# Common HTML Elements

There are currently about 150 HTML element types used on the web, but to be successful in eLearning work, there are only 20 or so that you really need to know. Here are the most important ones with brief descriptions.

|  |  |  |  |
| --- | --- | --- | --- |
| Tag | Name | Description | Key Attributes |
| Document-Level Tags |
| <head> | Header (Machine) | This header is invisible to users, includes metadata, loads necessary formatting for the page, and does other things behind the scenes. |  |
| <title> | Title | This is used in the header to set the title of the page to be shown in the browser tab. |  |
| <link> | Linked Files | This is used in the header to identify linked stylesheets that are shared between multiple pages. | src="[link to the file]" |
| <script> | JavaScript | This is used in the header or in the body to include JavaScript code or to include an external Javascript file. | src="[link to the file]" |
| <body> | Body | This is the main content area of a page. |  |
| Content-Level Tags |
| <article> | Article | Within the body, this is the main organizer for content. For instance, if you have a list of blog posts, each post would be a separate article. |  |
| <h1><h2><h3><h4><h5><h6> | Headings | Headings are used to designate titles and subtitles in a page, with H1 being the highest level (e.g., the title of the page) and H6 being the lowest (e.g., a section title nested deeply below other sections). |  |
| <a> | Link | A hyperlink to another page or section within a page. | href="[link to the target]" |
| <nav> | Navigation | A navigation menu or area. |  |
| <p> | Paragraph | A paragraph of content. |  |
| <br /> | Break Return | A hard break in content. This should typically only be used if you need to separate content within a paragraph, such as with a poem. Notice also that the tag is self-closing, so no content should be inside it. |  |
| <table> | Table | A content table (see Tables below). |  |
| <form> | Form | A form for accepting input from users. | method="POST" or "GET"action="[URL]" |
| <input> | Input | Accepts user-submitted content for a form (e.g., a textfield). | value="[value]" |
| <button> | Button | An interactive element (typically used with JavaScript) to change the page in some way. |  |
| <blockquote> | Blockquote | An indented paragraph of content. |  |
| <em> | Emphasis or Italic | Italicizes content, making it slanted. |  |
| <strong> | Strong or Bold | Bolds content, making it darker. |  |
| <img> | Image | Embeds an image file. | src="[link to the file]"alt="[description]" |
| <audio> | Audio | Embeds an audio file. | src="[link to the file]"alt="[description]" |
| <video> | Video | Embeds a video file. | src="[link to the file]"alt="[description]" |
| <ul><ol> | List | A list of content, either Unordered (using bullets) or Ordered (using numbers). |  |
| <sup><sub> | SuperscriptSubscript | These are used for exponents and scientific notation, such as E=mc2 (superscript) and H2O (subscript). |  |

## Tables

Tables are great for organizing and visualizing content (especially data), but they use a number of specific tags in specific orders, and there are some basic rules you must follow to make sure that tables are accessible and rendered properly. In addition, tables **should not be used simply to visually layout content** (e.g., to place two images side by side). Instead, you should use CSS properties for strictly visual manipulation.

The tags you should know for tables include the following:

|  |  |  |
| --- | --- | --- |
| Tag | Name | Description |
| <table> | Table | This is the enclosing tag for the entire table. |
| <thead> | Header | This encloses all of the header content for the table, such as labels for columns. |
| <tbody> | Body | This encloses the body content of the table. |
| <tr> | Row | This encloses a row of cells. |
| <th> | Cell Heading | This is a heading cell. |
| <td> | Cell Content | This is a content cell. |

Tables are laid out top to bottom, and then rows are laid out left to right. All tables should follow this basic structure:

<table>  <caption>Here's the optional caption.</caption>  <thead>    <tr>      <th></th>      <th>Column A</th>      <th>Column B</th>    </tr>  </thead>  <tbody>    <tr>      <th>Row 1</th>      <td>Content 1A</td>      <td>Content 1B</td>    </tr>    <tr>      <th>Row 2</th>      <td>Content 2A</td>      <td>Content 2B</td>    </tr>  </tbody></table>

This would render as follows:

|  |  |  |
| --- | --- | --- |
|  | Column A | Column B |
| Row 1 | Content 1A | Content 1B |
| Row 2 | Content 2A | Content 2B |
|

Visually, tables are laid out as follows:

Figure 1

Visual layout of a table from top-to-bottom and left-to-right



Figure 2

Visual layout of a table with tags



Here are a few things to notice:

1. The caption is included first but is actually rendered last. The position and style of the caption (first or last) depends on your CSS settings.
2. The first <th> element is intentionally left empty or blank but must be included so that the columns line up properly.
3. Instead of manually bolding or italicizing headers, we use the semantic <th> identifier instead of <td>. This communicates to indexers and screen readers that the heading applies to a column or row.

## Lists

Lists all follow the same structure of defining the list with <ul> or <ol> tags and then including <li> tags for each list item within them. Here's an example:

<ul>  <li>Apple</li>  <li>Banana</li>  <li>Orange</li></ul>

This would render as follows:

* Apple
* Banana
* Orange

Changing the encapsulating <ul> to <ol> would make it numbered instead of bulleted, as follows:

1. Apple
2. Banana
3. Orange

To indent (or nest) lists, you simply include new lists within list items, as follows:

<ol>  <li>Fruits    <ol>      <li>Apple</li>      <li>Banana</li>    </ol>  </li>  <li>Vegetables    <ol>      <li>Carrot</li>      <li>Broccoli</li>    </ol>  </li></ol>

This would render as follows:

1. Fruits
	1. Apple
	2. Banana
2. Vegetables
	1. Carrot
	2. Broccoli

Following this pattern, you can make infinitely nested lists as necessary. However, you should avoid including other tags within lists, such as <p>, <img>, etc. as they can quickly confuse lists and influence their structure and rendering.

## Things to Avoid

Just as there are common HTML elements that everyone should know and use, there are also some that are typically misused. Here are some of the most important to be aware of with explanations.

|  |  |  |
| --- | --- | --- |
| Tag | Name | Description |
| <iframe> | IFrame | This embeds one page within another and is often used to embed external resources. It can be used properly, but it is often a security risk and poses problems for browsers because you cannot control the content within the frame. |
| <table>\*\* for layouts | Table | Tables are great for data, but sometimes people use them for laying out content (e.g., creating rows and columns of images). This can pose serious accessibility problems and should be avoided. |
| <span> | Span | Spans have no semantic meaning but basically allow you to apply formatting to a chunk of content. They can be useful for visually adjusting content, but screen readers and other machine applications have difficulty making sense of them. They are also often used frequently for no reason, which makes the HTML messy and bloated. |
| <div> | Division Layer | Similar to spans, divs have no semantic meaning, and though they can (and should) be used effectively in web development, most of the time eLearning and content developers don't need to use them. |
| <i> | Italic | The <em> tag is preferred to the <i> tag because it is more semantic. They appear the same to the reader. |
| <b> | Bold | The <strong> tag is preferred to the <b> because it is more semantic.They appear the same to the reader. |
| style="" | Style Attribute | Editing the style attribute of an HTML element to change its formatting is not always bad, but if you are changing style attributes across multiple elements in the same way (e.g., making the text larger in multiple paragraph elements), then you should instead rely upon CSS. |

Read this online at <https://roycekimmons.com/elearning_hacker/common_html_elements>