

# Learn HTML: Structure

## <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.

```
<!-- Creating text links -->
<a href="http://www.codecademy.com">Visit
this site</a>

<!-- Creating image links -->
<a href="http://www.codecademy.com">
  Click this image
</a>
```

## <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>
```

## <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>
```

## 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>
```

## 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>
```

## <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>
```

## 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.
-->
```

## 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>
```

## 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.

### <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>
```

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