

Web Development

Web Development consists of three main things: HTML, CSS and JavaScript. **HTML** is the basic building blocks of a website which describes things on a webpage. For example, bricks in a building. **CSS** is for styling and looks of elements in a webpage. For example, paint on the walls of a building. **JavaScript** is actions that can be performed on a webpage. For example, switching on and off a ceiling fan in a room of a building.

HTML

- **HTML** stands for **HyperText Markup Language**.
- **HyperText** means some texts in a webpage are linked to other webpages in a website. These texts are called **hyperlinks** or **hypertexts**.
- **Markup Language** means we can markup web pages like setting a text as bold, button as red color, etc. Markup is done through **HTML Tags**.
- It defines the content and structure of a website.

These are all the tags that are available in HTML:

<a>	<canvas>	<dt>	<iframe>	<meta>	<rp>	<sup>
<abbr>	<caption>			<meter>	<rt>	<svg>
<address>	<cite>	<embed>	<input>	<nav>	<ruby>	<table>
<area>	<code>	<fieldset>	<ins>	<noscript>	<s>	<tbody>
<article>	<col>	<figcaption>	<kbd>	<object>	<samp>	<td>
<aside>	<colgroup>	<figure>	<keygen>		<script>	<template>
<audio>	<data>	<footer>	<label>	<optgroup>	<section>	<textarea>
	<datalist>	<form>	<legend>	<option>	<select>	<tfoot>
<base>	<dd>	<head>		<output>	<small>	<th>
<bdi>		<header>	<link>	<p>	<source>	<thead>
<bdo>	<details>	<hgroup>	<main>	<param>		<title>
<blockquote>	<dfn>	<h1> to <h6>	<map>	<picture>		<tr>
<body>	<dialog>	<hr>	<mark>	<pre>	<style>	
 	<div>	<html>	<menu>	<progress>	<sub>	<var>
<button>	<dl>	<i>	<menuitem>	<q>	<summary>	<video>
						<wbr>

And these are the essential ones that we'll use most of the time:

<a>	<canvas>	<dt>	<iframe>	<meta>	<rp>	<sup>
<abbr>	<caption>			<meter>	<rt>	<svg>
<address>	<cite>	<embed>	<input>	<nav>	<ruby>	<table>
<area>	<code>	<fieldset>	<ins>	<noscript>	<s>	<tbody>
<article>	<col>	<figcaption>	<kbd>	<object>	<samp>	<td>
<aside>	<colgroup>	<figure>	<keygen>		<script>	<template>
<audio>	<data>	<footer>	<label>	<optgroup>	<section>	<textarea>
	<datalist>	<form>	<legend>	<option>	<select>	<tfoot>
<base>	<dd>	<head>		<output>	<small>	<th>
<bdi>		<header>	<link>	<p>	<source>	<thead>
<bdo>	<details>	<hgroup>	<main>	<param>		<title>
<blockquote>	<dfn>	<h1> to <h6>	<map>	<picture>		<tr>
<body>	<dialog>	<hr>	<mark>	<pre>	<style>	
 	<div>	<html>	<menu>	<progress>	<sub>	<var>
<button>	<dl>	<i>	<menuitem>	<q>	<summary>	<video>
						<wbr>

Tags vs. Elements: There are two common terms in HTML that are Tags and Elements. Tags are a piece of a HTML element inside angle brackets (<>). For example, <h1> is a tag. Element consists of tags and its content. For example, <h1>Hello World</h1> is an element.

Basic HTML Tags:

1. **Heading Tags (h1-h6):** These are used to display headings in a webpage. Heading Tags can be written as:

<h1>Hello World</h1>

In this, **<h1>** is an opening tag and **</h1>** is a closing tag. Note that the closing tag contains a forward slash i.e. "/". Between these two tags is the content of that element.

There are 6 levels of heading starting from h1 (highest level) to h6 (lowest level). Each of these have their own default size. Note that there can only be 1 single h1 element in the whole HTML code.

2. **Paragraph Tag (p):** This tag is used to add paragraph or text content. This tag can be represented as:

<p>Paragraph 1</p>

3. **Void Elements (hr and br):** Void elements are those where we don't specify any content between tags. Above <h1>-<h6> and <p> tags are non-void elements and are called **Container Tags**. **hr** and **br** are void elements and are called **Empty Tags or Self-closing tags**. **hr** stands for **horizontal rule** and **br** stands for **break**. These void elements are represented as:

<hr /> or

<hr/> is used to separate different contents by providing a horizontal rule or line between them.
 is used to break a line into separate lines.

4. **Bold, underline, italic tags (b, u, i):** These tags are used to format the text contents. tag is used to bold the text, <u></u> makes the text underline and <i></i> is used to make text italics.
5. **Comments (<!--Content-->):** Comments are used to add some notes to the document and are not rendered by the browser. This is represented as:

<!--This is a comment-->

Intermediate HTML Elements:

1. **List Elements:** We can create 2 types of lists: unordered and ordered lists. **Unordered list** can be created using tag. **Ordered list** can be created using tag. In these tags, we need to add another tag for making a list and that is **List Item** or tag. So, the full list can be created by:

	
Item 1	Item 1
Item 2	Item 2
Item 3	Item 3
	

Difference between these two types of lists is that we get bullet points in unordered lists as sequence is not important but in ordered lists, we get numbers as sequence is important.

We can nest multiple lists inside a list like this:

```
<ul>
  <li>Item 1</li>
  <li>Item 2
    <ul>
      <li>Item 2.1</li>
      <li>Item 2.2</li>
    </ul>
  </li>
</ul>
```

In this, to nest another list, we need to add that list **inside a list item** element.

2. **Anchor Tag (a)**: This tag is used to add a hyperlink to a webpage which can be used to navigate to different pages of the same website or to another website. This is represented as:

```
<a>This is a link to a website</a>
```

Now, this is not an active link which means this is just a text. To make this an active link and able to click on it, we need to add an **attribute**. **Attributes** are used to add additional information to HTML Tags. So in this, we add an attribute called **href** which stands for **hyperlink reference** which tells the user where this link should take the user. So, final link tag should look like this:

```
<a href="https://www.google.com">This is a link to a
website</a>
```

In this, **href** is the attribute, **https://www.google.com** is the destination and **This is a link to a website** is the text that is shown on the webpage. Now we can click on this link and after clicking, it takes us to google homepage.

3. **Image Tag ()**: This tag is used to add images to websites. It looks like this:

```
<img src='url' />
```

In this, **src** attribute is used and it tells us what is the source of the image and the **url** will be the location of the image. This tag is a **void or empty tag** and doesn't have any closing tag.

```

```

Another attribute that is commonly used in this tag is **alt (alternative text description)**. This is used by screen readers to help people with blindness and is also used to display a temporary text if the image doesn't load.

Other attributes that we can use with tag:

- **Width**: to set the width of an image
- **Height**: to set the height of an image

File Paths (Absolute and Relative):

File path is a unique location of a file or folder on our systems. Let's say we've a file structure as below:

Root

- > essay.docx
- > Project
 - > dog.png
 - > index.html
 - > Images
 - > cat.png

A file path can be like this:

Root/Project/index.html

Absolute file path is relative to the root of the computer. For example:

Root/Project/Images/cat.png

In this, we're navigating from the root of the computer and traversing first into the **Project** folder, then in **Images** folder and lastly to **cat.png** file.

Relative file path is not relative to the root of the computer else it is relative to another file that we're working with. For example, we're currently in **index.html** file and we want to access an image stored in **Images** folder so file path should be:

Images/cat.png

There are some special characters that make it easier for us to navigate the file structure.

1. **Two double dots (..)**: two double dots at the beginning of file path means **to go up a level**. For example, we're in **index.html** and we want to access **essay.docx**, then we can write:

../essay.docx

So from **index.html**, we go one level up and reach the **Root** directory and then we can go to **essay.docx** file.

2. **Single dot (.)**: single dot at the beginning of file path means **stay within the current directory** and look for other files or folders. For example, we're in **index.html** and we want to access **dog.png**, then we can write:

./dog.png

Multi-pages Websites:

To create a multi-page website, we need multiple web pages like **index.html**, **about.html**, **contact.html**, etc. When these webpages go

into the same project folder, it becomes a multi-page website. A multi-page website will look like this:

Project

- > index.html
- > about.html
- > contact.html
- > assets
 - > images
 - user.png

To navigate to different webpages, we can use hyperlinks or anchor tags like below inside **index.html**:

```
<a href="/about.html">About page</a>
<a href="/contact.html">Contact Page</a>
```

HTML Boilerplate:

Every HTML webpage has a structure which is called HTML Boilerplate or Skeleton. It is a standardised markup that needs to go in every single HTML page. It look like this:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>My Website</title>
  </head>
  <body>
    <h1>Hello World!</h1>
  </body>
</html>
```

- **<!DOCTYPE html>:** This is a document type declaration and tells us what type of file and what version of the file was written in.

- <html></html> element: This is the root element of an HTML document and all other elements must go inside this tag. In this, `lang="en"` means the language of the text content.
- <head></head> element: This element contains information about the website and is not displayed to the user.
- <meta charset="UTF-8" /> element: This tells us that text content is encoded using the UTF-8 encoding system. This is a standard encoding as it contains all international symbols and we don't have to worry about proper rendering of text content.
- <title></title> element: This is used to define the title of the website and is shown in the browser's title bar.
- <body></body> element: This is where all our HTML code lies which is shown to the user. All the content like heading, paragraphs, images, links, etc. goes inside this tag.