Web Design

- HTML
  - Boilerplate
  - HTML elements – The basics
  - More tags

HTML

Boilerplate

In web design, **boilerplate code** or **boilerplate** refers to a section of HTML that must be included on each page with little or no alteration.

the most basic HTML boilerplate might look something like this:

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>My Website</title>
  </head>
  <body>


  </body>
</html>
```

But what does it all mean?

Let's start at the top with `<!DOCTYPE html>`. This piece of code tells the browser that the page it's about to read is written in HTML. the DOCTYPE declaration isn't an HTML tag, it's just information that the browser needs to know.

in older versions of HTML, this piece of code had to be much more complicated.

Next, we have `<html lang="en" dir="ltr">`. This is the first HTML tag. It's an opening tag (the closing tag `</html>` is at the end of the boilerplate). This tag has two **attributes**, `lang` for language which is `en` for English, and `dir` for direction with is `ltr` for "left to right".

The `<html>` tags represent the root of an HTML document. The `<html>` tag is the container for all other HTML elements.

The `<head>` tags are a container for metadata and is opened and closed after the `<html>` opening tag.

Metadata refers to information about an HTML document. The metadata is not shown in the browser viewport.

The document title, character set, styles, scripts, and other meta information are often defined via metadata.

The `<meta charset="utf-8">` tag defines an element of metadata, in this case `charset` for the character set, `utf-8` is the most common character encoding method used on the internet today, and is the default character set for HTML5.

other meta tags that might be included are:

```
<meta name="description" content="Free Web tutorials">
<meta name="keywords" content="HTML, CSS, JavaScript">
<meta name="author" content="John Doe">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

A `<title>` tags is used to set the name of the page, that will appear in the browser tab, or bookmark when saved.

The `<body>` tags are very important. Everything that is between these tags is displayed in the viewport.

HTML elements – The basics

An HTML element is is a piece of content (usually text) that is enclosed within tags. The tag name defines what type of content it is and how it should be displayed.

Some common tags you'll use include:

`<h1></h1>` Defines a Heading

`<p></p>` Defines a paragraph

`<a href="#"></a>` Defines an anchor link

Most HTML elements have an opening, and closing tag, the content sitting between them, for example:

`<h1>This is a heading</h1>`

`<p>This is my first paragraph</p>`

however, there are a few tags that don't have a closing tag. for example:

`<br>` This is used to insert a line break.

`<hr>` This with insert a horizontal rule.

`<img src="https://images.unsplash.com/photo-1596854407944-bf87f6fdd49e" alt="ginger cat">` the `img` tag is used to insert an image.

Attributes

Attributes are often included in an HTML open tag to define more information about the content. Attributes are included using a name and value pair. In the examples below the attribute href for "hypertext reference", defines the location the anchor link should go when the text "visit google" is clicked.

`<a href="https://google.com">visit google</a>`

in this image element:

`<img src="https://images.unsplash.com/photo-1596854407944-bf87f6fdd49e" alt="ginger cat">`

the `scr` or "source" attribute is used to define the location of the images to display. The `alt` attribute defines alternate text for when the image cannot be displayed or viewed. alt attributes are important for the accessibility of visually impaired readers.

another common attribute that is often included in an HTML tag is `class`. `class` attributes make it possible to differentiate different types of content with the same tags and style them accordingly in CSS.

the `id` attribute is used to define a unique identifier for an HTML element. This makes the element easy to select using javascript and can be navigated to using anchor links.

Block vs inline

The browser has two ways to display elements in the viewport.

Block elements always start on a new line and use the full width of the viewport.

Examples of block level elements include:

- `<p>`
- `<ol>`, `<ul>`, `<dl>`
- All headings `h1` – `h6`
- `<article>`, `<section>`, `<div>`

Inline elements display in a line. They do not force the text after them to a new line.

An anchor (or link) is an example of an inline element. You can put several links in a row, and they will display in a line.

Examples of inline elements:

- `<a>`
- `<strong>`, `<em>`, `<b>`, `<i>`, `<q>, <mark>`
- `<span>`

The `img` tag is also an example of an inline element, meaning that unless they are floated (covered later in CSS) they will flow horizontally with text and other inline elements.

Take a look at the codepen below, notice that each of the paragraph tags are presented as a block and the anchor tag stays inline.

Try moving the img tag inside one of the paragraph tags. ***What do you think will happen?***

Did you notice that the image can sit between words in the paragraph?

the default display properties of an element can be changed using CSS styles.

Nesting

Nesting in HTML refers to placing elements inside other elements. for example placing an <img> element inside a <p> element.

when creating a list, the line items (<li> tags) are nested within <ul> or <ol> tags. there are a couple of rule for nesting.

1. You can't open a tag in one element and close it in another, make sure you're closing tags are in the correct order.
2. You shouldn't nest block elements inside inline elements.

```
<a href="/about"><p>The about page</p></a>  <!-- paragraph elements are block elements, and shouldn't be inside an anchor tag -->
<p><a href="/about">The about page</a></p>  <!-- this is correct -->

<p><a href="/about">The about page</p></a>  <!-- the anchor tag is closed after the paragraph, this is wrong -->
```

More tags

Now with an understanding of the fundamental structure of and HTML page, let's look at a few more common tags and their uses.

Text formatting

We previously looked at the <p> tag and its use for defining paragraphs. sometimes we need to highlight one or just a few words within a paragraph. That's where text formatting tags can be useful.

these formatting elements are designed to highlight special types of text:

- `<b>` - Bold text
- `<strong>` - Important text
- `<i>` - Italic text
- `<em>` - Emphasized text
- `<mark>` - Marked text
- `<small>` - Smaller text
- `<del>` - Deleted text
- `<ins>` - Inserted text
- `<sub>` - Subscript text
- `<sup>` - Superscript text

Try adding some of the other formatting tags to the paragraphs in the codepen below.

heading tags

Heading tags are as the name suggests used to define headings. Heading tags range from `<h1>` (most important) to `<h6>` (least important). The block elements so will start on a new line and use the full width of the viewport.

take a look at the code pen below to their default styling. ***Try adding a new heading after heading 6, using your name.***

Lists

The two most common types of lists used in HTML are ordered (or numbered) lists or unordered lists. unordered lists are displayed with a bullet point by default.

A `<ul>` tag is used to define an unordered list and an `<ol>` tag is used to define an ordered list.

items in the list are written within `<li>` tags.

Try adding some new items to lists in the code pen below.

What happens if you add a step between 2 and 3 of the ordered list?

notice how the list numbering will always be in order.

There is another list type called a description list, which list of terms, with a description of each term.

The `<dl>` tag defines the description list, the `<dt>` tag defines the term (name), and the `<dd>` tag describes each term, for examples:

```
<dl>
 <dt>Coffee</dt>
 <dd>- black hot drink</dd>
 <dt>Milk</dt>
 <dd>- white cold drink</dd>
</dl>
```

Try copying the code above into the previous codepen to see the results.