Common HTML Elements

**Visible Content**

In the last unit, you learned about the minimum amount of HTML code required to successfully structure a web page, sometimes referred to as "boilerplate code."

We added a declaration, a head, a title, and a body, but we still need content that the web browser can display. In this unit, we'll learn how to use some of the most common HTML elements that add content to web pages.

Keep in mind that HTML was originally created to markup, or present, text. The first few units of this course will focus solely on how HTML can be used to mark up *text*. Later in the course, we'll dive deeper into more advanced styling techniques using CSS.

Let's get started!

**Headings**

Headings in HTML can be likened to headings in other types of media. For example, in newspapers, large headings are typically used to capture a reader's attention. Other times, headings are used to describe content, like the title of a movie or an educational article.

HTML follows a similar pattern. In HTML, there are six different *headings*, or *heading elements*. Headings can be used for a variety of purposes, like titling sections, articles, or other forms of content.

The following is the list of heading elements available in HTML. They are ordered from largest to smallest in size.

1. <h1> - used for main headings, all other smaller headings are used for subheadings.
2. <h2>
3. <h3>
4. <h4>
5. <h5>
6. <h6>

The following example code uses a headline intended to capture a reader's attention. It uses the largest heading available, the main heading element:

<h1>BREAKING NEWS</h1>

# Paragraphs

Often times, headings are meant to emphasize or enlarge only a few words.

If you want to add content in paragraph format, you can add a paragraph using the paragraph element <p>.

<p>The Nile River is the longest river in the world, measuring over 6,850 kilometers long (approximately 4,260 miles). It flows through eleven countries, including Tanzania, Uganda, Rwanda, Burundi, Congo-Kinshasa, Kenya, Ethiopia, Eritrea, South Sudan, Sudan, and Egypt.</p>

Paragraphs are great for expanding the amount of content (text) on your web page. As you begin to add more text to your web page, however, keep in mind that large amounts of text in paragraph format can overwhelm web page visitors. For example, if multiple paragraphs on your web page each contain large amounts of text, your web page could become difficult to consume.

# Unordered Lists

Often times, it's better to display certain types of content in an easy-to-read list.

In HTML, you can use the unordered list for text you decide to format in bullet points. An unordered list outlines individual list items with a bullet point. You've probably used an unordered list when writing down a grocery list or school supplies list.

To create a unordered list using HTML, you can use the <ul> element. The <ul> element, however, cannot hold raw text and cannot automatically format raw text with bullet points. Individual list items must be added to the unordered list using the <li> element.

<ul>

<li>Limes</li>

<li>Tortillas</li>

<li>Chicken</li>

</ul>

In the example above, the list was created using the <ul> element and all individual list items were added using <li> elements.

# Ordered Lists

Great job! Some lists, however, will require a bit more structure. HTML provides the ordered list when you need the extra ordering that unordered lists don't provide.

Ordered lists are like unordered lists, except that each list item is numbered. You can create the ordered list with the <ol> element and then add individual list items to the list using <li> elements.

<ol>

<li>Preheat the oven to 350 degrees.</li>

<li>Mix whole wheat flour, baking soda, and salt.</li>

<li>Cream the butter, sugar in separate bowl.</li>

<li>Add eggs and vanilla extract to bowl.</li>

</ol>

**Links**

You're off to a great start! So far, you've learned how to add headings, paragraphs, and lists to a web page. We wouldn't be taking advantage of the full power of HTML (and the Internet), however, if we didn't *link* to other web pages.

You can add links to a web page by adding an anchor element <a> and including the text of the link in between the opening and closing tags.

<a>This Is A Link To Wikipedia</a>

Wait a minute! Technically, the link in the example above is incomplete. How exactly is the link above supposed to work if there is no [URL](https://www.codecademy.com/courses/learn-html-css/lessons/common-elements/exercises/links?action=lesson_resume) that will lead users to the actual Wikipedia page?

The anchor element in the example above is incomplete without the href *attribute*.

Attributes provide even more information about an element's content. They live directly inside of the opening tag of an element. Attributes are made up of the following two parts:

1. The *name* of the attribute.
2. The *value* of the attribute.

For anchor elements, the name of the attribute is href and its value must be set to the URL of the page you'd like the user to visit.

<a href="https://www.wikipedia.org/">This Is A Link To Wikipedia</a>

In the example above, the href attribute has been set to the value of the correct URL https://www.wikipedia.org/. The example now shows the correct use of an anchor element.

**Note:** When reading technical documentation, you may come across the term *hyperlink*. Not to worry, this is simply the technical term for link and, often times, these terms are used interchangeably.

# More Link Attributes

Have you ever clicked on a link and observed the resulting web page open in a new browser window? If so, you can thank the anchor element's target attribute.

The target attribute specifies that a link should open in a new window. Why is it beneficial to open links in a new window?

It's possible that one or more links on your web page link to an entirely different website. In that case, you may want users to read the linked website, but hope that they return to your web page. This is exactly when the target attribute is useful!

For a link to open in a new window, the target attribute requires a value of \_blank. The target attribute can be added directly to the opening tag of the anchor element, just like the href attribute.

<a href="https://en.wikipedia.org/wiki/Brown\_bear" target="\_blank">The Brown Bear</a>

In the example above, the link would read The Brown Bear and open up the relevant Wikipedia page in a new window.

**Note:** In this exercise, we've used the terminology "open in a new window." It's highly likely that you are using a modern browser that opens up websites in new tabs, rather than new windows. Before the advent of browsers with tabs, additional browser windows had to be opened to view more websites. The target attribute, when used in modern browsers, will open new websites in a new tab.

# Images

All of the elements you've learned about so far (headings, paragraphs, lists, and links) all share one thing in common: they're composed entirely of text! What if you want to add content to your web page that isn't composed of text, like images?

The <img> element lets you add images to a web page. This element is special because it does not have a closing tag, it only has an opening tag. This is because the <img> element is a self-closing element.

<img src="https://www.example.com/picture.jpg" />

Note that the <img> element has a required attribute called src, which is similar to the href attribute in links. In this case, the value of src must be the URL of the image. Also note that the end of the <img> element has a forward slash /. This is required for a self-closing element.

# Alt

Part of being an exceptional web developer is making your site accessible to users of all backgrounds. Specifically, visually impaired users require more support from your web page so that they can experience the content on your page.

HTML helps support visually impaired users with the alt attribute.

The alt attribute is applied specifically to the <img> element. The value of alt should be a description of the image.

<img src="#" alt="A field of yellow sunflowers" />

The alt attributes also serves the following purposes:

1. If an image fails to load on a web page, a user can mouse over the area originally intended for the image and read a brief description of the image. This is made possible by the description you provide in the alt attribute.
2. Visually impaired users often browse the web with the aid of of screen reading software. When you include the alt attribute, the screen reading software can read the image's description outloud to the visually impaired user.

**Note**: If the image on the web page is not one that conveys any meaningful information to a user (visually impaired or otherwise), the alt attribute should not be used.

# Linking At Will

You've probably visited websites where not all links were made up of text. Maybe the links you clicked on were images, or some other form of content.

So far, we've added links that were made up of only text, like the following:

<a href="https://en.wikipedia.org/wiki/Opuntia" target="\_blank">Prickly Pear</a>

Text-only links, however, would significantly decrease your flexibility as a web developer!

Thankfully, HTML allows you to turn nearly any element into a link by wrapping that element with an anchor element. With this technique, it's possible to turn images into links by simply wrapping the <img> element with an <a> element.

<a href="https://en.wikipedia.org/wiki/Opuntia" target="\_blank"><img src="#" alt="A red prickly pear fruit"/></a>

In the example above, an image of a prickly pear has been turned into a link by wrapping the outside of the <img> element with an <a> element.

# White Space

It's important to understand that the formatting of the code in **index.html** will not affect the positioning of the elements within the browser.

For example, if you wanted to increase the space between a paragraph and an image on your web page, you would not be able to accomplish this by simply adding more spacing between the paragraph element and image element within **index.html**. This is because the browser ignores whitespace present in HTML files like **index.html**.

On the other hand, using whitespace in HTML files is important so that your code is easy to read and follow.

For example, the following code is difficult to read and understand:

<html><head></head>

<body><h1>Hello!</h1><p>This is a paragraph!</body>

</html>

The following code, however, uses spacing effectively to make it easier to read:

<html>

<head>

<title>My Web Page</title>

</head>

<body>

<h1>Hello!</h1>

<p>This is a paragraph!</p>

</body>

</html>

# Line Breaks

You saw how modifying the spacing between code in an HTML file doesn't affect the positioning of elements in the browser. If you are interested in modifying the spacing in the browser, you can use HTML's line break element: <br />.

The line break element is one self-closing tag. You can use it anywhere within your HTML code and a line break will be shown in the browser.

Shall I compare thee to a summer's day?<br />Thou art more lovely and more temperate

The code in the example above will result in an output that looks like the following:

Shall I compare thee to a summer's day?

Thou art more lovely and more temperate

**Note**: Line breaks are not the standard way of manipulating the positioning of HTML elements, but it's likely that you'll come across them every now and then. In later units, you'll learn more advanced techniques for positioning HTML elements.

# Indentation

Whitespace makes code easier to read by increasing (or decreasing) the spacing between lines of code. To make the structure of code easier to read, web developers often use indentation.

The World Wide Web Consortium ([W3C](https://en.wikipedia.org/wiki/World_Wide_Web_Consortium)) is responsible for maintaining the style standards of HTML. At the time of writing, the W3C recommends 2 spaces of indentation when writing HTML code. Indentation is intended for elements nested within other elements.

<ul>

<li>Violin</li>

<li>Viola</li>

<li>Cello</li>

<li>Bass</li>

<ul>

In the example above, the list items are indented with two spaces. The spaces are inserted using the spacebar on your keyboard. Unless your [text editor](https://en.wikipedia.org/wiki/Text_editor) has been configured properly, the "TAB" key on your keyboard should not be used for indentation.

**Comments**

HTML files also allow you to add *comments* to your code.

Comments begin with <!-- and end with -->. Any characters in between will be treated as a comment.

<!-- This is a comment that the browser will not display. -->

Including comments in your code is helpful for many reasons:

1. They help you (and others) understand your code if you decide to come back and review it at a much later date.
2. They allow you to experiment with new code, without having to delete old code.

<!-- Favorite Films Section -->

<p>The following is a list of my favorite films:</p>

In the example above, the comment is used to denote that the following text makes up a particular section of the page.

<!-- <a href="#" target="\_blank>Codecademy</a> -->

In the example above, a valid HTML element (an anchor element) has been "commented out." This practice is useful when you want to experiment with new code without having to delete old code.