codecademy

Elements and Structure

HTML

HTML (HyperText Markup Language) is used to give content to a web page and instructs web browsers on how to structure that content.

HTML Element

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.
- `</p>` is the closing tag.

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

HTML Tag

The syntax for a single HTML tag is an opening angle bracket `<` followed by the element name and a closing angle bracket `>`. Here is an example of an opening `<div>` tag.

```
<div>
```

Element Content

The content of an HTML element is the information between the opening and closing tags of an element.

```
<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>
```

<body> Body Element

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

Note: There can be only one `<body>` element in a document.

```
<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>It's div's child and body's grandchild</h1>
    <h2>It's h1's sibling</h2>
  </div>
</body>
```

<h1>-<h6> Heading Elements

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 1st subheading</h2>
<h3>This is the 2nd subheading</h3>
...
<h6>This is the 5th subheading</h6>
```

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>A section of grouped elements</h1>
  <p>Here's some text for the section</p>
</div>
<div>
  <h1>Second section of grouped elements</h1>
  <p>Here's some text</p>
</div>
```

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.

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

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.

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

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>
```

<p> Paragraph Element

The `<p>` paragraph element contains and displays a block of text.

```
<p>This is a block of text! Lorem ipsum dolor sit amet, consectetur adipisicing 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>
```

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 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>
```


 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.

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

`<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.

```
<ul>
  <li>Play more music 🎸</li>
  <li>Read more books 📚</li>
</ul>
```

`<li>` List Item Element

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

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

```
<ol>
  <li>Head east on Prince St</li>
  <li>Turn left on Elizabeth</li>
</ol>

<ul>
  <li>Cookies</li>
  <li>Milk</li>
</ul>
```

`<ol>` Ordered List Element

The `<ol>` ordered list element creates a list of items in sequential order. Each list item appears numbered by default.

```
<ol>
  <li>Preheat oven to 325 F 🍞</li>
  <li>Drop cookie dough 🍪</li>
  <li>Bake for 15 min ⏰</li>
</ol>
```

`<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.

```
<img src="image.png">
```

`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.
The text will be read aloud if screen reading software is used and helps support visually impaired users by providing a text descriptor for the image content on a webpage.

```
<img src="path/to/image" alt="text
describing image" />
```

<video> Video Element

The `<video>` element embeds a media player 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.
Note: The content inside the opening and closing tag is shown as a fallback in browsers that don't support the element.

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

Document Type Declaration

The document type declaration `<!DOCTYPE html>` is required as the first line of an HTML document. The doctype declaration is an instruction to the browser about what type of document to expect and which version of HTML is being used, in this case it's HTML5.

```
<!DOCTYPE html>
```

<html> 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>
```

<head> 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>
```

<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.

```
<!-- Linking a text -->
<a href="codecademy.com">Visit this
site</a>

<!-- Linking an image -->
<a href="codecademy.com">
<img src="logo.jpg">Click this image</a>
```

<target> Target Attribute

The `target` attribute on an `<a>` anchor element specifies where a hyperlink should be opened. 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>
```

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>
```

Link to a Different Part of the Page

The 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>
```

Indentation

HTML code should be formatted such that the indentation level of text increases once for each level of nesting.

It is a common convention to use two or four `space` 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.
-->
```