain a single sentence or several sentences.
- **em element:** Used for “stress emphasis.”
- **a (anchor) element:** The basic page defines a link to another page with the a element.
 - The most powerful element because it links to other pages, specific page sections, and files.
- HTML doesn't provide an element for every type of content, instead focusing on elements that cover the vast majority of cases.



Why Semantics Matter

- Improved accessibility, the practice of making your content available to all users, regardless of their capabilities, and interoperability
- Improved search engine optimization (SEO)
- Easier code maintenance and styling with CSS
- (Often) lighter code and faster pages



A Browser's Default Display of Webpages

- Every web browser has a built-in CSS file (a style sheet) that dictates how each HTML element displays by default.
- Writing your own CSS overwrites these settings.
- Default presentation varies slightly from browser to browser, but on the whole it is fairly consistent.
- More importantly, the content's underlying structure and meaning as defined by your HTML remain the same.



Block-level, Inline, and HTML5

- **Block-level elements:** Display on their own line.
- **Inline elements:** Display within a line of text.
- Browsers haven't changed the default display rules for these elements.



Key Takeaways

- A webpage has three main components: text content, references to other files, and markup.
- HTML markup is composed of elements, attributes, and values.
- Write HTML in all lowercase (DOCTYPE is an exception) and surround attribute values with quotes.
- Create file and folder names in all lowercase.
- Separate words with a dash instead of a space or underscore.



Key Takeaways

- Always begin HTML documents with the DOCTYPE declaration so browsers know it's an HTML5 page.
- Content goes in the body element. Instructions for the browser and search engines go in the head.
- Mark up content with semantic HTML.
- Semantic HTML improves accessibility, site efficiency, and ease of maintenance and styling.
- CSS controls presentation of HTML content.
- A browser's style sheet dictates default presentation of HTML. You can overwrite it with CSS.