

Introduction to HTML

What is HTML?

HTML stands for HyperText Markup Language. It is the standard language used to create and structure the content of a webpage seen in web browsers.

Unlike programming language (which compute logic) HTML marks up text and other content to tell the browser how to display it.

- "Markup" means: adding annotations to content that define structure
- Think of HTML as the skeleton of a web page - it gives structure and meaning to content.

The Browser & HTML

When your browser loads a .html file:

- 1.) It reads the HTML tags
- 2.) It interprets them to render structure and element (headings, paragraphs, links, images, lists, etc).
- 3.) It renders visually based on that structure.

You don't need to run HTML like a script - the browser just reads and displays.

Phantom of an HTML Document

Every HTML document you create should follow this pattern:

```
<!DOCTYPE html>
```

```
<html> (root, makes it a document, starts HTML)
```

```
  <head> (inside of body, contains metadata, like title)
```

```
    <title> Page Title </title> (in browser tab)
```

```
  <head> (inside of body, contains metadata, like title)
```

```
  <body> (inside of body, contains visible content)
```

```
    </> - Visible page content goes here →
```

```
  </body>
```

```
</html> (inside of body, ends document)
```

Breakdown:

- `<!DOCTYPE html>` → tells the browser this is an HTML page
- `<html>` - `</html>` → root element containing everything
- `<head>` - `</head>` → contains metadata e.g.: page title, linked files
- `<title>` - `</title>` → the title shown in browser tab
- `<body>` - `</body>` → everything inside here visibly appears on the webpage

HTML tags

Tags are the building blocks of HTML.
Think of them as labels you add your content

Typical Structure:

- Opening tag: <tagname>
- Content goes here
- closing tag: </tagname>

for eg: <p> *text* </p> *closed*

• <p>: paragraph tag (not block)

- </p>: end of paragraph

Common HTML Tags & what they Mean

Text Structure

- <h1> - <h6>: Heading A *h1* is biggest/most important; <h6> is smallest/least important *h6*
- Use these to create visual hierarchy in your document *h1, h2, h3, h4, h5, h6*

Ex:-

<h1> Main Heading </h1>

<h2> SubHeading </h2>

<p> Some text under a heading </p>

Paragraphs

- <p> → denotes a "block" of text *<1>*
HTML ignores plain spaces and line breaks - only
<p> defines a separate paragraph on screen

Links (Anchor Tags)

Used to link between pages or to other websites

` Visit Example `

- href means "hypertext reference" - the URL to go to when clicked
- The linked text appears clickable in the browser

Images

~ Insert pictures with the img tag:

``

- src: file path or web URL of the image
- alt: alternative text shown if img files fail to load

- Important for accessibility (Screen readers) and SEO.

Lists

HTML provides two common list types:

Unordered list: a collection of items

~ ~ ~

``

` Item1 ` with a bullet

` Item2 ` with a bullet

`` to close the list

- Shows with bullets

Ordered list

```
<ol>
  <li> first </li>
  <li> Second </li>
```

```
</ol>
```

- Shows with numbers

lists are great for menus or any structured grouping

Line Breaks and Horizontal Rules

• `
` → line break (like pressing 'Enter' in text)

• `<hr>` → horizontal rule (a dividing line)

These don't need closing tags and help organize content visually

Semantic HTML

Semantic elements clearly describe their meaning:

Tag

`<header>`

Meaning

Header Section

`<nav>`

Navigation Links

`<main>`

Main Page Content

`<footer>`

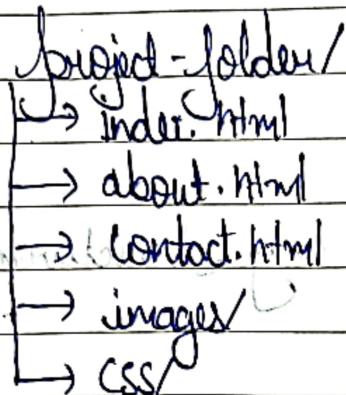
Footer Area

Using semantic tags makes your page:

- Easier to read
- Better for accessibility
- Better for search engines

HTML Project Structure

Once you start building bigger websites
you organize files like this:



`<head> </head>`

`<body> </body>`

`</>`

minimum after word.

no where safe from user click.

origins

0 0

what information has stored will

This keeps things tidy and lets you reuse photos
and styles across pages

Count, children of your information → ``

Practice makes perfect `new fresh test`

Doing mini challenges is

how you absorb HTML

- build a personal homepage
- add headings and paragraphs
- Create navigation between pages
- insert images with meaningful alt text

private

Top Tips from the Course

- Always use semantic tags when possible - Search engines and accessibility tools love them
- Keep your folder structure organized - think ahead as project grows
- Don't just copy code - type it yourself to solidify muscle memory

utilizing new methods

designs, tools, etc. available