**Headings:** Headings are block-level elements, and they come in six different rankings, <h1> through <h6>. Headings help to quickly break up content and establish hierarchy, and they are key identifiers for users reading a page. They also help search engines to index and determine the content on a page.

### Paragraphs: Headings are often followed by supporting paragraphs. Paragraphs are defined using the <p> block-level element. Paragraphs can appear one after the other, adding information to a page as desired.

### Divs: Sección generic que tiende a ser más grande que un párrafo y es el contenido relacionado que te gustaría agrupar.

### Bold Text with Strong: To make text bold and place a strong importance on it, we’ll use the <strong> inline-level element. There are two elements that will bold text for us: the <strong> and <b> elements.

### Italicize Text with Emphasis: To italicize text, thereby placing emphasis on it, we’ll use the <em> inline-level element. he other option, the <i> element, is used semantically to convey text in an alternative voice or tone, almost as if it were placed in quotation marks.

## Building Structure: HTML5 introduced new [structurally based elements](https://dev.opera.com/articles/new-structural-elements-in-html5/), including the <header>, <nav>, <article>, <section>, <aside>, and <footer> elements.

### Navigation: The <nav> element identifies a section of major navigational links on a page. The <nav> element should be reserved for primary navigation sections only, such as global navigation, a table of contents, previous/next links, or other noteworthy groups of navigational links.

### Article: The <article> element is used to identify a section of independent, self-contained content that may be independently distributed or reused. We’ll often use the <article> element to mark up blog posts, newspaper articles, user-submitted content, and the like.

### Section: The <section> element is used to identify a thematic grouping of content, which generally, but not always, includes a heading.

### Aside: The <aside> element holds content, such as sidebars, inserts, or brief explanations, that is tangentially related to the content surrounding it. When used within an <article> element, for example, the <aside> element may identify content related to the author of the article.

### Footer: The <footer> element identifies the closing or end of a page, article, section, or other segment of a page. Generally the <footer> element is found at the bottom of its parent. Content within the <footer> element should be relative information and should not diverge from the document or section it is included within.

### Creating Hyperlinks: yperlinks are established using the anchor, <a>, inline-level element. In order to create a link from one page (or resource) to another, the href attribute, known as a hyperlink reference, is required. The href attribute value identifies the destination of the link.

### <a href="http://shayhowe.com">Shay</a>

### Relative & Absolute Paths: Links pointing to other pages of the same website will have a relative path, which does not include the domain (.com, .org, .edu, etc.) in the href attribute value. Because the link is pointing to another page on the same website, the href attribute value needs to include only the filename of the page being linked to: about.html, for example. Here clicking on the text “About” will open the about.html page inside our browser. Clicking the text “Google,” on the other hand, will open <http://google.com/> within our browser. Osea, que uno abre una pàgina de nuestra página web (para enviarte a otra sección…) y el otro abre paginas externas.

### Linking to an Email Address: To create an [email link](https://yoast.com/guide-mailto-links/), the href attribute value needs to start with mailto: followed by the email address to which the email should be sent. To create an email link to shay@awesome.com, for example, the href attribute value would be mailto:shay@awesome.com.

Opening Links in a New Window: One feature available with hyperlinks is the ability to determine where a link opens when clicked. Typically, links open in the same window from which they are clicked; however, links may also be opened in new windows. To trigger the action of opening a link in a new window, use the target attribute with a value of \_blank. The target attribute determines exactly where the link will be displayed, and the \_blank value specifies a new window. To open http://shayhowe.com/ in a new window, the code would look like this: <a href="http://shayhowe.com/" target="\_blank">Shay Howe</a>

Para abrir en la misma pagina, en vez de blank seria self

### Linking to Parts of the Same Page: Periodically we’ll see hyperlinks that link to part of the same page the link appears on. A common example of these same-page links are “Back to top” links that return a user to the top of a page. We can create an on-page link by first setting an id attribute on the element we wish to link to, then using the value of that id attribute within an anchor element’s href attribute. Using the “Back to top” link as an example, we can place an id attribute value of top on the <body> element. Now we can create an anchor element with an href attribute value of #top, pound sign and all, to link to the beginning of the <body> element. Our code for this same-page link would look like the following:

<body id="top">

...

<a href="#top">Back to top</a>

...

</body>

**Line break** : is simply a br, stands for line break, and this is a self-closing tag. Now one of the things that's very important that I tell you is when you're typing your code you can put as many line breaks in your code as you want. Enter, enter, enter, enter, enter. But the browser ignores all of it. The browser is not going to put something on a new line, unless it runs out of space or it specifically sees the br tag. <br>



## Adding Images



To add images to a page, we use the <img> inline element. The <img> element is a self-containing, or empty, element, which means that it doesn’t wrap any other content and it exists as a single tag. For the <img> element to work, a src attribute and value must be included to specify the source of the image. The src attribute value is a URL, typically relative to the server where a website is hosted.

In conjunction with the src attribute, the alt (alternative text) attribute, which describes the contents of an image, should be applied. The alt attribute value is picked up by search engines and assistive technologies to help convey the purpose of an image. The alt text will be displayed in place of the image if for some reason the image is not available.

<img src="dog.jpg" alt="A black, brown, and white dog wearing a kerchief">



Figcaption es un texto que le pones a la imagen

### Sizing Images

It is important to identify the size of an image in order to tell the browser how large the image should be before the page even loads; thus the browser can reserve space for the image and render the page faster. There are a few different ways to size images so that they work well on a page. One option is to use the width and height attributes directly within the <img> tag in HTML.

Additionally, images may be sized using the width and height properties in CSS. When both the HTML attributes and CSS properties are used, the CSS attributes will take precedence over the HTML attributes.

Specifying either a width or height will cause the other dimension to adjust automatically to maintain the aspect ratio of the image. As an example, if we want an image to be 200 pixels tall but are less specifically concerned about how wide it is, we can set the height to 200 pixels, and the width of the image will adjust accordingly. Setting both a width and height will work also; however, doing so may break the aspect ratio of an image, causing it to appear distorted.

img {

height: 200px;

width: 200px;

}

#### Block Positioning Images

Adding the display property to an image and setting its value to block forces the image to be a block-level element. This makes the image appear on its own line, allowing the surrounding content to be positioned above and below the image.

img {

display: block;

}

#### Positioning Images Flush Left or Right

Sometimes displaying an image as inline or block, or perhaps even inline-block, isn’t ideal. We may want the image to appear on the left or right side of its containing element, while all of the other content wraps around the image as necessary. To do this, we use the float property with a value of either left or right.

Remembering back to Lesson 5, “[Positioning Content](https://learn.shayhowe.com/html-css/positioning-content/),” we recall that the float property was originally intended to position images to the left or right of a containing element. Now we’ll use it for that original purpose.

Floating an image is a start; however, all other content will align directly against it. To provide spacing around an image, we’ll use the margin property. Additionally, we can use the padding, border, and background properties to build a frame for the image, if desired.

img {

background: #eaeaed;

border: 1px solid #9799a7;

float: right;

margin: 8px 0 0 20px;

padding: 4px;

}

## Adding Audio

<https://learn.shayhowe.com/html-css/adding-media/>

### <https://learn.shayhowe.com/html-css/organizing-data-with-tables/>

### <https://learn.shayhowe.com/html-css/>

### <https://www.wpkube.com/html5-cheat-sheet/>

## Unordered Lists: An unordered list is simply a list of related items whose order does not matter. Creating an unordered list in HTML is accomplished using the unordered list block-level element, <ul>. Each item within an unordered list is individually marked up using the list item element, <li>.

<ul>

<li>Orange</li>

<li>Green</li>

<li>Blue</li>

</ul>

## Ordered Lists

The ordered list element, <ol>, works very much like the unordered list element; individual list items are created in the same manner. The main difference between an ordered list and an unordered list is that with an ordered list, the order in which items are presented is important.

<ol>

<li>Head north on N Halsted St</li>

<li>Turn right on W Diversey Pkwy</li>

<li>Turn left on N Orchard St</li>

</ol>

### Start Attribute

The start attribute defines the number from which an ordered list should start. By default, ordered lists start at 1. However, there may be cases where a list should start at 30 or another number. When we use the start attribute on the <ol> element, we can identify exactly which number an ordered list should begin counting from.

The start attribute accepts only integer values, even though ordered lists may use different numbering systems, such as roman numerals.

<ol start="30">

<li>Head north on N Halsted St</li>

<li>Turn right on W Diversey Pkwy</li>

<li>Turn left on N Orchard St</li>

</ol>

### Reversed Attribute

The reversed attribute, when used on the <ol> element, allows a list to appear in reverse order. An ordered list of five items numbered 1 to 5 may be reversed and ordered from 5 to 1.

The reversed attribute is a Boolean attribute, and as such it doesn’t accept any value. It is either true or false. False is the default value; the value becomes true when the attribute name reversed appears on the <ol> element.

<ol reversed>

<li>Head north on N Halsted St</li>

<li>Turn right on W Diversey Pkwy</li>

<li>Turn left on N Orchard St</li>

</ol>

### Value Attribute

The value attribute may be used on an individual <li> element within an ordered list to change its value within the list. The number of any list item appearing below a list item with a value attribute will be recalculated accordingly.

As an example, if the second list item has a value attribute value of 9, the number on that list item marker will appear as if it is the ninth item. All subsequent list items will be numbered upwards from 9.

|  |
| --- |
| <ol>  <li>Head north on N Halsted St</li>  <li value="9">Turn right on W Diversey Pkwy</li>  <li>Turn left on N Orchard St</li>  </ol> |

## Description Lists

Another type of list seen online (but not as often as unordered or ordered lists) is the description list. Description lists are used to outline multiple terms and their descriptions, as in a glossary, for example.

Creating a description list in HTML is accomplished using the description list block-level element, <dl>. Instead of using a <li> element to mark up list items, the description list requires two block-level elements: the description term element, <dt>, and the description element, <dd>.

A description list may contain numerous terms and descriptions, one after the other. Additionally, a description list may have multiple terms per description, as well as multiple descriptions per term. A single term may have multiple meanings and warrant multiple descriptions. Conversely, a single description may be suitable for multiple terms.

When adding a description list, the <dt> element must come before the <dd> element. The definition term and the description that directly follows it correspond to one another; thus, the order of these elements is important.

By default, the <dl> element will include vertical margins, just like the <ul> and <ol> elements. Additionally, the <dd> element includes a left margin by default.

<dl>

<dt>study</dt>

<dd>The devotion of time and attention to acquiring knowledge on an academic subject, especially by means of books</dd>

<dt>design</dt>

<dd>A plan or drawing produced to show the look and function or workings of a building, garment, or other object before it is built or made</dd>

<dd>Purpose, planning, or intention that exists or is thought to exist behind an action, fact, or material object</dd>

<dt>business</dt>

<dt>work</dt>

<dd>A person's regular occupation, profession, or trade</dd>

</dl>

## Nesting Lists

One feature that makes lists extremely powerful is their ability to be nested. Every list may be placed within another list; they can be nested continually. But the potential to nest lists indefinitely doesn’t provide free rein to do so. Lists should still be reserved specifically for where they hold the most semantic value.

One trick with nesting lists is to know where to begin and end each list and list item. Speaking specifically about unordered and ordered lists, as that is where most nesting will occur, the only element that may reside directly within the <ul> and <ol> elements is the <li> element. To repeat, the only element we can place as a direct child of the <ul> and <ol> elements is the <li> element.

That said, once inside the <li> element, the standard set of elements may be added, including any <ul> or <ol> elements.

To nest a list rather than closing a list item, begin a new list. Once the nested list is complete and closed, close the wrapping list item and continue on with the original list.

<ol>

<li>Walk the dog</li>

<li>Fold laundry</li>

<li>

Go to the grocery and buy:

<ul>

<li>Milk</li>

<li>Bread</li>

<li>Cheese</li>

</ul>

</li>

<li>Mow the lawn</li>

<li>Make dinner</li>

</ol>

#### List Style Type Values

As previously mentioned, the list-style-type property comes with a handful of different values. The following list outlines these values as well as their corresponding content.

| **List Style Type Value** | **Content** |
| --- | --- |
| none | No list item |
| disc | A filled circle |
| circle | A hollow circle |
| square | A filled square |
| decimal | Decimal numbers |
| decimal-leading-zero | Decimal numbers padded by initial zeros |
| lower-roman | Lowercase roman numerals |
| upper-roman | Uppercase roman numerals |
| lower-greek | Lowercase classical Greek |
| lower-alpha / lower-latin | Lowercase ASCII letters |
| upper-alpha / upper-latin | Uppercase ASCII letters |
| armenian | Traditional Armenian numbering |
| georgian | Traditional Georgian numbering |

ul {

list-style-type: square;

}

### Using an Image as a List Item Marker

There may come a time when the default list-style-type property values are not enough, and we want to customize our own list item marker. Doing so is most commonly accomplished by placing a background image on each <li> element within a list.

The process includes removing any default list-style-type property value and adding a background image and padding to the <li> element.

In detail, the list-style-type property value of none will remove existing list item markers. The background property will identify a background image, along with its position and repeat value, if necessary. And the padding property will provide space to the left of the text for the background image.

And many others in <https://learn.shayhowe.com/html-css/creating-lists/>

li {

background: url("arrow.png") 0 50% no-repeat;

list-style-type: none;

padding-left: 12px;

}