

# Introduction to HTML : Elements and Structure

## HTML Element Definition and Syntax

An HTML element is a piece of content in an HTML document and uses the following syntax: opening tag + content + closing tag. In the code provided, `<p>` is the opening tag, `Hello World!` is the content, and `</p>` is the closing tag.

```
<p>Hello World!</p>
```

## HTML Tag Syntax

The syntax for a single HTML tag is an opening angle bracket (`<`) followed by the element name and closed with the closing angle bracket (`>`). The above code block is an example of a single opening `<div>` tag.

```
<div>
```

## Element Content

The content of an HTML element is the information between the opening and closing tags of an element. In the example, the element content is "Codecademy is awesome!".

```
<h1>Codecademy is awesome!</h1>
```

## Closing Tag

An HTML closing tag is used to denote the end of an HTML element. The syntax for a closing tag is a left angle bracket `<` followed by a forward slash `/` then the element name and a right angle bracket to close `>`.

```
<body>  
...  
</body>
```

## Unique ID Attributes

In HTML, specific and unique `id` attributes can be assigned to different elements in order to differentiate between them.

When needed, the `id` value can be called upon by CSS and JavaScript to manipulate, format, and perform specific instructions on that element and that element only. Valid `id` attributes should begin with a letter and should only contain letters (`a-z`), digits (`0-9`), hyphens (`-`), underscores (`_`), and periods (`.`).

```
<h1 id="A1">Hello World</h1>
```

## <body> Body Element

The `<body>` element represents the content of an HTML document. Content inside `<body>` tags are rendered on the web browsers.

```
<body>
  <h1>Learn to code with Codecademy :)</h1>
</body>
```

## HTML Structure

HTML is organized into a family tree structure. HTML elements can have parents, grandparents, siblings, children, grandchildren, etc.

```
<body>
  <div>
    <h1>A grandchild of body and a child of div</h1>
    <h2>A sibling of h1</h2>
  </div>
</body>
```

## <h1> <h2> <h3> <h4> <h5> <h6> Headings

HTML can use six different levels of heading elements. The heading elements are ordered from the highest level `<h1>` to the lowest level `<h6>`.

```
<h1>Breaking News</h1>
<h2>This is the first subheading</h2>
<h3>This is the second subheading</h3>
...
<h6>This is the fifth subheading</h6>
```

## <div> Div Element

The `<div>` element is used as a container that divides an HTML document into sections and is short for "division". `<div>` elements can contain *flow content* such as headings, paragraphs, links, images, etc.

```
<div>
  <h1>This is a section of grouped elements</h1>
  <p>Here's some text for the first section</p>
</div>
<div>
  <h1>This is a second section</h1>
  <p>Here's some text for the second section</p>
</div>
```

## <ul> Unordered List Element

The `<ul>` unordered list element is used to create a list of items in no particular order. Each individual list item will have a bullet point by default. For example:

- Oculus Quest
- Fender Telecaster
- Record Player

```
<ul>
  <li>Oculus Quest</li>
  <li>Fender Telecaster</li>
  <li>Record Player</li>
</ul>
```

## <p> Paragraph Element

Paragraph elements, <p>, contain and display blocks of text.

```
<p>This is a block of text! Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
```

## <span> Element

The <span> element is an inline container for text and can be used to group text for styling purposes. However, as <span> is a generic container to separate pieces of text from a larger body of text, its use should be avoided if a more semantic element is available.

```
<p><span>This text</span> may be styled differently than the surrounding text.</p>
```

## <em> Emphasis Element

The <em> emphasis element emphasizes text and browsers will usually *italicize* the emphasized text by default.

```
<p>This <em>word</em> will be emphasized in italics.</p>
```

## <strong> Strong Element

The <strong> element highlights important, serious, or urgent text and browsers will normally render this highlighted text in **bold** by default.

```
<p>This is <strong>important</strong> text!</p>
```

## <br> Line Break Element

The <br> line break element will create a line break in text and is especially useful where a division of text is required, like in a postal address. The line break element requires only an opening tag and must not have a closing tag.

```
<p>A line break haiku. <br>Poems are a great use case. <br>Oh joy! A line break.</p>
```

## <li> List Item Element

The <li> list item element create list items inside:

- *Ordered lists* <ol>
- *Unordered lists* <ul>

```
<ol>  
  <li>one</li>  
  <li>two</li>  
</ol>  
<ul>  
  <li>list item</li>  
  <li>list item</li>  
</ul>
```

## <img> Image Element

HTML image `<img>` elements embed images in documents. The `src` attribute contains the image URL and is mandatory. `<img>` is an *empty element* meaning it should not have a closing tag.

### alt Attribute

An `<img>` element can have alternative text via the `alt` attribute. The alternative text will be displayed if an image fails to render due to an incorrect URL, if the image format is not supported by the browser, if the image is blocked from being displayed, or if the image has not been received from the URL.

## HTML Attributes

HTML attributes are values added to the opening tag of an element to configure the element or change the element's default behavior. In the provided example, we are giving the `<p>` (paragraph) element a unique identifier using the `id` attribute and changing the color of the default text using the `style` attribute.

## Attribute Name and Values

HTML attributes consist of a name and a value using the following syntax: `name="value"` and can be added to the opening tag of an HTML element to configure or change the behavior of the element.

## <ol> Ordered List Elements

HTML ordered list, `<ol>`, elements are used to create lists of items with a sequential order. Each list item in an ordered list appears numbered by default. For example:

1. Code every day.
2. Build a portfolio.
3. Never give up.

```

```

```

```

```
<p id="my-paragraph" style="color: green;">Here's some text for a paragraph that is being altered by HTML attributes</p>
```

```
<elementName name="value"></elementName>
```

```
<ol>
  <li>First Item</li>
  <li>Second Item</li>
  <li>Third Item</li>
</ol>
```

## `<title>` Title Element

The `<title>` element contains a text that defines the title of an HTML document. The title is displayed in the browser's title bar or tab in which the HTML page is displayed. The `<title>` element can only be contained inside a document's `<head>` element.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Title of the HTML page</title>
  </head>
</html>
```

## `<a>` Anchor Element

The `<a>` anchor element is used to create hyperlinks in an HTML document. The hyperlinks can point to other webpages, files on the same server, a location on the same page, or any other URL via the hyperlink reference attribute, `href`. The `href` determines the location the anchor element points to.

```
<!-- Hyperlink to Codecademy -->
<a href="https://www.codecademy.com">Codecademy</a>
```

## `<target>` Attribute

The `target` attribute on an anchor (`<a>`) element specifies where a hyperlink should be opened. For example, a `target` value of `"_blank"` will tell the browser to open the hyperlink in a new tab in modern browsers, or in a new window in older browsers or if the browser has had settings changed to open hyperlinks in a new window.

```
<a href="https://www.google.com" target="_blank">This
anchor element links to google and will open in a new
tab or window.</a>
```

## `<video>` Video Element

The `<video>` element embeds a media player into an HTML document for video playback. The `src` attribute will contain the URL to the video. Adding the `controls` attribute will display video controls in the media player.

The `<video>` element can contain text content between its opening and closing tags which will be displayed if the browser does not support the video format or if the video fails to load.

```
<video src="test-video.mp4" controls>
  Video not supported
</video>
```

## `<html>` Element

The `<html>` element, the root of an HTML document, should be added after the `!DOCTYPE` declaration. All content/structure for an HTML document should be contained between the opening and closing `<html>` tags.

```
<!DOCTYPE html>
<html>
  <!-- I'm a comment -->
</html>
```

## File Path

URL paths in HTML can be absolute paths, like a full URL, for example:

```
https://developer.mozilla.org/en-
```

```
US/docs/Learn
```

 or a relative file path that links to a local file in the same folder or on the same server, for example: `./style.css`. Relative file paths begin with `./` followed by a path to the local file. `./` tells the browser to look for the file path from the current folder.

```
<a href="https://developer.mozilla.org/en-US/docs/Web">The URL for this anchor element is an absolute file path.</a>
<a href="./about.html">The URL for this anchor element is a relative file path.</a>
```

## # Linking to Spot on the Same Page

The HTML anchor element, `<a>` can create hyperlinks to different parts of the same HTML document using the `href` attribute to point to the desired location with `#` followed by the `id` of the element to link to.

```
<div>
  <p id="id-of-element-to-link-to">A different part of the page!</p>
</div>

<a href="#id-of-element-to-link-to">Take me to a different part of the page</a>
```

## Whitespace

Whitespace, such as line breaks, added to an HTML document between block-level elements will generally be ignored by the browser and are not added to increase spacing on the rendered HTML page. Rather, whitespace is added for organization and easier reading of the HTML document itself.

```
<p>Test paragraph</p>

<!-- The whitespace created by this line, and above/below this line is ignored by the browser-->

<p>Another test paragraph, this will sit right under the first paragraph, no extra space between.</p>
```

## Indenting Nested in HTML

HTML code should be formatted such that the indentation level of text increases once for each level of nesting. The W3C standards recommend two spaces of indentation per level of nesting.

```
<div>
  <h1>Heading</h1>

  <ul>
    <li>Item 1</li>
    <li>Item 2</li>
  </ul>
</div>
```

## Indenting Nested in HTML

HTML code should be formatted such that the indentation level of text increases once for each level of nesting. The W3C standards recommend two spaces of indentation per level of nesting.

```
<div>
  <h1>Heading</h1>

  <ul>
    <li>Item 1</li>
    <li>Item 2</li>
  </ul>
</div>
```

## Comments

In HTML, comments can be added between an opening `<!--` and closing `-->`. Content inside of comments will not be rendered by browsers, and are usually used to describe a part of code or provide other details.

Comments can span single or multiple lines.

```
<!-- Main site content -->
<div>Content</div>

<!--
  Comments can be
  multiple lines long.
-->
```

## `<head>` Element

The `<head>` element contains general information about an HTML page that isn't displayed on the page itself. This information is called metadata and includes things like the title of the HTML document and links to stylesheets.

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata is contained in this element-->
  </head>
</html>
```