

Introduction to HyperText Markup Language (HTML)

by Sophia



WHAT'S COVERED

In this lesson, you will learn about the HTML language itself. Additionally, you will learn about the language terminology and how to properly structure HTML tags and elements. Finally, you will be introduced to the purpose and benefits of semantic tags.

Specifically, this lesson will cover the following:

1. Introduction to HTML

1a. Syntax Terminology

1b. Parent–Child Relationships

1c. Semantic Tags

1. Introduction to HTML

HyperText Markup Language (HTML) is the markup language of webpages. HTML provides structure to a webpage that defines elements, such as tables, forms, lists, and headings, and identifies where different portions of our content begin and end. HTML is also responsible for linking to Cascading Style Sheets (CSS) and JavaScript (JS) resources. Webpages are created using HTML to provide structure and meaning to content. This tutorial will introduce you to the syntax, structure, and terminology of the latest version of HTML language, **HTML version 5 (HTML5)**. Note that HTML as a language is generally simply referred to as “HTML”, even though we are working with HTML5.

1a. Syntax Terminology

Recall that syntax is the order, spelling, and grammar in which elements are written. Syntax rules help ensure that languages are always written in a way that can be read or interpreted by a computer or web browser. The smallest HTML object is called a **tag**.

⇒ **EXAMPLE** Tags include a tagname enclosed within angle brackets.

```
<tagname>
```

Tags are like keywords that usually surround content to help tell your browser how to display the content to the user. An **HTML element** is created when you combine an opening, or starting, tag and a closing, or ending, tag.

⇒ **EXAMPLE** The closing tag only includes the tagname preceded with a forward slash.

```
<tagname></tagname>
```



KEY CONCEPT

Some HTML elements have no content. While most tags require both the opening and closing tags to create an HTML element, empty elements without content only require the opening tag. These tags are referred to as “self-closing” because they do not require a closing tag. The `` tag used to display images on a webpage only requires one `` tag. Another example is the `<hr>` horizontal rule tag used for creating a horizontal line across the page.

⇒ **EXAMPLE**

```
<header>
  <img>
  <hr>
</header>
```



REFLECT

Notice that both the `` and `<hr>` tags do not have closing tags. You may see errors in your web browser or layout inconsistencies if you add closing tags to HTML elements that don't require them.

One thing you may see from time to time is self-closing tags with a forward slash at the end of the starting tag. This is a holdover from an older version of HTML called **XHTML**, which required all tags to have some indication that a tag has ended.

⇒ **EXAMPLE** You may still see self-closing tags like the following:

```
<header>
  <img />
  <hr />
</header>
```



REFLECT

Notice that a forward slash was added just before the closing angle bracket of the `<hr>` and `` tag. This is completely optional in HTML4 and HTML5, however it is required if you work on a legacy system that uses any

of the XHTML standards.

HTML contains a library of common tags that serve different purposes and functions, such as the `<p>` paragraph tag used to denote a paragraph of text, or the `` tag used to italicize text for emphasis. Additionally, any content that should receive the effects of the tags will be placed between the starting and ending tags.

🔗 EXAMPLE HTML Code:

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. <em>Ut enim ad minim veniam</em>, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>
```

🔗 EXAMPLE Rendered Output:

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. *Ut enim ad minim veniam*, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.



A comprehensive list of existing HTML tags can be located on W3schools.com's [reference section](#).

🔗 EXAMPLE Tags have default characteristics that can be modified by placing an **attribute** name and setting it equal to a value within the starting tag's brackets.

```
<tagname attribute="value"></tagname>
```

🔗 EXAMPLE We can revisit the previous example to add attributes as follows:

```
<header>
  
  <hr width="80%">
</header>
```



In this case, we have added a source (src) and a width attribute to the image tag. We have also added a width attribute to the horizontal rule tag.



KEY CONCEPT

One thing to note about HTML tags is their capitalization. While HTML code is not case sensitive, you need to keep these in mind.

1. By common convention, tag names and attributes are all lowercase. The one notable exception is the `<!DOCTYPE>` tag.
2. Most attribute values are case sensitive. For example, `href="Index.html"` is NOT the same as `href="index.html"`.

Another example is the `<input>` tag. The `<input>` is used to create controls or fields on web forms. This tag can take the form of any type of input control, such as a textbox, checkbox, or button, when you add the “type” attribute and set it equal to the desired type.

Type Attribute	Output
<code><input type="text"></code>	<code><input type="text"></code>
<code><input type="checkbox"></code> <code><input type="checkbox" checked="true"></code>	<code><input type="checkbox"></code> <code><input type="checkbox" checked="true"></code>
<code><input type="button" value="click me!"></code>	<code><input type="button" value="click me!"></code>

While in the past, attributes were used to control some of the formatting for how the element will look, the latest version of HTML has moved away from that practice. The purpose of attributes today is to control what the element is and how it behaves. Formatting and appearance are handled through CSS, which will be discussed later in this unit.



TERMS TO KNOW

HTML version 5 (HTML5)

The latest version of the HyperText Markup Language as of 2008.

Tag

An HTML tag is a component of HTML used to denote the start and end of HTML elements.

HTML Element

An HTML component that is made up of a starting tag, ending tag, and content in between and rendered to the webpage.

XHTML

A variation of HTML that enforces stricter syntax and defines HTML as an XML application requiring all documents to be correct and well-formed markup.

Attribute

Special keywords used inside the starting tag to control the element, its behavior, or the element’s type.

1b. Parent–Child Relationships

HTML is a language that uses nesting to create structure and define **parent–child relationships** between elements and their content. **Nesting** is the process of placing child elements completely within another parent element. For example, an unordered bullet list contains two types of tags—the container tag `` and the list item tag ``. The list item `` tags are nested within the unordered list `` tag. This makes the `` container the parent and its children are the `` elements contained within.

🔗 EXAMPLE Source Code:

```
<ul>
    <li>Item one</li>
    <li>Item two</li>
    <li>Item three</li>
</ul>
```

🔗 EXAMPLE Output:

- Item one
- Item two
- Item three



KEY CONCEPT

It is important that children are nested completely within the parent. You do not want to start a new child within the parent, close the parent, and THEN close the child. All children should be closed before the parent is closed.

🔗 EXAMPLE Improper Nesting:

```
<p>The quick <em>brown fox</p></em>
```

🔗 EXAMPLE Proper Nesting:

```
<p>The quick <em>brown fox</em></p>
```



TERMS TO KNOW

Parent–Child Relationships

A programming concept wherein an element, the child, belongs to another element, the parent.

Nesting

The act of placing an element completely within another element.

1c. Semantic Tags

The latest version of HTML introduced a type of tag called **semantic tags** that are used to group related HTML code together and to provide some meaning to the section of content. For example, the basic paragraph `<p>` tag is non-semantic and only specifies that a section of text is a paragraph. The paragraph does not specify if it is part of the main article, sidebar, navigation, etc. However, the navigation `<nav>` tag would be used to

surround all of the navigation content. Not only does this tell the browser that the content within is the navigation menu, but this also informs the developer that all code within the `<nav>` tags is related to the navigation menu.

🔗 **EXAMPLE** The following is a list of semantic tags that are commonly used to define different parts of a webpage.

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

🔗 **EXAMPLE** A visual sample of the semantic tags in use:

<header>	
<nav>	
<section>	<aside>
<article>	
<footer>	

In addition to helping on the development side, proper use of semantic tags can have a positive impact on search engine optimization (SEO) scores as well as improve the site's accessibility.



In this video, you will review how to use semantic HTML tags.



TERM TO KNOW

Semantic Tags

A special set of tags that define and give meaning to the different parts of a webpage.



SUMMARY

In this lesson, you were **introduced to the HTML** language and learned about its **syntax terminology**. Additionally, you learned about how **HTML structures** are organized. Finally, you learned about the benefits of using **semantic tags** to define different sections of your webpages.

Source: This Tutorial has been adapted from "The Missing Link: An Introduction to Web Development and Programming " by Michael Mendez. Access for free at <https://open.umn.edu/opentextbooks/textbooks/the-missing-link-an-introduction-to-web-development-and-programming>. License: **Creative Commons attribution: CC BY-NC-SA**.



TERMS TO KNOW

Attribute

Special keywords used inside the starting tag to control the element, its behavior, or the element's type.

HTML Element

An HTML component that is made up of a starting tag, ending tag, and content in between and rendered to the webpage.

HTML version 5 (HTML5)

The latest version of the HyperText Markup Language as of 2008.

Nesting

The act of placing an element completely within another element.

Parent–Child Relationships

A programming concept wherein an element, the child, belongs to another element, the parent.

Semantic Tags

A special set of tags that define and give meaning to the different parts of a webpage.

Tag

An HTML tag is a component of HTML used to denote the start and end of HTML elements.

XHTML

A variation of HTML that enforces stricter syntax and defines HTML as an XML application requiring all documents to be correct and well-formed markup.