



Table of Contents

Introduction	1
Purpose	1
Scope	1
Procedure	1
Problem Statement	1
How the Web Works	2
1. What is the Web?	2
2. What is Frontend Development? (The "Face")	3
The Golden Trio of Frontend	3
3. What is Backend Development? (The "Brain")	4
Key Components of Backend	4
The Core Structure (html, head, body)	5
1. The Declaration: <!DOCTYPE html>	5
2. The Root: <html>	5
Example	6
3. The Brain: <head>	7
1. The <meta> Tag: Name & Content	7
2. Styling the Page: <link> & <style>	8
3. The Favicon: Adding an Icon	9
4. The <script> Tag: JavaScript	9
4. The Visible Content: <body>	10
Text & Structure	11
1. Headings (<h1> to <h6>)	11
2. Paragraphs (<p>)	12
3. Preformatted Text (<pre>)	12
Text Formatting & Special Characters	13
1. Physical Formatting Tags	13
2. HTML Entities (Special Symbols)	15
Links & Navigation (<a>)	16
1. Linking to External/Internal Pages	16
2. Linking to a Specific Section (Bookmark)	17
3. Linking to Email	17
Lists	18
1. Unordered List ()	18

2. Ordered List ()	19
3. The Definition List (<dl>)	20
Layout & Grouping	21
1. The <div> (Block Container)	21
2. The (Inline Container)	22
Media Tags (img, audio, video, iframe)	23
1. Images ()	23
2. Audio (<audio>)	25
3. Video (<video>)	27
4. The Inline Frame (<iframe>)	29
5. Note on File Paths (The most common error!)	30
Tables (<table>)	31
1. The Container & Semantics	31
2. The Rows & Cells	31
3. Merging & Sizing (The Magic Attributes)	31
Data Collection (Forms)	33
1. The Container (<form>)	33
2. The Connection (<label>)	34
3. The Info (<input>)	35
Input Type Values	36
4. Grouping & Structure (<fieldset> & <legend>)	37
5. Miscellaneous	38
The <button> Tag	38
6. Code summing it all	39
Advanced Input Controls	41
1. The Dropdown List (<select>)	41
2. The Data List (<datalist>)	42
3. The Text Area (<textarea>)	43
HTML Semantics	44
1. The Header (<header>)	45
2. The Navigation (<nav>)	46
3. The Main Content (<main>)	47
4. The Footer (<footer>)	48
5. The Article (<article>)	49
6. The Section (<section>)	50

7. The Sidebar (<aside>)	50
8. The Interactive Toggle (<details> & <summary>)	51
9. The Image Container (<figure> & <figcaption>)	52
10. The Highlighter (<mark>)	53
11. The Time (<time>)	53
12. The Ultimate Semantic Layout Code	54
Conclusion	55
Feedback & Contribution	55
Copyright & Usage	55
Acknowledgments	56

Introduction

Purpose

The purpose of this project is to design and develop a structured, web-based digital document using HyperText Markup Language (HTML). The goal is to create a dynamic and accessible platform that organizes complex study materials into a navigable, user-friendly interface. By leveraging web technologies, this document aims to transform static information into an interactive format that enhances readability and accessibility across different devices and browsers.

Scope

The scope of this document encompasses the complete structural development of the web page. This includes the implementation of semantic HTML5 elements to define the layout (headers, sections, and footers), the organization of content hierarchies, and the integration of multimedia resources such as images and code blocks. The project focuses on creating a clean, valid codebase that serves as the backbone for presenting technical data effectively on the web.

Procedure

The development process followed a standard web engineering workflow. It began with **requirement analysis**, identifying the necessary sections and navigation flow. This was followed by **structural coding**, where the core HTML skeleton was built to house the content. The procedure involved using specific tags for text formatting, list creation, and hyperlinking resources. Finally, the document underwent **browser testing** to ensure consistent rendering and structural integrity across various viewing environments.

Problem Statement

Managing large volumes of educational content in traditional formats (like plain text files or handwritten notes) often leads to fragmentation and difficulty in retrieval. Users struggle to quickly locate specific topics or visualize the relationships between different modules. This HTML document addresses that problem by providing a centralized, hyperlinked architecture. It solves the issue of navigability and content isolation, allowing for seamless transition between topics and a more organized user experience.

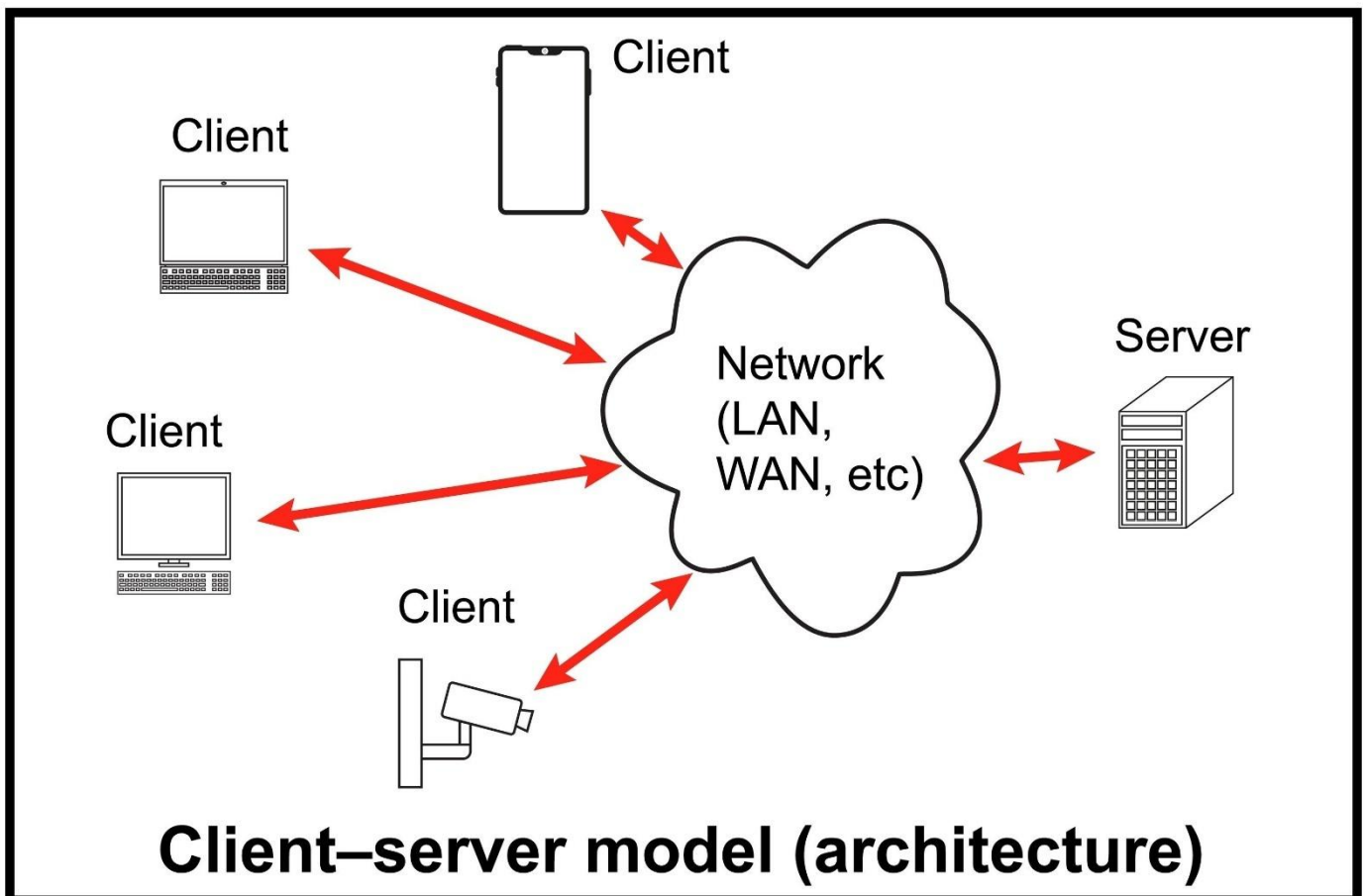
How the Web Works

1. What is the Web?

At its core, the World Wide Web isn't magic—it is simply a massive network of computers talking to each other. It operates on a model called the **Client-Server Model**.

To understand this, let's use the **Restaurant Analogy** 🍽️:

- **The Client (You/Your Browser)** 🖥️: You are the customer sitting at a table. You look at the menu (the website) and decide what you want. You don't cook the food; you just ask for it.
- **The Request** 🗣️: This is you telling the waiter, "I want the steak." In technical terms, your browser sends an **HTTP Request** to a server asking for a specific page (like google.com).
- **The Server (The Kitchen)** 🏢: This is a powerful computer located somewhere else (maybe in a data center in Virginia or Frankfurt). It receives your order, finds the files you need (images, text, code), and prepares them.
- **The Response** 📦: The waiter brings your food to the table. The server sends the files back to your computer, and your browser displays them as a website.



2. What is Frontend Development? (The "Face")

Frontend (also called Client-Side) is everything the user **sees, touches, and interacts with**. It is the "User Interface" (UI).

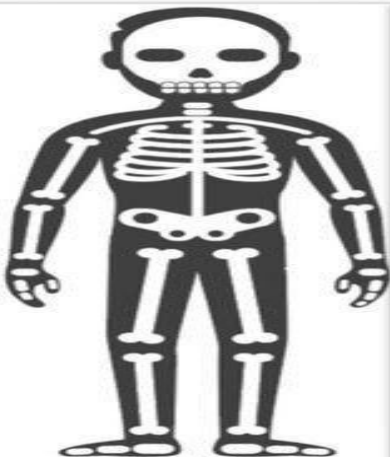
Going back to our **Restaurant Analogy**: The Frontend is the **dining area**. It includes the layout of the tables, the color of the paint on the walls, the design of the menu, and how polite the waiter is. It is all about the **Experience**.

The Golden Trio of Frontend

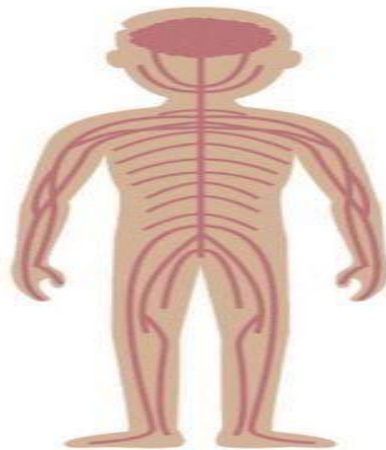
To build a frontend, you need three main technologies working together:

1. **HTML (HyperText Markup Language)** 🦴 :
 - **Role:** Structure.
 - **Analogy:** The skeleton or the bricks of a house. It defines *what* is on the page (e.g., "This is a heading," "This is a button," "This is an image").
2. **CSS (Cascading Style Sheets)** 🎨 :
 - **Role:** Presentation.
 - **Analogy:** The paint, curtains, and decoration. It defines *how* the page looks (colors, fonts, spacing). Without CSS, the web would look like a boring Word document.
3. **JavaScript (JS)** ⚡ :
 - **Role:** Behavior/Interactivity.
 - **Analogy:** The electricity or the plumbing. It makes things *do* stuff (e.g., when you click "Like," the heart turns red; or a popup window appears).

HTML



JS



CSS



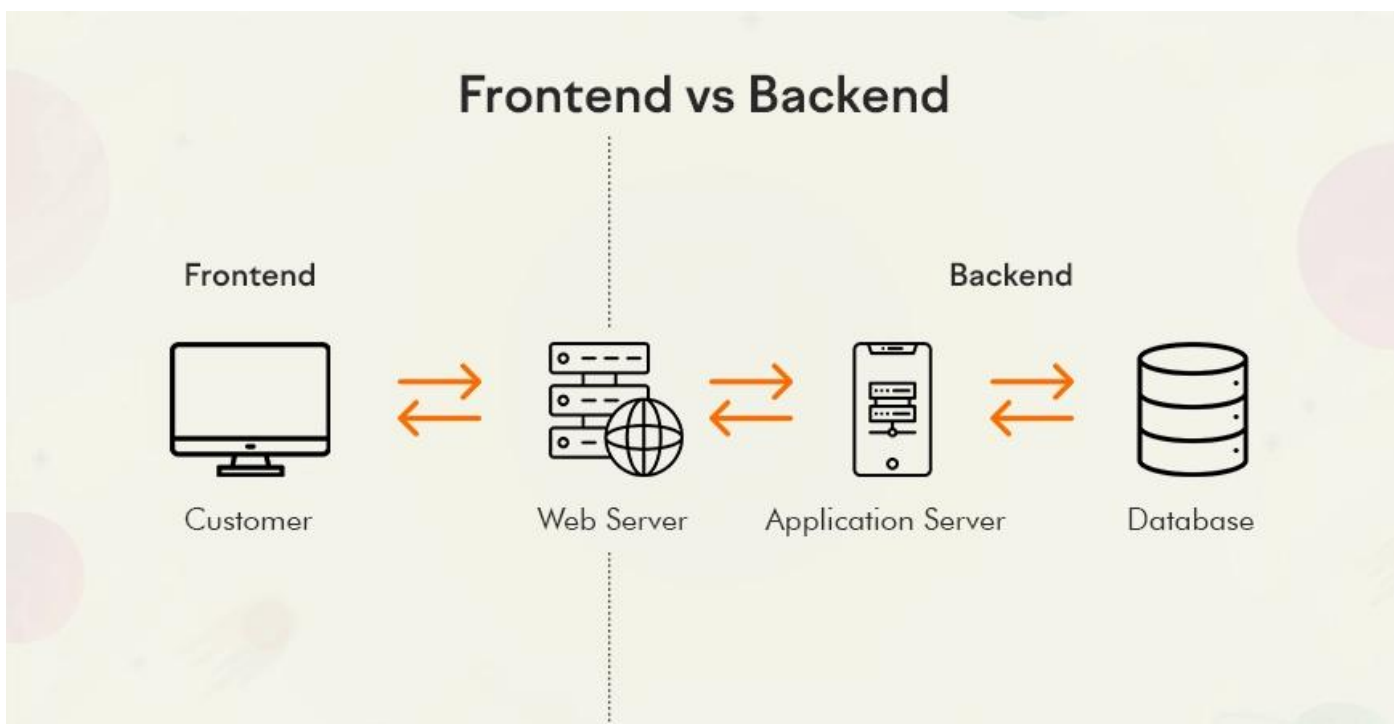
3. What is Backend Development? (The "Brain")

Backend (also called Server-Side) is the part of the application that the user **never sees**. It handles the logic, database interactions, and user authentication.

Going back to our **Restaurant Analogy**: The Backend is the **Kitchen**. You (the customer) aren't allowed in the kitchen. You don't see the chef chopping onions or the messy pots and pans. You only care that the food (data) comes out correctly.

Key Components of Backend ⚙️

1. **The Server:** The computer listening for requests.
2. **The Logic:** The code that decides what to do. (e.g., "If the user types the wrong password, send an error message.")
3. **The Database** 📁: This is the filing cabinet where data is stored (Usernames, Passwords, Product details). Common databases include MySQL, PostgreSQL, or MongoDB.



The Core Structure (*html, head, body*)

Think of an HTML file like a **human body**.

1. **<html>**: The entire person.
 2. **<head>**: The brain (thoughts, memories, instructions that others don't see directly).
 3. **<body>**: The physical body (face, arms, clothes) that everyone sees and interacts with.
-

1. The Declaration: *<!DOCTYPE html>*

Before we even open the html tag, we must declare the document type.

- **Purpose:** Tells the browser "Hey! This is a modern **HTML5** document, not an old version from 1999."
 - **Attributes:** None. It is a statement, not a tag.
 - **Placement:** Must be the very first line of code.
-

2. The Root: *<html>*

This is the container for **everything**. All other elements must be inside this tag.

- **Purpose:** Defines the root of the HTML document.
- **Key Attributes:**
 - **lang:** Specifies the language of the page (Crucial for Search Engines & Screen Readers).
 - "en": English US
 - "ar": Arabic EG
 - "fr": French FR
 - "es": Spanish ES
 - **dir:** Specifies the text direction.
 - "ltr": Left-to-Right (Default for English).
 - "rtl": Right-to-Left (For Arabic, Hebrew).

Example

```
<html lang="en" dir="ltr">  
  <!-- Everything else goes here -->  
</html>
```

3. The Brain: <head>

While the <body> handles what the user *sees*, the <head> handles what the browser and search engines *know*. Let's break down the advanced tags found here.

1. The <meta> Tag: Name & Content

The meta tag is a self-closing tag used to communicate with the browser or search engines (like Google). It often uses two key attributes: name (what type of info) and content (the actual info).

- **Tag:** <meta>
 - **Attributes:**
 - **name:** Specifies the type of metadata (e.g., description, keywords, author, viewport).
 - **content:** The value associated with the name.
-

A. Description (SEO Gold): This is the text that appears under your website link in Google search results.

```
<meta name="description" content="A comprehensive documentation of my AI & ML Diploma journey at Route.">
```

B. Keywords (Old School SEO): Words that help define what your page is about.

```
<meta name="keywords" content="AI, Machine Learning, Python, Route, Data Science, HTML">
```

C. Author: Gives credit to the creator.

```
<meta name="author" content="[Your Name]">
```

D. Viewport (Crucial for Mobile): Without this, phones will shrink your website to look like a desktop site (tiny text).

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

2. Styling the Page: <link> & <style>

There are two ways to add CSS (styles) inside the head.

A. The <link> Tag (External CSS)

This is the professional way. You link to a separate .css file.

- **Tag:** <link> (Self-closing).
- **Attributes:**
 - **rel** (Relationship): Must be set to "stylesheet".
 - **href** (Hypertext Reference): The path to your CSS file.

```
<link rel="stylesheet" href="style.css">
```

B. The <style> Tag (Internal CSS)

You write CSS code directly inside the HTML file. Good for small projects.

- **Tag:** <style>

```
<style>
  body {
    background-color: #f0f0f0;
    font-family: Arial, sans-serif;
  }
</style>
```

3. The Favicon: Adding an Icon

That little logo you see next to the page title in the browser tab is called a **Favicon** (Favorite Icon).

- **Tag:** `<link>`
- **Attributes:**
 - **rel:** Set to "icon" (or "shortcut icon").
 - **href:** The path to your image (usually a .png or .ico file).
 - **type** (Optional): Helps the browser know the format (e.g., image/x-icon or image/png).

```
<link rel="icon" href="images/my-logo.png" type="image/png">
```

4. The `<script>` Tag: JavaScript

This is how you add logic (interactivity) to the page.

A. External Script (Best Practice)

Links to a separate .js file.

- **Tag:** `<script>` (Not self-closing! You need the closing tag).
- **Attributes:**
 - **src:** Path to the JS file.

```
<script src="main.js"></script>
```

B. Internal Script

Writing JS directly in the HTML.

```
<script>  
  console.log("Welcome to my AI Journey website!");  
</script>
```

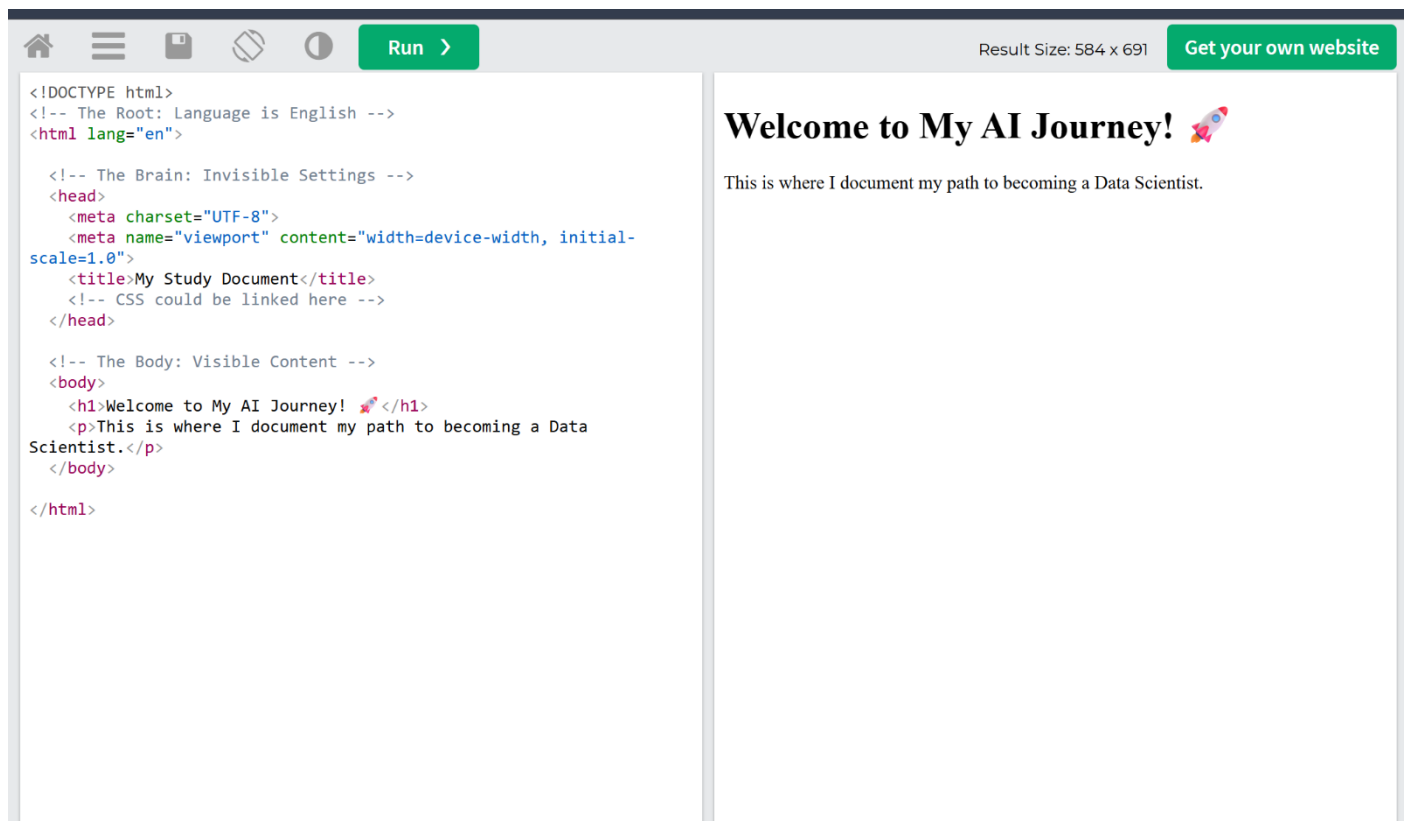
4. The Visible Content: <body>

This is the viewport. If you want the user to see it, it **must** go here.

- **Purpose:** Contains text, images, videos, games, buttons, etc.
- **Key Attributes (Global):**
 - **style:** Used for inline CSS (e.g., style="background-color: blue;").
 - **id:** A unique nickname for an element (e.g., id="main-header").
 - **class:** A group name to style multiple things at once (e.g., class="button").
 - **link:** color of the link before clicking on it
 - **alink:** color of the link after clicking on it
 - **vlink:** color of the link after visiting it

The Full Code Structure

Here is how they fit together.



```
<!DOCTYPE html>
<!-- The Root: Language is English -->
<html lang="en">

  <!-- The Brain: Invisible Settings -->
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>My Study Document</title>
    <!-- CSS could be linked here -->
  </head>

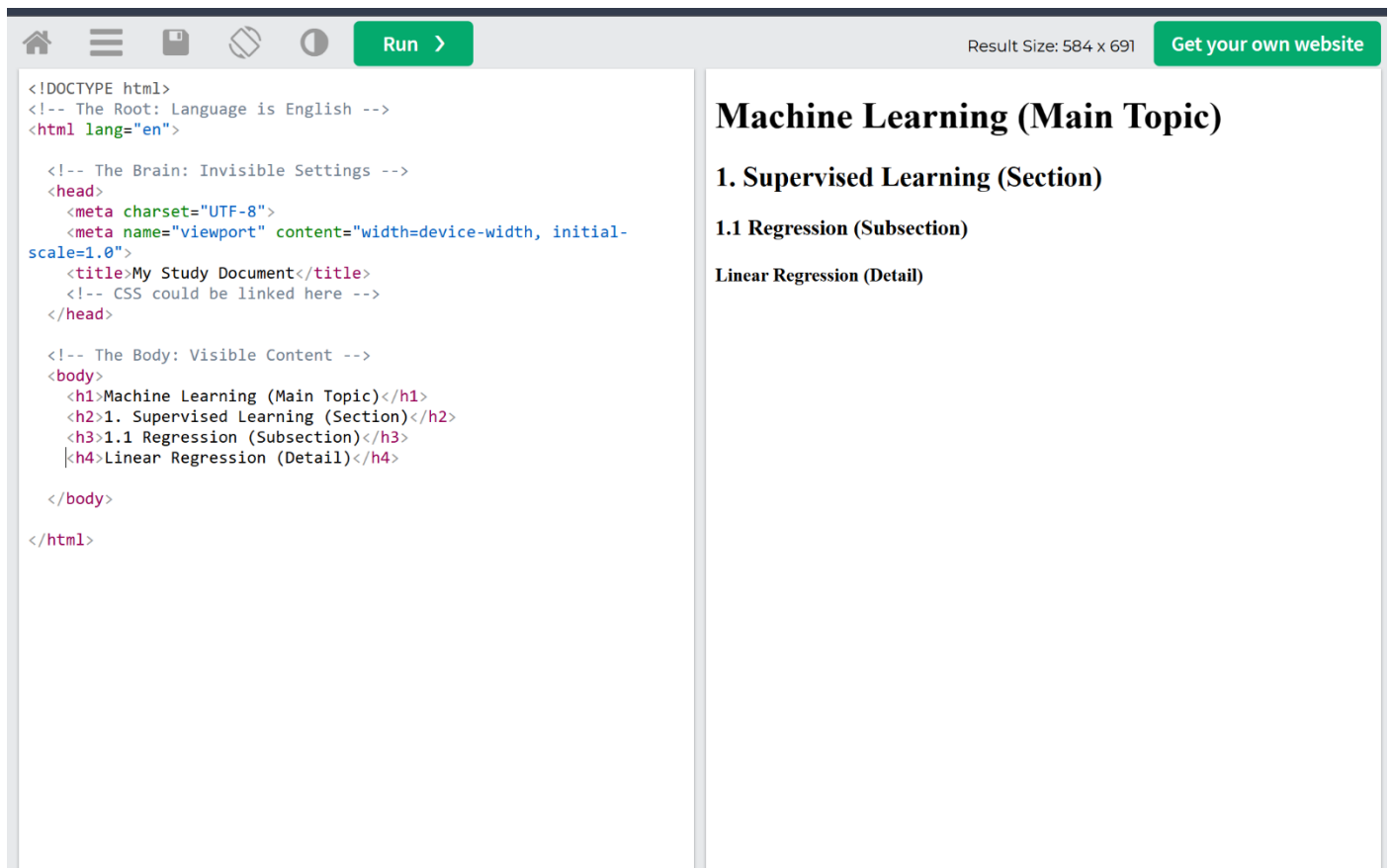
  <!-- The Body: Visible Content -->
  <body>
    <h1>Welcome to My AI Journey! 🚀</h1>
    <p>This is where I document my path to becoming a Data
Scientist.</p>
  </body>
</html>
```

Text & Structure

1. Headings (<h1> to <h6>)

Headings give your page a hierarchy. Search engines use them to understand the outline of your content.

- **Tags:** <h1>, <h2>, <h3>, <h4>, <h5>, <h6>
- **Rule:** Use <h1> **only once** per page (usually the main title). Use <h2> for main sections, <h3> for subsections, etc.



The screenshot shows a web browser interface. On the left, there is a code editor displaying HTML code. On the right, the rendered page is shown. The code editor contains the following HTML code:

```
<!DOCTYPE html>
<!-- The Root: Language is English -->
<html lang="en">

  <!-- The Brain: Invisible Settings -->
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>My Study Document</title>
    <!-- CSS could be linked here -->
  </head>

  <!-- The Body: Visible Content -->
  <body>
    <h1>Machine Learning (Main Topic)</h1>
    <h2>1. Supervised Learning (Section)</h2>
    <h3>1.1 Regression (Subsection)</h3>
    <h4>Linear Regression (Detail)</h4>
  </body>
</html>
```

The rendered page on the right shows the following structure:

- Machine Learning (Main Topic)** (h1)
- 1. Supervised Learning (Section)** (h2)
- 1.1 Regression (Subsection)** (h3)
- Linear Regression (Detail)** (h4)

2. Paragraphs (<p>)

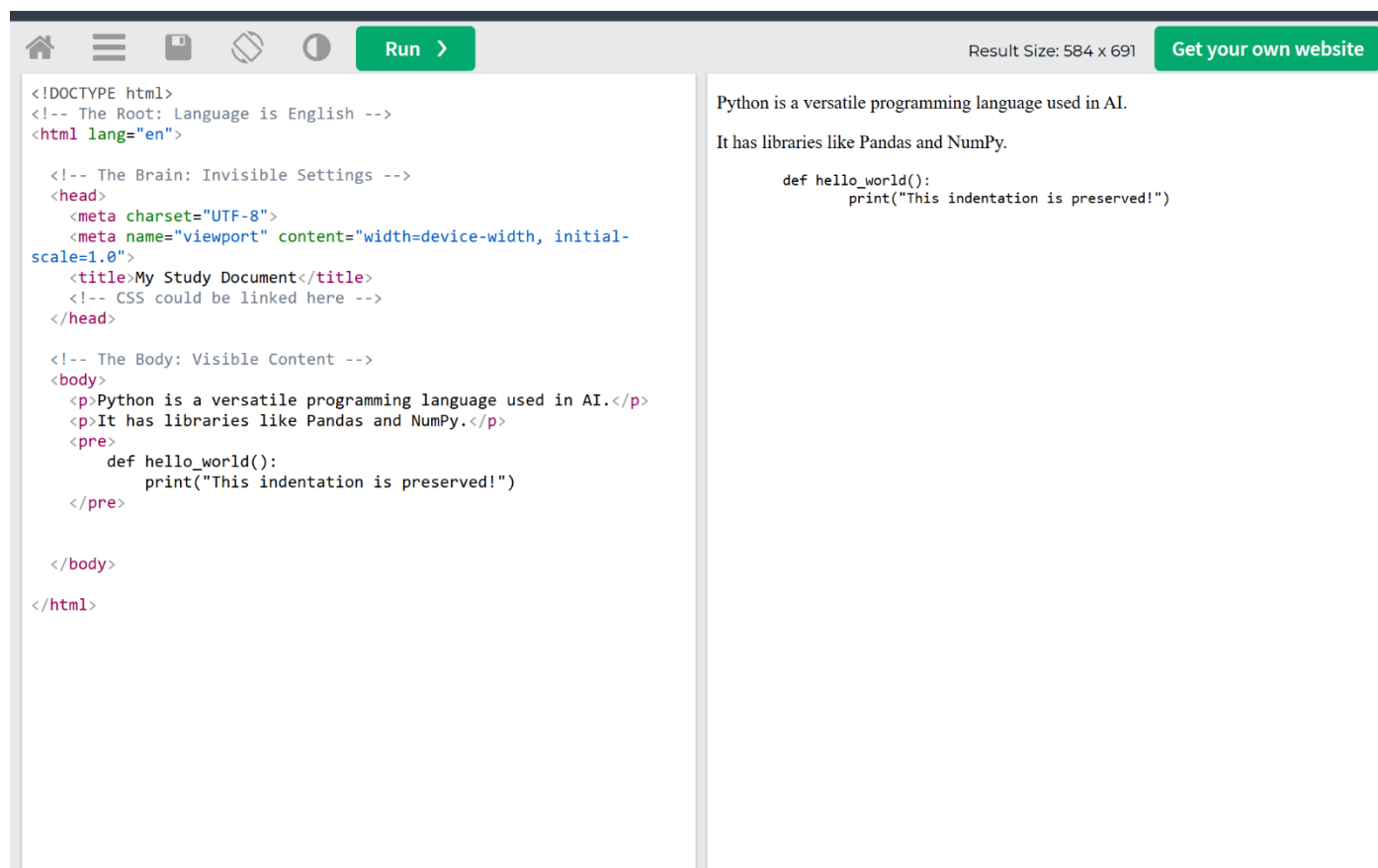
The standard tag for blocks of text.

- **Tag:** <p>
- **Behavior:** Browsers automatically add a margin (white space) before and after a paragraph.
- **Note:** You cannot put a paragraph inside another paragraph.

3. Preformatted Text (<pre>)

By default, HTML ignores extra spaces and line breaks. The <pre> tag **preserves** them. This is perfect for displaying **Code Snippets** or poetry.

- **Tag:** <pre>
- **Font:** Usually displays in a fixed-width font (like Courier).



```
<!DOCTYPE html>
<!-- The Root: Language is English -->
<html lang="en">

  <!-- The Brain: Invisible Settings -->
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>My Study Document</title>
    <!-- CSS could be linked here -->
  </head>

  <!-- The Body: Visible Content -->
  <body>
    <p>Python is a versatile programming language used in AI.</p>
    <p>It has libraries like Pandas and NumPy.</p>
    <pre>
      def hello_world():
        print("This indentation is preserved!")
    </pre>

  </body>
</html>
```

Python is a versatile programming language used in AI.
It has libraries like Pandas and NumPy.

```
def hello_world():
    print("This indentation is preserved!")
```

Result Size: 584 x 691 [Get your own website](#)

Text Formatting & Special Characters

1. Physical Formatting Tags

These change the visual appearance of the text.

Tag	Name	Visual Effect	Analogy
	Bold	Darker, heavier text	Emphasis without importance
	Strong Importance	Bold + semantic meaning	Saying it louder AND it matters
<i>	Italic	Slanted text	Thoughts, foreign words
	Emphasized	Italic + semantic meaning	Stressing a word when speaking
<u>	Underline	Underlined text	Highlighted term / spelling
<s>	Strikethrough	Crossed-out text	Saying "ignore this"
	Deleted Text	Strikethrough + semantic	Removed info in document history
<ins>	Inserted Text	Underlined + semantic	New addition in document
<mark>	Highlight	Yellow background	Marker pen
<small>	Small Text	Smaller font size	Footnotes or disclaimers
<big> (obsolete)	Big Text	Larger font	Shouting (deprecated)
<sup>	Superscript	Text above baseline	Exponents (x ²)
<sub>	Subscript	Text below baseline	Chemical formulas (H ₂ O)
<abbr>	Abbreviation	Dashed underline on hover	Hover to reveal meaning
<cite>	Citation	Usually italic	Book/movie titles
<q>	Inline Quote	Adds quotation marks	Short quoted text
<code>	Inline Code	Monospaced font	Programmer text
<kbd>	Keyboard Input	Monospace + borders sometimes	"Press Ctrl+C"
<samp>	Sample Output	Monospaced	Program output

Run >
Result Size: 584 x 691
Get your own website

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>HTML Text Formatting Demo</title>
</head>
<body>

<h1>HTML Text Formatting Tags Demo</h1>

<h2>Bold and Strong</h2>
<p>This is <b>bold</b> text — emphasis without importance.</p>
<p>This is <strong>strong</strong> text — saying it louder AND it matters.</p>

<h2>Italic and Emphasized</h2>
<p>This is <i>italic</i> text — slanted for thoughts or foreign words.</p>
<p>This is <em>emphasized</em> text — stressing a word when speaking.</p>

<h2>Underline and Strikethrough</h2>
<p>This is <u>underlined</u> text — highlighted term.</p>
<p>This is <s>strikethrough</s> — saying "ignore this".</p>
<p>This is <del>deleted</del> text — removed info in document history.</p>
<p>This is <ins>inserted</ins> text — new addition in document.</p>

<h2>Highlight, Small, Big</h2>
<p>This is <mark>highlighted</mark> text — marker pen style.</p>
<p>This is <small>small</small> text — footnotes or disclaimers.</p>
<p>This is <big>big</big> text — shouting (deprecated).</p>

<h2>Subscript and Superscript</h2>
<p>Chemical formula: H<sub>2</sub></p>
<p>Math exponent: x<sup>2</sup> + y<sup>2</sup> = z<sup>2</sup></p>

<h2>Abbreviation and Citation</h2>

```

HTML Text Formatting Tags Demo

Bold and Strong

This is **bold** text — emphasis without importance.

This is **strong** text — saying it louder AND it matters.

Italic and Emphasized

This is *italic* text — slanted for thoughts or foreign words.

This is *emphasized* text — stressing a word when speaking.

Underline and Strikethrough

This is underlined text — highlighted term.

This is ~~strikethrough~~ — saying "ignore this".

This is ~~deleted~~ text — removed info in document history.

This is inserted text — new addition in document.

Highlight, Small, Big

This is **highlighted** text — marker pen style.

This is small text — footnotes or disclaimers.

This is big text — shouting (deprecated).

Subscript and Superscript

Run >
Result Size: 584 x 691
Get your own website

```

<p>This is <i>italic</i> text — slanted for thoughts or foreign words.</p>
<p>This is <em>emphasized</em> text — stressing a word when speaking.</p>

<h2>Underline and Strikethrough</h2>
<p>This is <u>underlined</u> text — highlighted term.</p>
<p>This is <s>strikethrough</s> — saying "ignore this".</p>
<p>This is <del>deleted</del> text — removed info in document history.</p>
<p>This is <ins>inserted</ins> text — new addition in document.</p>

<h2>Highlight, Small, Big</h2>
<p>This is <mark>highlighted</mark> text — marker pen style.</p>
<p>This is <small>small</small> text — footnotes or disclaimers.</p>
<p>This is <big>big</big> text — shouting (deprecated).</p>

<h2>Subscript and Superscript</h2>
<p>Chemical formula: H<sub>2</sub></p>
<p>Math exponent: x<sup>2</sup> + y<sup>2</sup> = z<sup>2</sup></p>

<h2>Abbreviation and Citation</h2>
<p>HTML stands for <abbr title="HyperText Markup Language">HTML</abbr>.</p>
<p>Book reference: <cite>The Great Gatsby</cite></p>

<h2>Inline Quote</h2>
<p>He said, <q>Hello, world!</q></p>

<h2>Code and Keyboard Input</h2>
<p>Inline code: <code>console.log("Hello")</code></p>
<p>Press <kbd>Ctrl</kbd> + <kbd>C</kbd> to copy.</p>
<p>Sample program output: <samp>Hello, World!</samp></p>
</body>
</html>

```

This is inserted text — new addition in document.

Highlight, Small, Big

This is **highlighted** text — marker pen style.

This is small text — footnotes or disclaimers.

This is big text — shouting (deprecated).

Subscript and Superscript

Chemical formula: H₂O

Math exponent: x² + y² = z²

Abbreviation and Citation

HTML stands for HTML.

Book reference: *The Great Gatsby*

Inline Quote

He said, "Hello, world!"

Code and Keyboard Input

Inline code: console.log("Hello")

Press Ctrl + C to copy.

Sample program output: Hello, World!

2. HTML Entities (Special Symbols)

HTML uses symbols like < and > for code. If you want to actually *display* those symbols on the screen, you must use "Entities."

Symbol	Entity Code	Description
Space	 	Non-Breaking Space. Forces a space where HTML usually collapses it.
<	<	Less Than. Used to show tags like on screen.
>	>	Greater Than.
&	&	Ampersand.

The screenshot shows a web development tool interface. On the left, there is a code editor with the following HTML code:

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<p>I learned that 6 &amp; 10 are even.</p>
<p>The tag for bold is &lt;b&gt;.</p>

</body>
</html>
```

On the right, the rendered output is displayed:

I learned that 6 & 10 are even.
The tag for bold is .

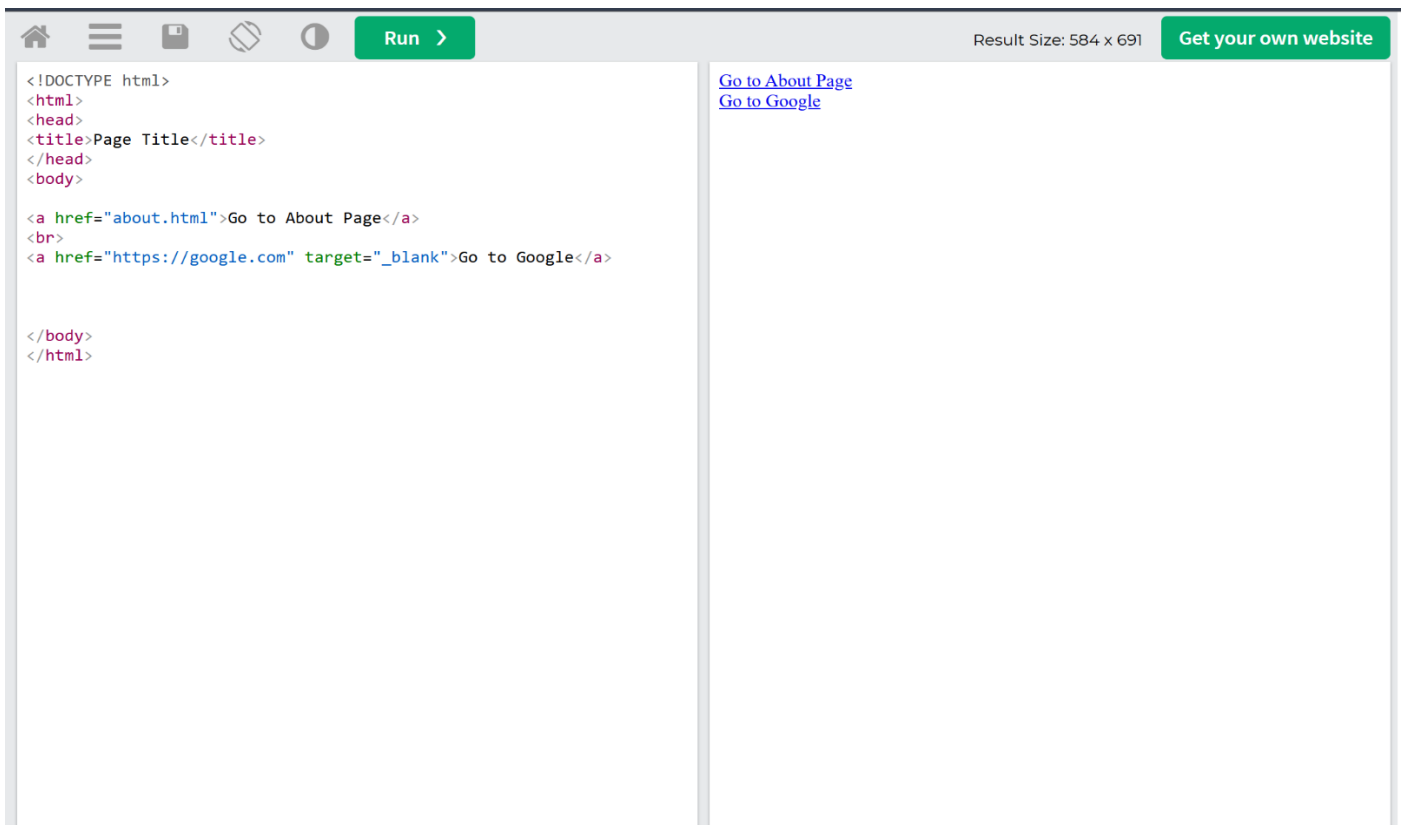
The interface includes a top navigation bar with icons for home, menu, save, undo, and refresh, a "Run" button, and a "Get your own website" button. The result size is indicated as 584 x 691.

Links & Navigation (<a>)

The **Anchor** tag is what makes the "Web" a web. It connects pages together.

- **Tag:** `<a>`
- **Key Attribute:** `href` (Hypertext Reference) - The destination.
- **Attribute:** `target` - Where to open the link.

1. Linking to External/Internal Pages



The screenshot shows a web browser interface. On the left, the HTML source code is displayed, and on the right, the rendered page is shown. The code includes a title 'Page Title' and two links: one to 'about.html' and one to 'https://google.com' with a target attribute of '_blank'. The rendered page shows two blue links: 'Go to About Page' and 'Go to Google'.

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<a href="about.html">Go to About Page</a>
<br>
<a href="https://google.com" target="_blank">Go to Google</a>

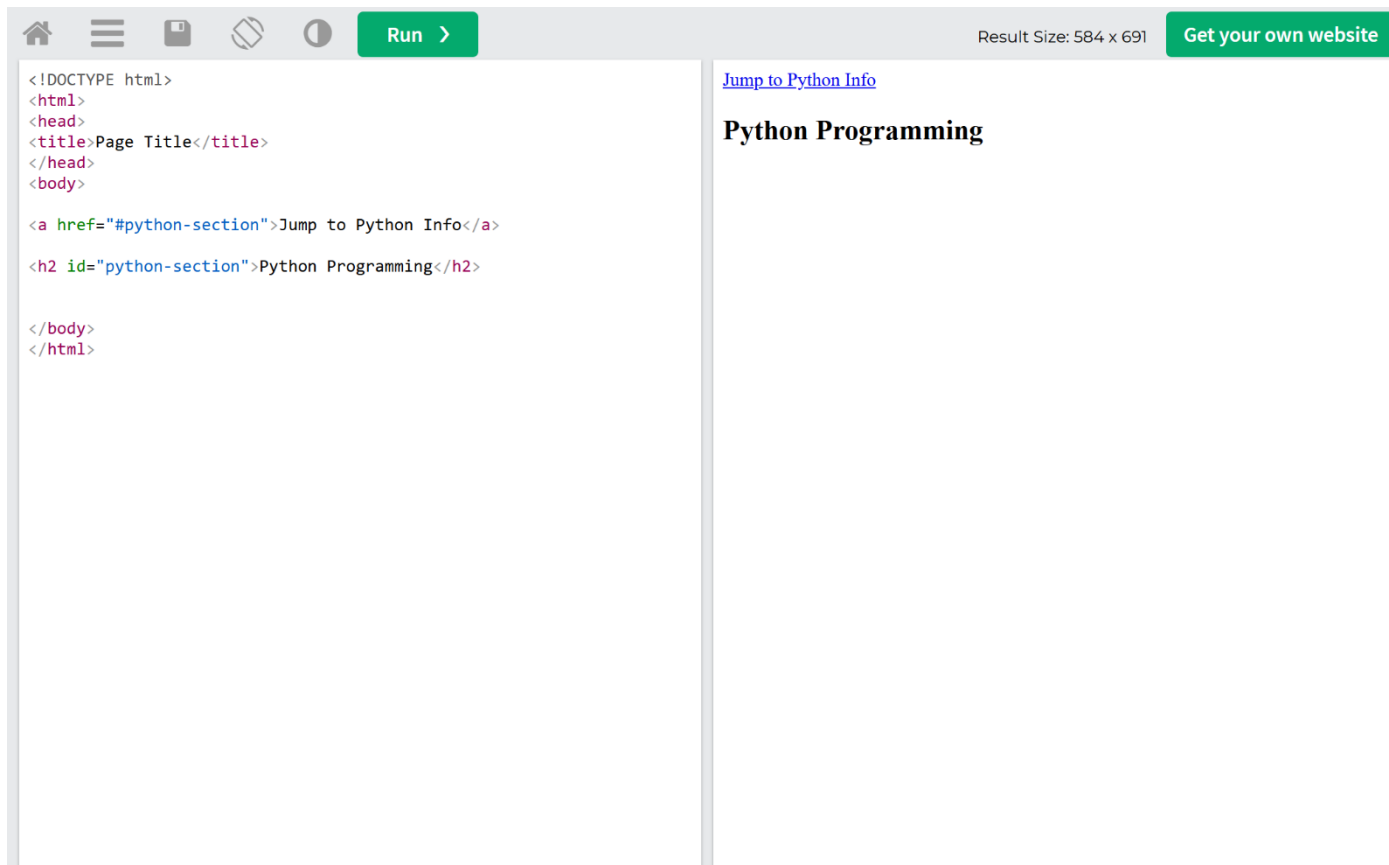
</body>
</html>
```

Go to About Page
Go to Google

2. Linking to a Specific Section (Bookmark)

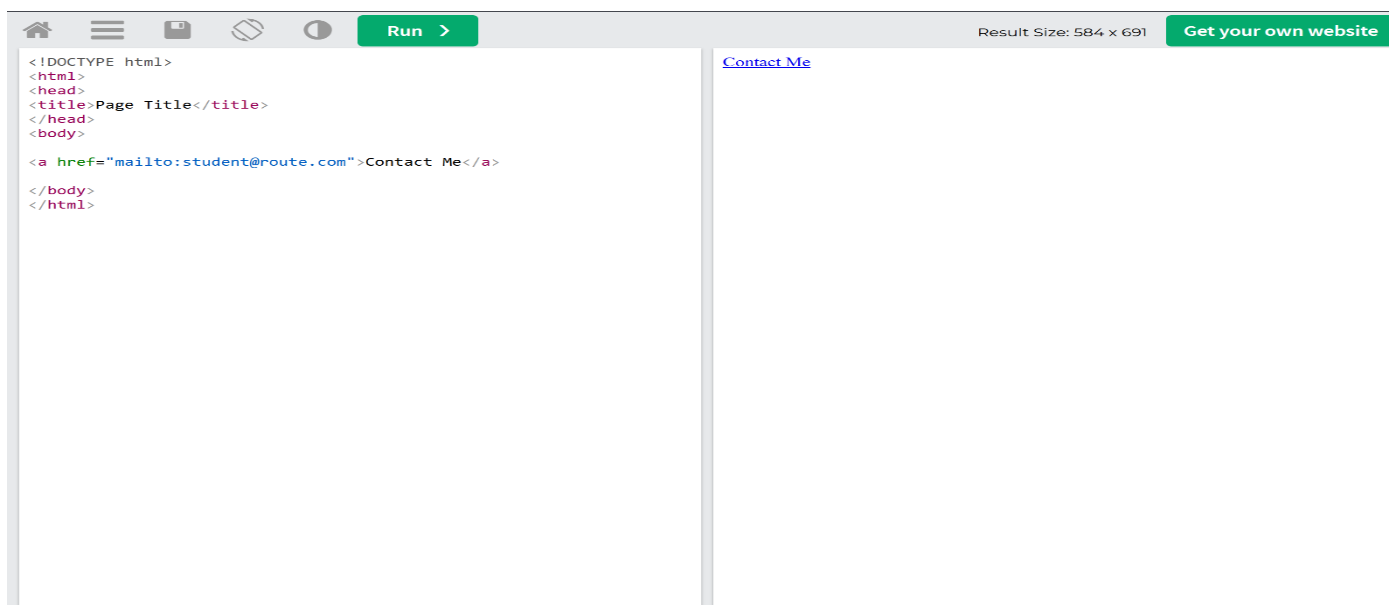
This is how "Table of Contents" work.

1. Give the target element an **id**.
2. Link to it using **#idname**.



3. Linking to Email

Opens the user's default mail app (like Outlook or Gmail).

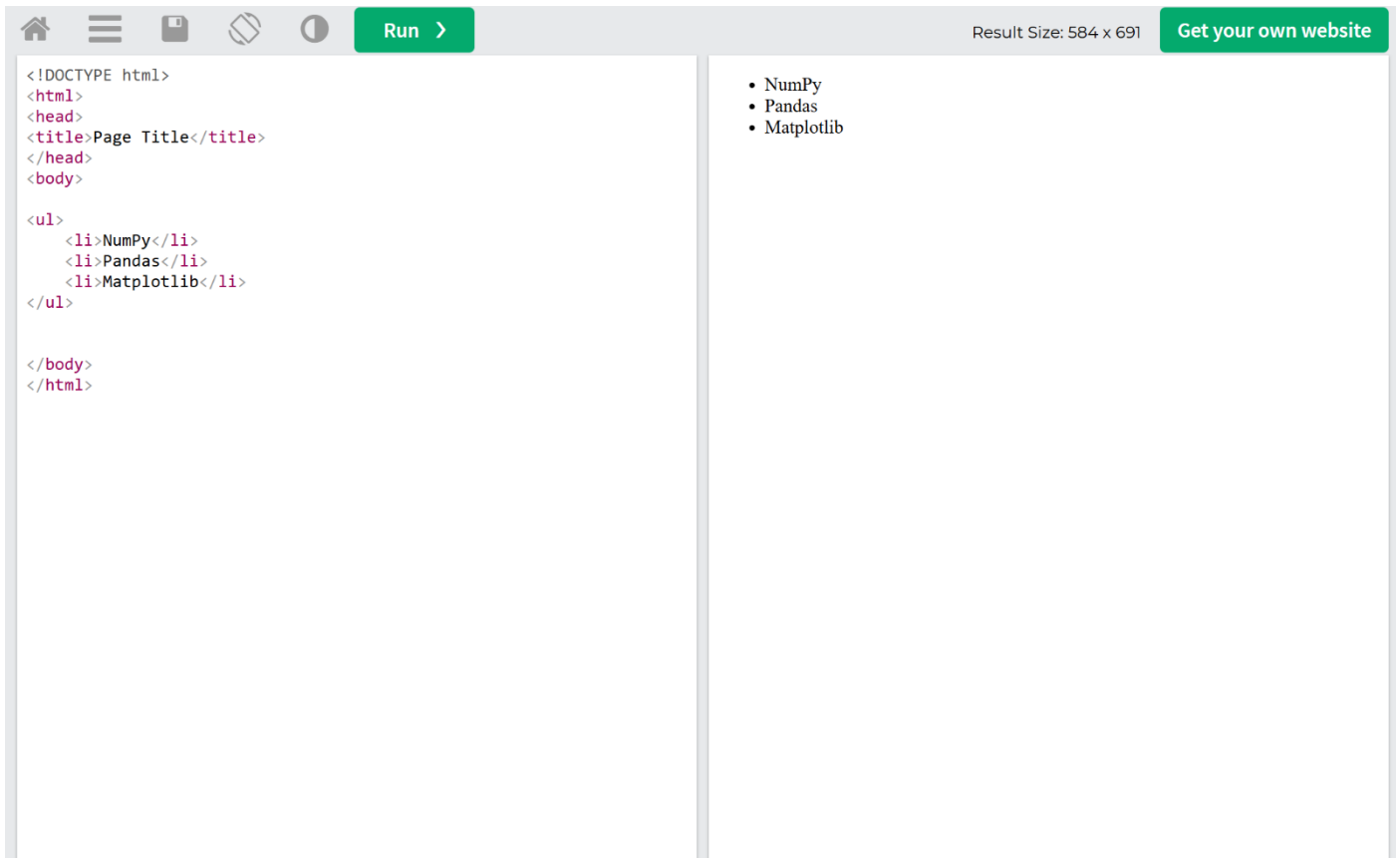


Lists

1. Unordered List ()

Used when the order **does not** matter (bullet points).

- **Parent Tag:**
- **Child Tag:** (List Item)



The screenshot shows a web editor interface. On the left, there is a code editor with the following HTML code:

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<ul>
  <li>NumPy</li>
  <li>Pandas</li>
  <li>Matplotlib</li>
</ul>

</body>
</html>
```

On the right, there is a preview window showing the rendered output of the code, which is an unordered list with three items:

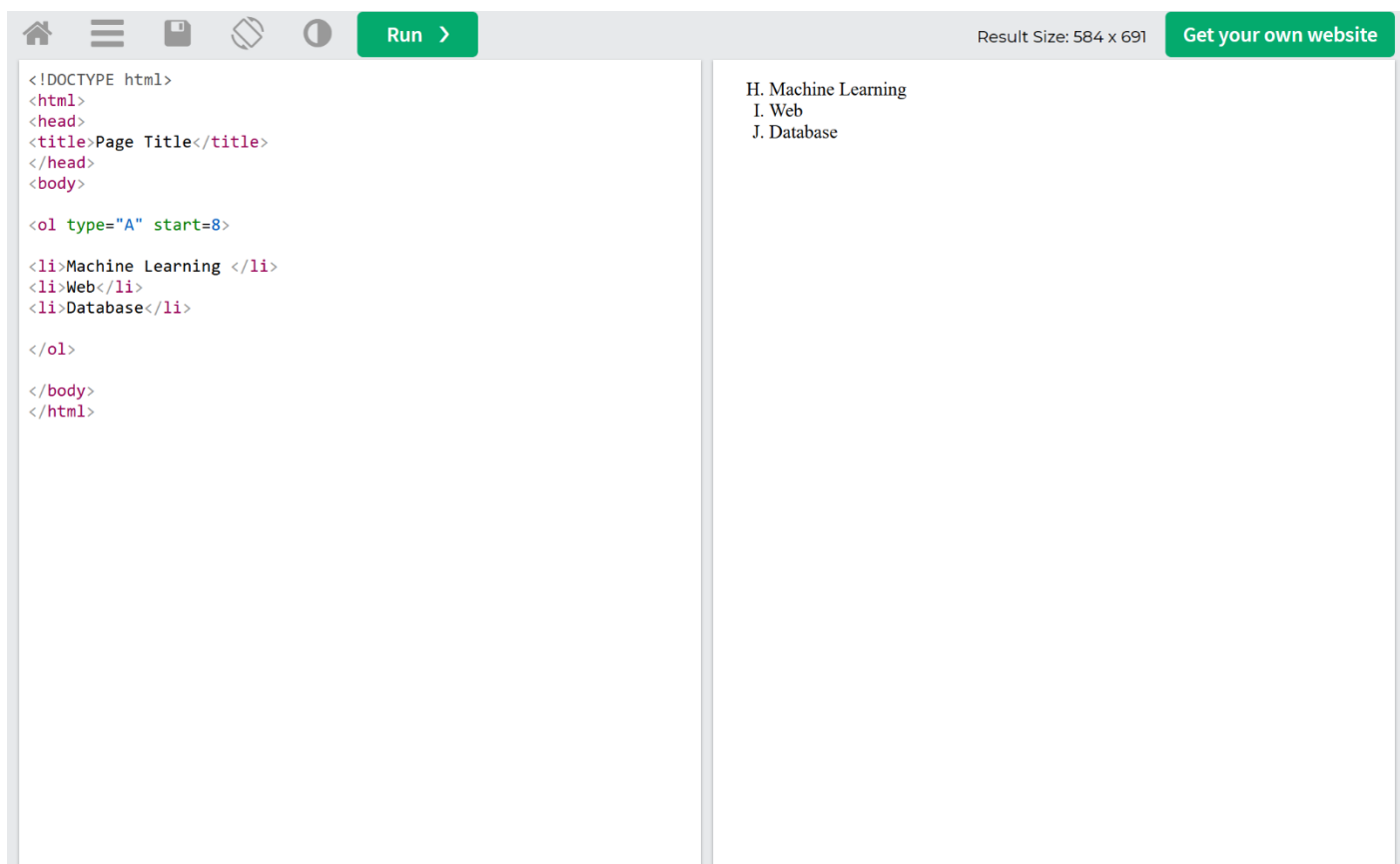
- NumPy
- Pandas
- Matplotlib

The editor interface includes a top bar with a 'Run' button and a 'Get your own website' button. The result size is indicated as 584 x 691.

2. Ordered List ()

Used when the order **does** matter (steps, ranking).

- **Parent Tag:**
- **Child Tag:** (List Item)
- **Attribute:** type (Controls numbering style)
 - "1" (Default: 1, 2, 3)
 - "A" (A, B, C)
 - "I" (Roman: I, II, III)
- **Attribute:** start (controls start of listing).



The screenshot shows a web browser interface. The top bar includes a home icon, a menu icon, a save icon, a refresh icon, and a 'Run' button. The browser title is 'Result Size: 584 x 691' and there is a 'Get your own website' button. The main content area is split into two panes. The left pane shows the HTML code for an ordered list:

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<ol type="A" start=8>
<li>Machine Learning </li>
<li>Web</li>
<li>Database</li>
</ol>

</body>
</html>
```

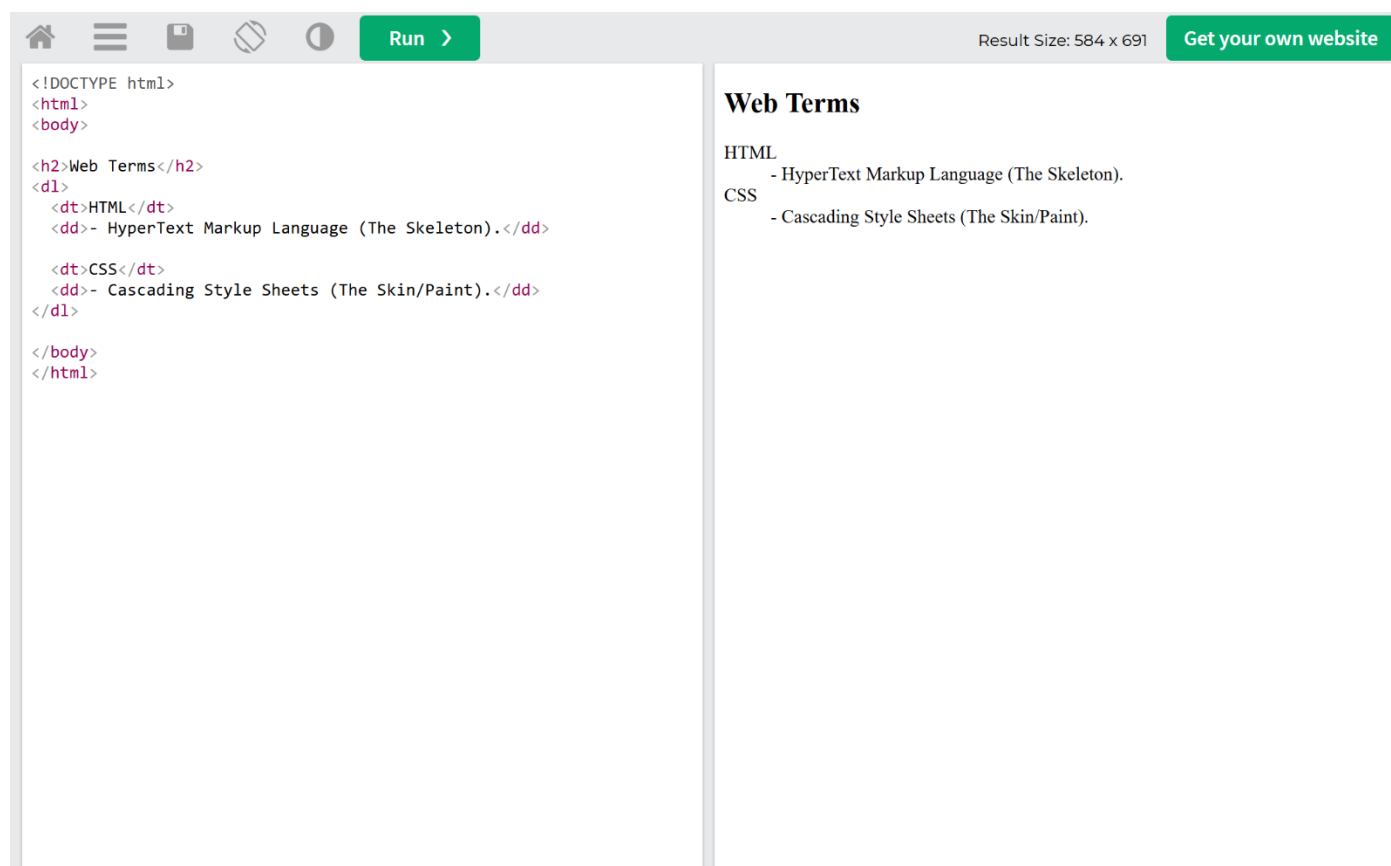
The right pane shows the rendered output of the HTML code, which is an ordered list with uppercase letters:

- H. Machine Learning
- I. Web
- J. Database

3. The Definition List (<dl>)

Used to display terms and their descriptions.

- **Parent Tag:** <dl> (Definition List)
- **Term Tag:** <dt> (Definition Term - The "Name")
- **Description Tag:** <dd> (Definition Description - The "Value")



The screenshot shows a web editor interface. On the left, the HTML code is displayed in a monospace font. On the right, the rendered output is shown. The code defines a definition list with two terms: HTML and CSS. The rendered output shows the title 'Web Terms' followed by the two terms and their descriptions.

```
<!DOCTYPE html>
<html>
<body>

<h2>Web Terms</h2>
<dl>
  <dt>HTML</dt>
  <dd>- HyperText Markup Language (The Skeleton).</dd>

  <dt>CSS</dt>
  <dd>- Cascading Style Sheets (The Skin/Paint).</dd>
</dl>

</body>
</html>
```

Web Terms

HTML
- HyperText Markup Language (The Skeleton).

CSS
- Cascading Style Sheets (The Skin/Paint).

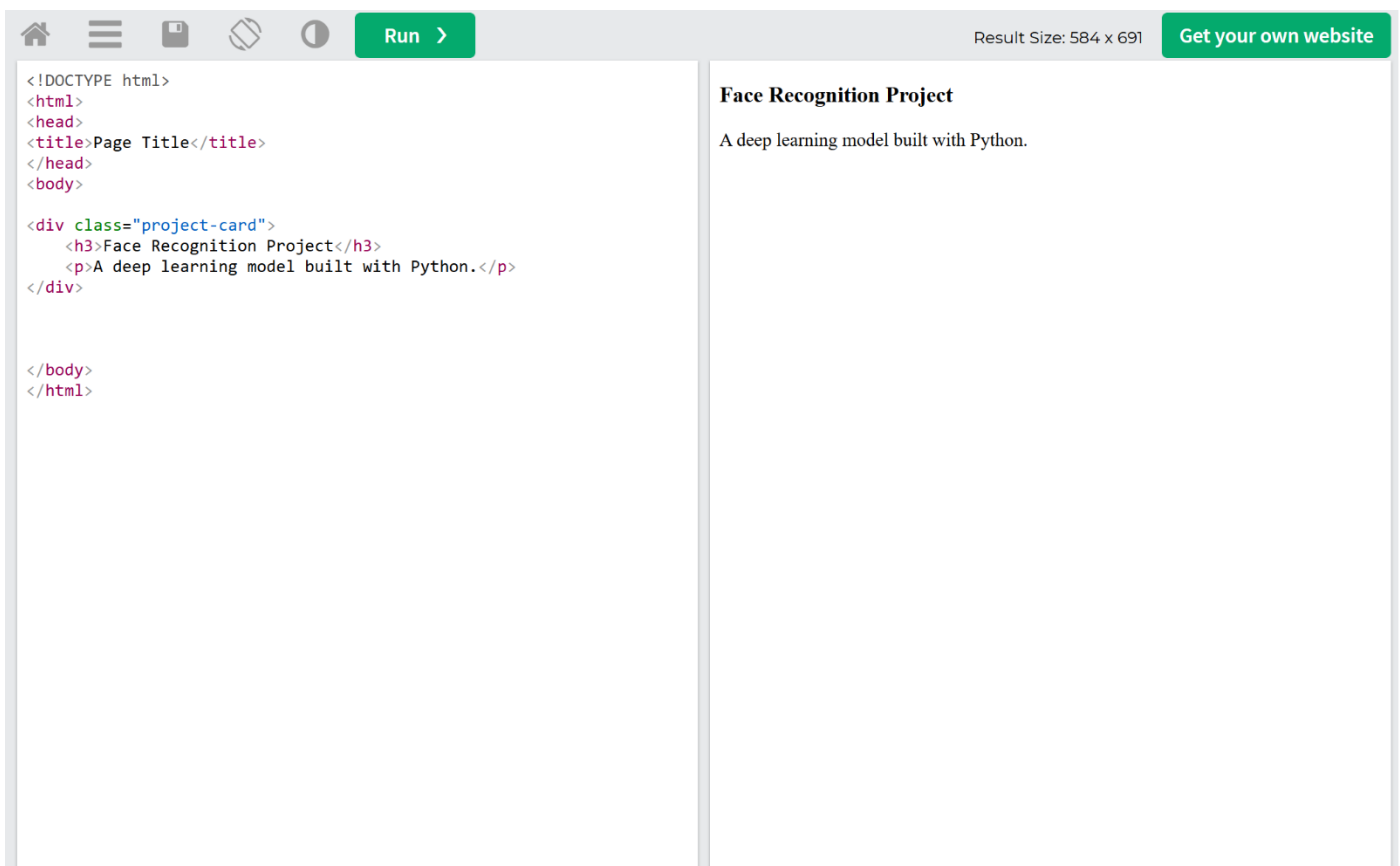
Layout & Grouping

These tags don't show anything themselves. They are **containers** used to group other tags so you can style them with CSS or manipulate them with JS.

1. The <div> (Block Container)

The **Division** tag. It is a "block-level" element, meaning it takes up the full width of the page and starts a new line.

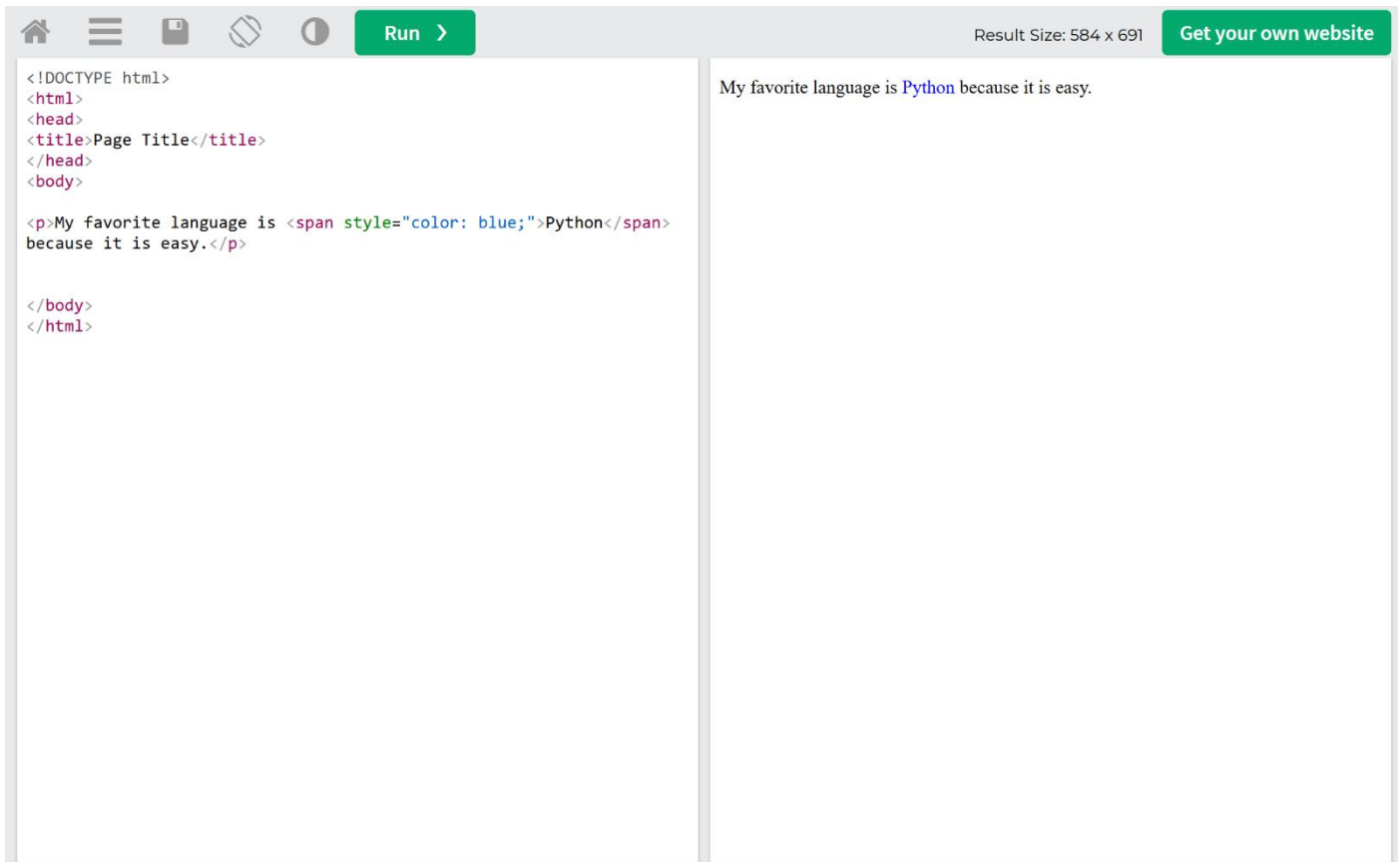
- **Use Case:** Grouping major sections (Header, Sidebar, Footer, Article Card).



2. The `` (Inline Container)

It is an "inline" element. It wraps a small piece of text *without* breaking the line.

- **Use Case:** Styling a specific word inside a paragraph (e.g., making one word red).



Media Tags (*img*, *audio*, *video*, *iframe*)

1. Images (**)

This is one of the most common tags. It embeds an image into the document.

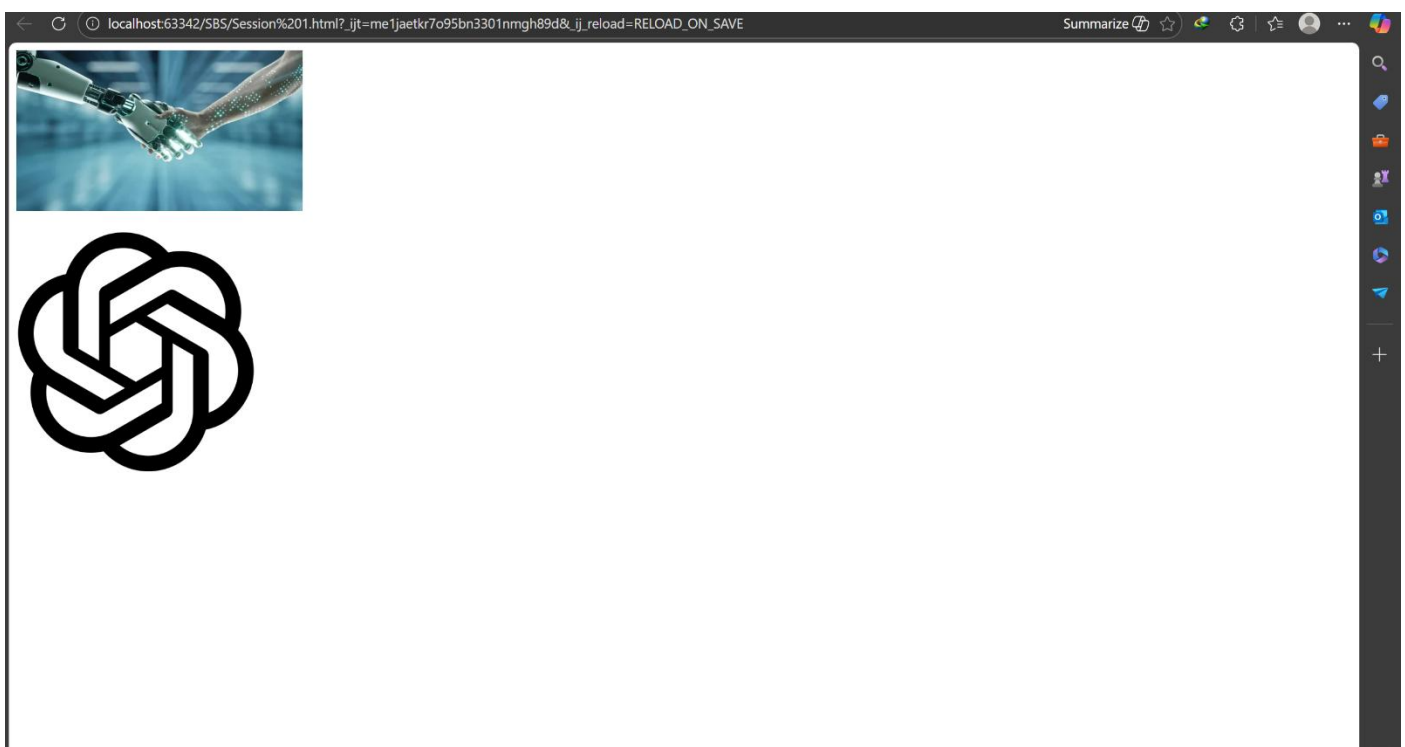
- **Tag:** `` (Self-closing).
- **Behavior:** It is an "inline" element (does not start a new line by default).

Attribute	Description	Important Values
src	Source. The path to the image file.	Relative path (images/logo.png) or URL (https://site.com/pic.jpg).
alt	Alternative Text. Shows if image breaks or for screen readers (blind users).	A description like "Diagram of Neural Network".
width	Width of the image.	Pixels (500) or Percentage (50%).
height	Height of the image.	Pixels (300). <i>Tip: Only set one (width OR height) to keep the aspect ratio.</i>
title	Tooltip text.	Shows when you hover the mouse over the image.
loading	Controls the delay of image loading	eager, lazy, auto

```
Project
├── SBS
│   ├── .venv
│   │   ├── library root
│   │   ├── Lib
│   │   ├── Scripts
│   │   ├── .gitignore
│   │   └── pyvenv.cfg
│   ├── assets
│   │   ├── chatgpt-logo.png
│   │   └── robot.webp
│   └── Session 1.html
├── External Libraries
└── Scratches and Consoles
```

```
<> Session 1.html x
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5 <meta charset="UTF-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 <title>My Study Document</title>
8 </head>
9
10 <body>
11 
12 <br><br>
13 <a href="https://openai.com">
14 
15 </a>
16
17 </body>
18
19 </html>
20
```

SBS > <> Session 1.html 20:1 CRLF UTF-8 2 spaces* Python 3.11 (SBS)



2. Audio (<audio>)

Used to play sound files like music, podcasts, or voice notes.

- **Tag:** `<audio>`
- **Supported Formats:** MP3 (Best), WAV, OGG.

Attribute	Description
controls	Crucial. Adds the Play/Pause buttons, volume, and seek bar. Without this, the audio is invisible!
autoplay	Starts playing immediately when page loads. (Browsers often block this to prevent annoyance).
loop	Replays the audio automatically when it ends.
muted	Starts the audio on mute.

The screenshot shows the PyCharm IDE interface. On the left, the 'Project' view displays a directory structure for 'SBS' located at 'C:\Users\Youssef Elkhatib\PycharmProjects\SBS'. The 'assets' folder is expanded, showing files like 'chatgpt-logo.png', 'robot.webp', 'World Cup 2010.mp3', and 'World-Cup 2014.mp3'. The main editor window shows the content of 'Session 1.html' with the following code:

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>My Study Document</title>
8 </head>
9
10 <body>
11   <audio controls loop>
12     <source src="assets/World%20Cup%202010.mp3" type="audio/mpeg">
13     Your browser does not support the audio element.
14   </audio>
15
16 </body>
17
18 </html>
19
```

The status bar at the bottom indicates the file encoding is UTF-8, line endings are CRLF, and the Python interpreter is Python 3.11 (SBS).



3. Video (<video>)

Used to embed video clips directly (not YouTube links, that uses <iframe> which is different).

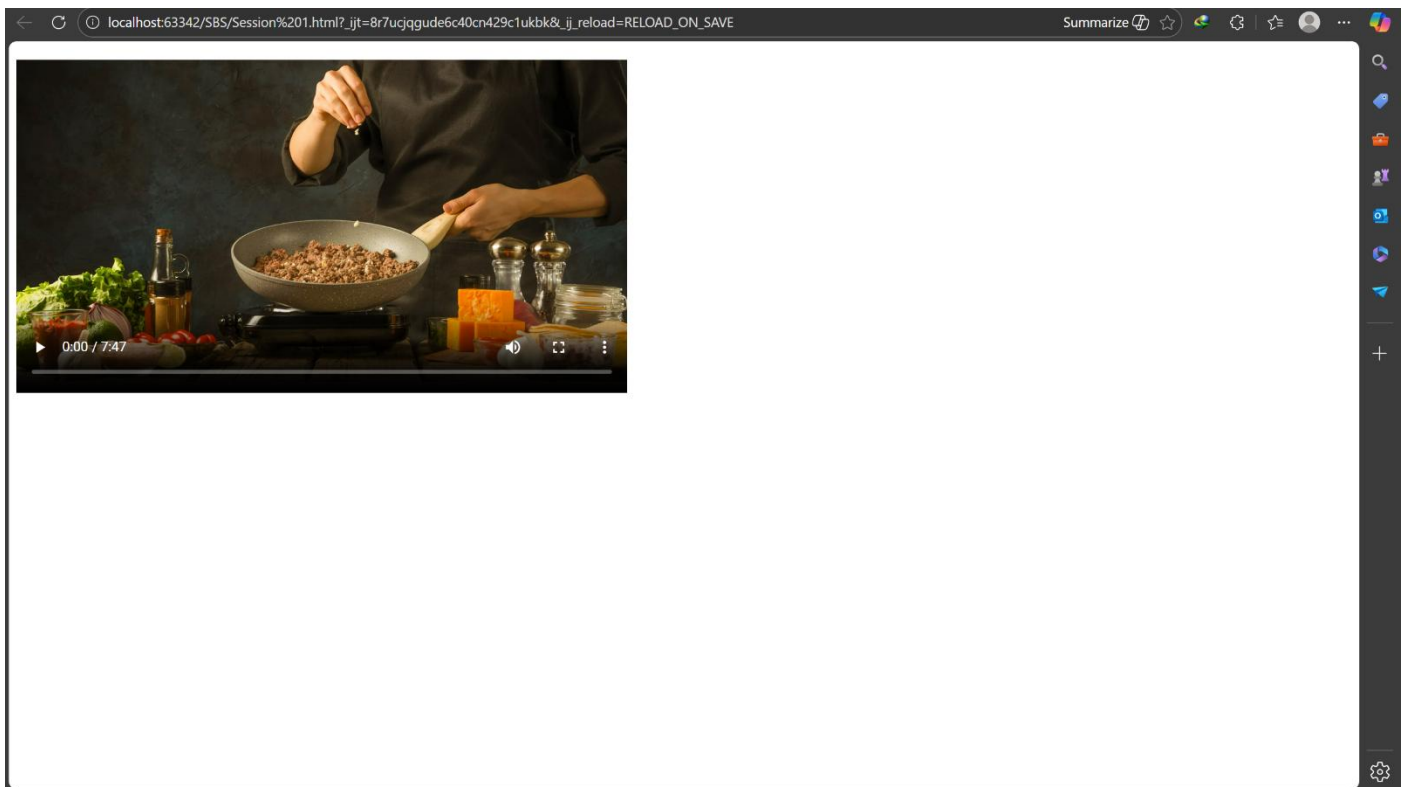
- **Tag:** <video>
- **Supported Formats:** MP4 (Best), WebM.

Attribute	Description
controls	Adds Play/Pause, Fullscreen, Volume.
width / height	Sets the size of the video player window.
poster	Thumbnail. An image shown before the video plays.
autoplay, loop, muted	Same as audio. <i>Note: autoplay usually only works if muted is also present.</i>

```
Project
├── SBS
│   ├── .venv library root
│   │   ├── Lib
│   │   ├── Scripts
│   │   ├── .gitignore
│   │   └── pyvenv.cfg
│   ├── assets
│   │   ├── C&C.jpg
│   │   ├── chatgpt-logo.png
│   │   ├── Click_and_Cook.mp4
│   │   ├── robot.webp
│   │   ├── World Cup 2010.mp3
│   │   └── World-Cup 2014.mp3
│   └── Session 1.html
├── External Libraries
└── Scratches and Consoles
```

```
<> Session 1.html x
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5 <meta charset="UTF-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 <title>My Study Document</title>
8 </head>
9
10 <body>
11 <video width="640" height="360" controls poster="assets/C&C.jpg">
12 <source src="assets/Click_and_Cook.mp4" type="video/mp4">
13 Your browser does not support the video tag.
14 </video>
15
16 </body>
17
18 </html>
19
```

SBS > <> Session 1.html 19:1 CRLF UTF-8 2 spaces* Python 3.11 (SBS)



4. The Inline Frame (<iframe>)

Embeds another HTML page within the current page. The uploaded document mentioned that for YouTube videos, you use <iframe> rather than <video>¹.

- **Tag:** <iframe>

Attribute	Description	Important Values
src	Source. The URL of the page to show.	URL (e.g., https://www.wikipedia.org).
width / height	Sets the size of the frame.	Pixels (500) or Percentage (100%).
title	Describes the content (Critical for Screen Readers).	Text (e.g., Map of Cairo).
allowfullscreen	Allows the iframe to go full screen (e.g., videos).	allowfullscreen

The screenshot shows a web development tool interface. On the left, the HTML code is displayed in a text editor. The code includes the following elements:

```
<!DOCTYPE html>
<html>
<body>

<h2>Embedded Wikipedia</h2>
<iframe
  src="https://www.wikipedia.org"
  width="100%"
  height="300"
  title="Wikipedia Home">
</iframe>

</body>
</html>
```

On the right, the rendered preview shows the title "Embedded Wikipedia" above a dark-themed Wikipedia homepage. The homepage features the Wikipedia logo and the text "The Free Encyclopedia". Below this, there are several language options with their respective article counts:

- English:** 7,102,000+ articles
- العربية:** +1,288,000 مقالة
- 日本語:** 1,483,000+ 記事
- Русский:** 2 074 000+ статей
- Deutsch:** 3,075.000+ Artikel
- Français:** 2 724 000+ articles

The interface also includes a "Run" button and a "Get your own website" button. The result size is indicated as 584 x 691.

5. Note on File Paths (The most common error!)

When using src, you need to tell the HTML file exactly where to look.

1. **Same Folder:** src="image.jpg"
2. **Inside a Subfolder:** src="images/image.jpg" (Go into 'images' folder, then find file).
3. **Parent Folder:** src="../image.jpg" (Go **back** one folder, then find file).
4. **Absolute URL:** src="https://google.com/logo.png" (Go to the internet).

Tables (<table>)

Tables are essential for displaying structured data like financial reports, calendars, or AI model comparisons. A professional table isn't just a grid of boxes; it has semantic meaning (Header, Body, Footer) and flexible layout capabilities.

1. The Container & Semantics

We don't just throw data into a pile. We organize it into logical sections.

- **<table>**: The main wrapper.
 - **<caption>**: A title for the table. **Must be the first tag** inside the table.
 - **<thead>**: **Table Head**. Contains the column titles. It prints on every page if you print the document.
 - **<tbody>**: **Table Body**. Contains the actual data rows.
 - **<tfoot>**: **Table Footer**. Contains summaries (like "Total" or "Average").
-

2. The Rows & Cells

Inside the semantic sections, we build the grid row by row.

- **<tr>**: **Table Row**. Starts a new horizontal line.
 - **<th>**: **Table Header Cell**. By default, text inside here is **Bold** and **Centered**. Used for titles.
 - **<td>**: **Table Data Cell**. The standard cell for information.
-

3. Merging & Sizing (The Magic Attributes)

Sometimes a cell needs to be bigger than the others. We use attributes to merge them.

- **colspan="x"**: Stretches a cell **Horizontally** across x columns.
- **rowspan="y"**: Stretches a cell **Vertically** down y rows.
- **width / height**: Sets the size (e.g., width="50%" or height="200").

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Advanced Table Structure</title>
</head>
<body>

  <table border="1" width="80%" cellpadding="10">

    <caption><strong> AI Model Performance Report</strong>
</caption>

    <thead style="background-color: #e3f2fd;">
      <tr>
        <th rowspan="2" width="15%">Domain</th>

        <th rowspan="2" width="25%">Algorithm</th>

        <th colspan="2">Metrics</th>
      </tr>
      <tr>
        <th>Accuracy</th>
        <th>Training Time</th>
      </tr>
</thead>

    <tbody>
      <tr>
        <td rowspan="2" align="center" style="background-color: #fff9c4;">
          <strong>Computer Vision </strong>
        </td>

        <td>CNN (Custom)</td>
        <td>92.5%</td>
        <td>4 Hours</td>
      </tr>

```

Domain	Algorithm	Metrics	
		Accuracy	Training Time
Computer Vision	CNN (Custom)	92.5%	4 Hours
	YOLO v8	94.2%	6 Hours
NLP	LSTM (RNN)	88.0%	3 Hours
	BERT Transformer	99.1%	12 Hours
Global Average:		93.45%	6.25 Hours

```

      <td>YOLO v8</td>
      <td>94.2%</td>
      <td>6 Hours</td>
    </tr>

    <tr>
      <td rowspan="2" align="center" style="background-color: #ffc0cb;">
        <strong>NLP </strong>
      </td>

      <td>LSTM (RNN)</td>
      <td>88.0%</td>
      <td>3 Hours</td>
    </tr>
    <tr>
      <td>BERT Transformer</td>
      <td>99.1%</td>
      <td>12 Hours</td>
    </tr>
  </tbody>

  <tfoot style="background-color: #e0e0e0;">
    <tr>
      <td colspan="2" align="right"><strong>Global
Average:</strong></td>

      <td><strong>93.45%</strong></td>
      <td><strong>6.25 Hours</strong></td>
    </tr>
  </tfoot>
</table>

</body>
</html>

```

Domain	Algorithm	Metrics	
		Accuracy	Training Time
Computer Vision	CNN (Custom)	92.5%	4 Hours
	YOLO v8	94.2%	6 Hours
NLP	LSTM (RNN)	88.0%	3 Hours
	BERT Transformer	99.1%	12 Hours
Global Average:		93.45%	6.25 Hours

Data Collection (Forms)

1. The Container (<form>)

Just as <table> wraps rows and cells, the <form> tag wraps all the input fields. It defines **where** the data goes and **how** it gets there.

- **Tag:** <form>

Attribute	Description	Important Values
action	Destination. The URL where the data is sent when the user clicks submit.	URL (e.g., /submit-login.php) or leave empty to reload current page.
method	The Delivery Truck. How the data travels.	"GET" : Data is visible in the URL (Good for Search). "POST" : Data is hidden inside the body (Good for Passwords/Secure data).
target	Where to open the response.	_blank (New Tab), _self (Same Tab).
autocomplete	Browser remembers previous answers.	on (Default), off (For banking/sensitive forms).

2. The Connection (<label>)

The <label> tag defines a caption for an item in the user interface.

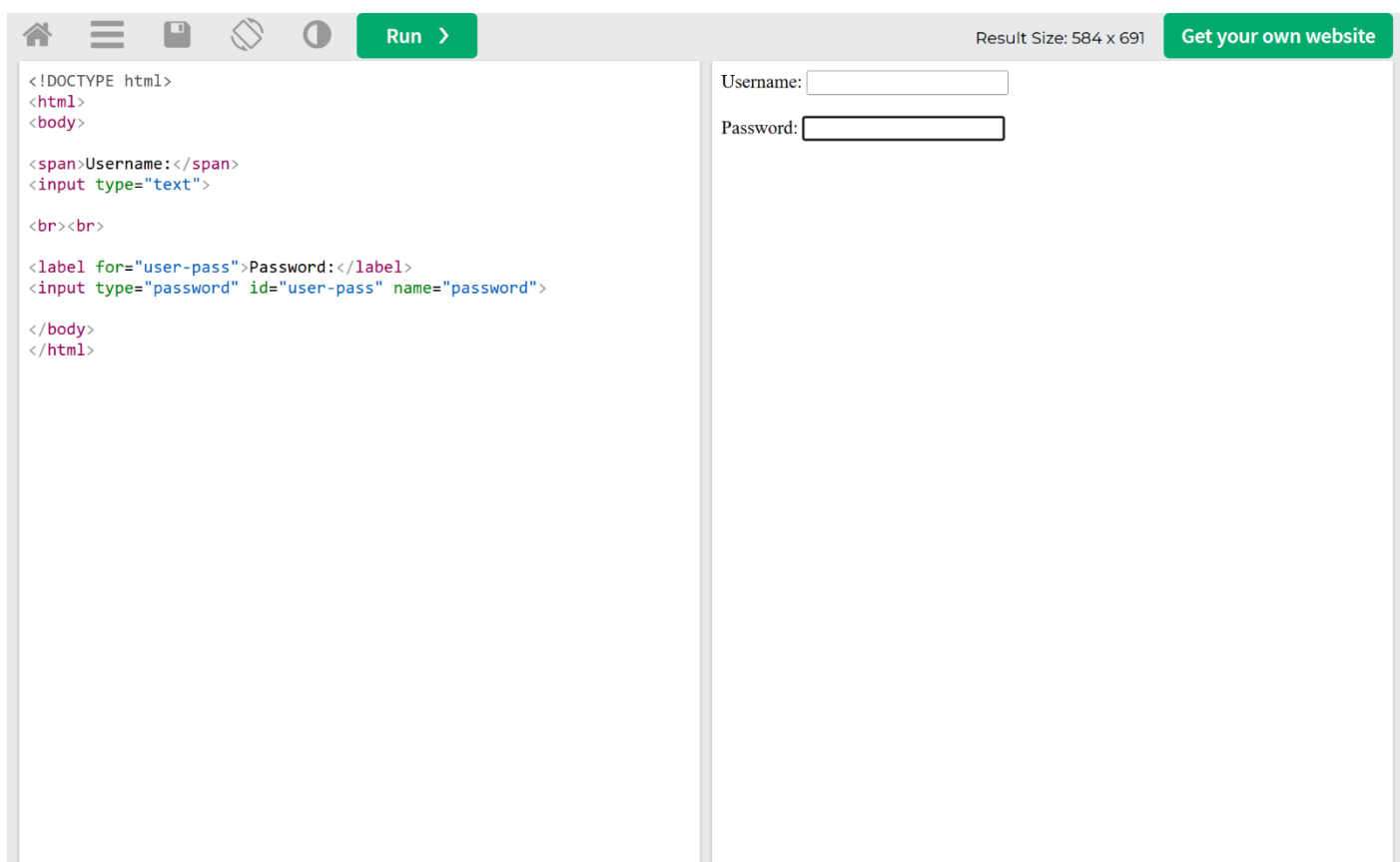
- **Tag:** <label>
- **The Magic Connection:** If you click on the text inside a label, the browser automatically "focuses" (activates) the connected input box.

The for Attribute (Crucial)

To connect a label to an input, the for attribute of the label must match the id attribute of the input.

Note: Do not confuse id with name.

- **id:** Used by CSS, JavaScript, and Labels (Client-side).
- **name:** Used by the Server to identify the data (Server-side).



The screenshot displays a web browser interface. On the left, the HTML source code is visible, showing the following structure:

```
<!DOCTYPE html>
<html>
<body>

<span>Username:</span>
<input type="text">

<br><br>

<label for="user-pass">Password:</label>
<input type="password" id="user-pass" name="password">

</body>
</html>
```

On the right, the rendered form is shown. It consists of two rows of labels and input fields. The first row has the label "Username:" followed by a text input field. The second row has the label "Password:" followed by a password input field. The browser's address bar shows "Result Size: 584 x 691" and a "Get your own website" button.

3. The Info (<input>)

The <input> tag is the most powerful tag in forms. It can change its shape and behavior completely based on the type attribute.

- **Tag:** <input> (Self-closing).

Attribute	Description
name	Mandatory. If an input has no name, the data is not sent to the server.
value	The default text inside the input (or the button text).
placeholder	A grey hint that disappears when you type (e.g., "Enter your email").
required	Prevents the user from submitting if empty.
readonly	The user can copy the text but cannot change it.
disabled	The input is grayed out, and its data is not sent.
autofocus	Focuses when the page is loaded
minlength / maxlength	Determine min and max input length

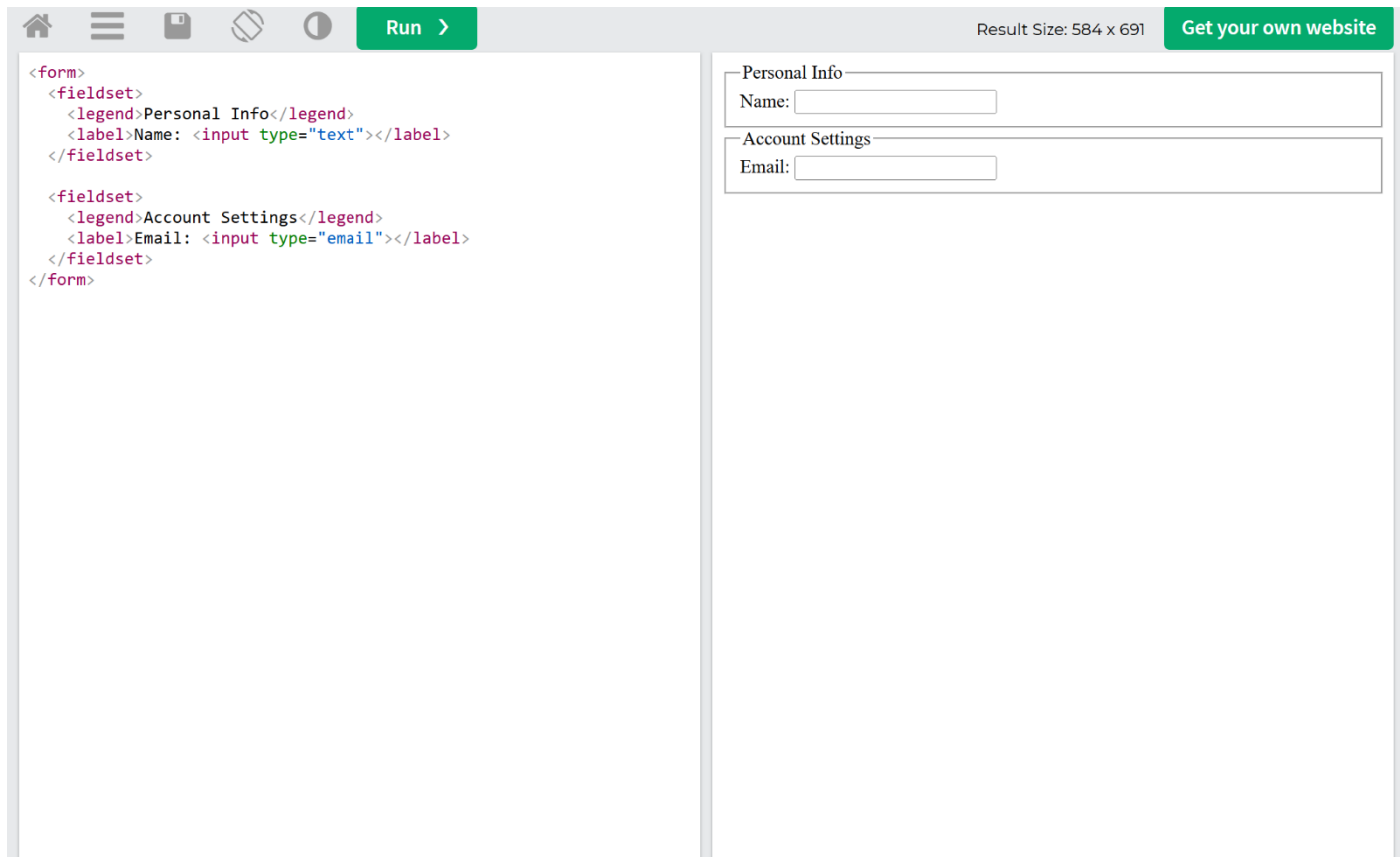
Input Type Values

Type	Description	Visual Effect
text	Standard one-line text.	A simple box.
password	Hides characters as dots or asterisks (••••).	A secure box.
email	Validates that text contains "@".	Mobile keyboards show "@" button.
number	Numeric only. Prevents letters.	Shows up/down spinner arrows.
range	Slider. Selects a value within a range (e.g., 1-100).	A draggable slider bar.
color	Color Picker. Selects a hex color code.	A color square/palette popup.
search	Optimized for search queries.	Adds an "X" to clear text quickly.
url	Validates that text is a link (http://...).	Mobile keyboards show ".com" button.
radio	Pick One. Mutually exclusive choices (Male/Female).	Round circle.
checkbox	Pick Many. Select zero or more options.	Square box.
date	Full date picker (Day, Month, Year).	Opens a calendar popup.
month	Selects only Month and Year.	A specific month/year picker.
time	Selects Time (Hours and Minutes).	A clock or time dial popup.
file	Uploads a document or image.	"Choose File" button.
button	Dead Button. Does nothing until you add JavaScript.	A standard clickable button.
image	Graphical Submit. Uses an image instead of text to submit.	The image acts as the button.
submit	Submit Button. Sends the form data to the server.	A clickable button (default text: "Submit").
reset	Clear Form. Reverts all inputs to their original state (empty or default value).	A clickable button (Default text: "Reset").

4. Grouping & Structure (<fieldset> & <legend>)

When a form becomes too long, it is good practice to group related data together.

- **Tag: <fieldset>** (The container)
- **Tag: <legend>** (The caption)



The screenshot shows a web browser interface. On the left, there is a code editor displaying HTML code for a form. On the right, the rendered form is displayed. The form is divided into two sections: 'Personal Info' and 'Account Settings'. The 'Personal Info' section contains a 'Name' input field. The 'Account Settings' section contains an 'Email' input field. The browser's address bar shows 'Run >' and the result size is '584 x 691'. A green button in the top right corner says 'Get your own website'.

```
<form>
  <fieldset>
    <legend>Personal Info</legend>
    <label>Name: <input type="text"></label>
  </fieldset>

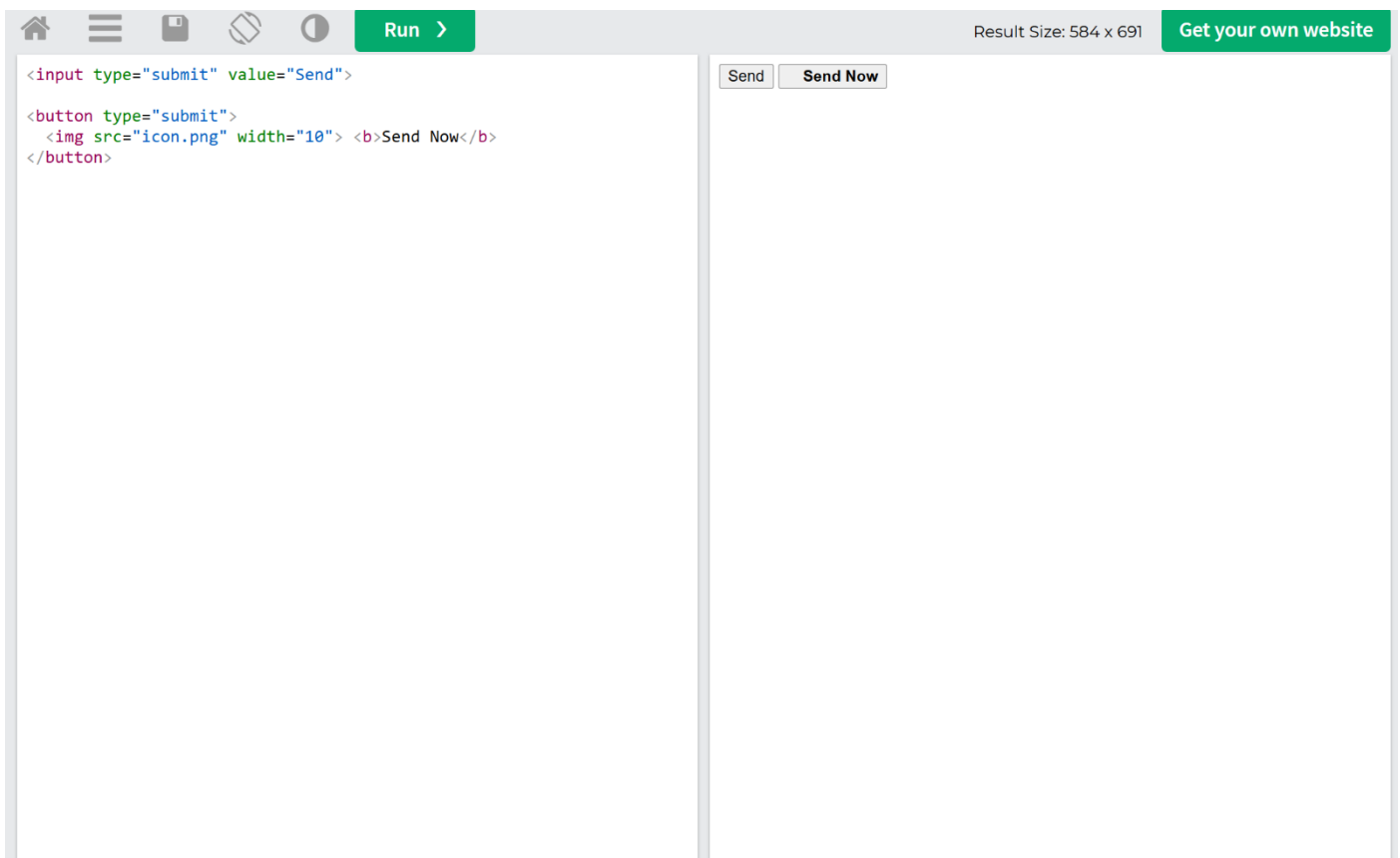
  <fieldset>
    <legend>Account Settings</legend>
    <label>Email: <input type="email"></label>
  </fieldset>
</form>
```

5. Miscellaneous

The <button> Tag

We already saw <input type="button">, but there is a newer, better tag: <button>.

- **Tag:** <button>
- **Difference:** Unlike <input>, the <button> tag has an **opening and closing tag**. This means you can put **content** inside it like images, bold text, or icons.
- **Types:** It also has a type attribute (**submit, reset, or button**).



6. Code summing it all

Result Size: 584 x 691 [Get your own website](#)

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>The Ultimate Form</title>
</head>
<body>

  <h2>Job Application Portal</h2>

  <form action="/submit-application.php" method="POST"
  autocomplete="on">

    <fieldset>
      <legend>🔒 Account Details</legend>

      <label for="uname">Username (Text):</label><br>
      <input type="text" id="uname" name="username"
      placeholder="user123" required autofocus>
      <br><br>

      <label for="pass">Password:</label><br>
      <input type="password" id="pass" name="password"
      required minlength="8">
      <br><br>

      <label for="email">Email:</label><br>
      <input type="email" id="email" name="user_email"
      placeholder="you@mail.com">
    </fieldset>
    <br>

    <fieldset>
      <legend>👤 Personal Info</legend>

      <label for="fullname">Full Name:</label><br>
      <input type="text" id="fullname" name="fullname">
      <br><br>
  
```

Job Application Portal

🔒 Account Details

Username (Text):
user123

Password:
[input type="password"]

Email:
you@mail.com

👤 Personal Info

Full Name:
[input type="text"]

Date of Birth (Date):
mm / dd / yyyy 📅

Upload Resume (File):
Choose File No file chosen

Gender (Radio):
 Male Female

📊 Skills & Stats

Experience Level (Range):
0 [range slider] 10+

Desired Role (Dropdown):
Developer

Result Size: 584 x 691 [Get your own website](#)

```

<legend>👤 Personal Info</legend>

<label for="fullname">Full Name:</label><br>
<input type="text" id="fullname" name="fullname">
<br><br>

<label for="dob">Date of Birth (Date):</label><br>
<input type="date" id="dob" name="birthdate">
<br><br>

<label for="photo">Upload Resume (File):</label><br>
<input type="file" id="photo" name="resume">
<br><br>

<p>Gender (Radio):</p>
<input type="radio" id="m" name="gender" value="male">
<label for="m">Male</label>
<input type="radio" id="f" name="gender"
value="female">
<label for="f">Female</label>
</fieldset>
<br>

<fieldset>
  <legend>📊 Skills & Stats</legend>

  <label for="exp">Experience Level (Range):</label><br>
  0 <input type="range" id="exp" name="years_exp" min="0"
max="10"> 10+
  <br><br>

  <label for="role">Desired Role (Dropdown):</label><br>
  <select id="role" name="job_role">
    <optgroup label="Tech">
      <option value="dev">Developer</option>
      <option value="ds">Data Scientist</option>
    </optgroup>
    <optgroup label="Design">
      <option value="ui">UI Designer</option>
  
```

👤 Personal Info

Full Name:
[input type="text"]

Date of Birth (Date):
mm / dd / yyyy 📅

Upload Resume (File):
Choose File No file chosen

Gender (Radio):
 Male Female

📊 Skills & Stats

Experience Level (Range):
0 [range slider] 10+

Desired Role (Dropdown):
Developer

Preferred IDE (Datalist):
[input type="text"]

Cover Letter (Textarea):
why hire you?

⚙️ Misc Inputs

Color: [color picker] Search: [input type="text"] URL: [input type="text"]

Time: [time picker] Month: [month picker]

Result Size: 584 x 691 Get your own website

```

<legend> Skills & Stats</legend>

<label for="exp">Experience Level (Range):</label><br>
0 <input type="range" id="exp" name="years_exp" min="0"
max="10"> 10+
<br><br>

<label for="role">Desired Role (Dropdown):</label><br>
<select id="role" name="job_role">
  <optgroup label="Tech">
    <option value="dev">Developer</option>
    <option value="ds">Data Scientist</option>
  </optgroup>
  <optgroup label="Design">
    <option value="ui">UI Designer</option>
  </optgroup>
</select>
<br><br>

<label for="browser">Preferred IDE (Datalist):</label>
<br>
<input list="ides" id="browser" name="ide_choice">
<datalist id="ides">
  <option value="VS Code">
  <option value="PyCharm">
  <option value="Sublime">
</datalist>
<br><br>

<label for="bio">Cover Letter (Textarea):</label><br>
<textarea id="bio" name="cover_letter" rows="3"
cols="30" placeholder="Why hire you?"></textarea>
</fieldset>
<br>

<fieldset>
  <legend> Misc Inputs</legend>
  <label>Color: <input type="color" name="favcolor">
  </label>
</fieldset>

```

Date of Birth (Date):
mm/dd/yyyy

Upload Resume (File):
Choose File No file chosen

Gender (Radio):
 Male Female

Skills & Stats

Experience Level (Range):
0 10+

Desired Role (Dropdown):
Developer

Preferred IDE (Datalist):

Cover Letter (Textarea):
why hire you?

Misc Inputs

Color: Search: URL:
Time: Month:

Number:

Clear Form Submit Application Save Draft

Result Size: 584 x 691 Get your own website

```

</datalist>
<br><br>

<label for="bio">Cover Letter (Textarea):</label><br>
<textarea id="bio" name="cover_letter" rows="3"
cols="30" placeholder="Why hire you?"></textarea>
</fieldset>
<br>

<fieldset>
  <legend> Misc Inputs</legend>
  <label>Color: <input type="color" name="favcolor">
  </label>
  <label>Search: <input type="search" name="q"></label>
  <label>URL: <input type="url" name="portfolio"></label>
  <label>Time: <input type="time" name="avail_time">
</label>
  <label>Month: <input type="month" name="start_month">
</label>
  <label>Number: <input type="number" name="salary"
step="100"></label>
</fieldset>
<br>

<input type="reset" value="Clear Form">

<input type="submit" value="Submit Application">

<button type="button" onclick="alert('Saved!')">
  Save Draft
</button>
<br><br>
</form>
</body>
</html>

```

Date of Birth (Date):
mm/dd/yyyy

Upload Resume (File):
Choose File No file chosen

Gender (Radio):
 Male Female

Skills & Stats

Experience Level (Range):
0 10+

Desired Role (Dropdown):
Developer

Preferred IDE (Datalist):

Cover Letter (Textarea):
why hire you?

Misc Inputs

Color: Search: URL:
Time: Month:

Number:

Clear Form Submit Application Save Draft

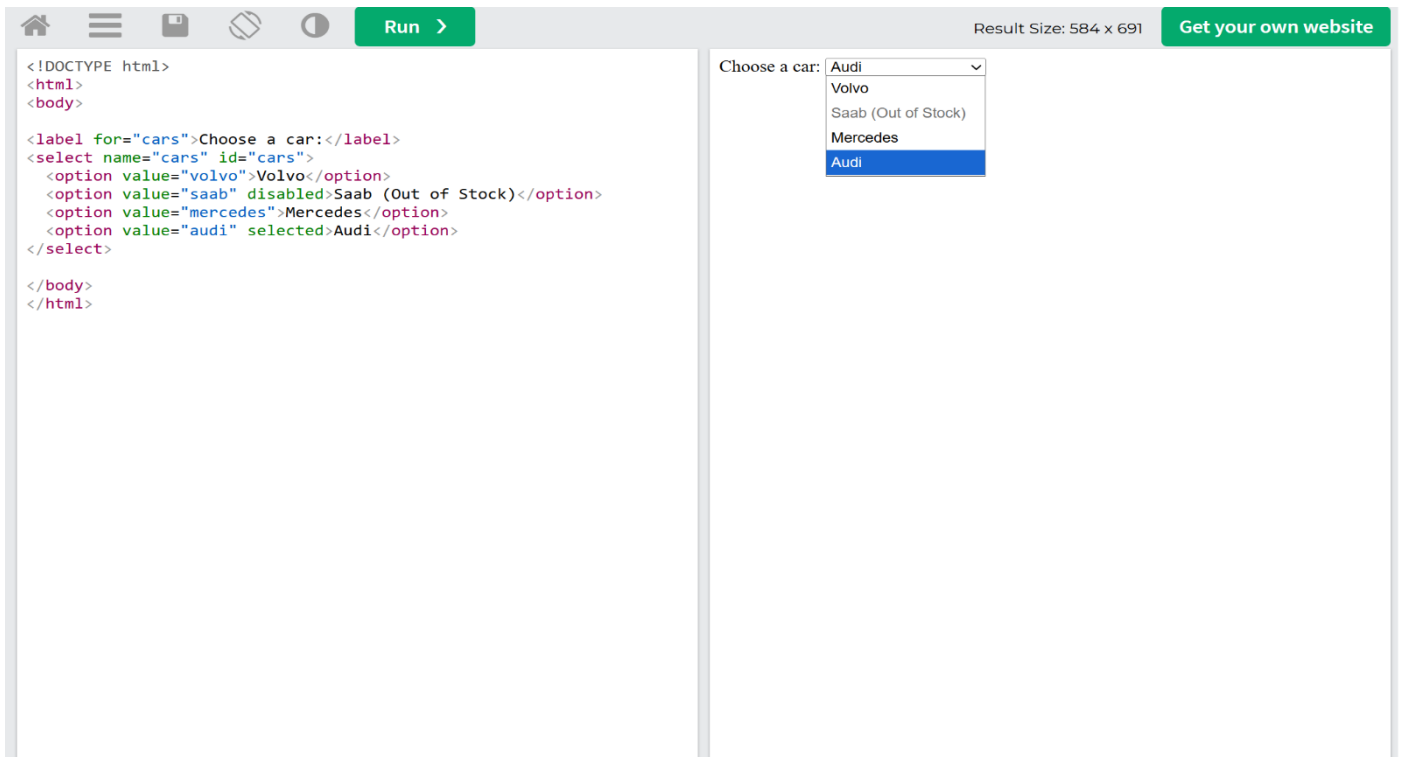
Advanced Input Controls

1. The Dropdown List (<select>)

Used when you want the user to select one (or more) options from a fixed list. The user **must** choose what you provide; they cannot type a new answer.

- **Tag:** <select> (Parent)
- **Child Tag:** <option> (The items)

Attribute	Placed In	Description	Important Values
name	<select>	Identifies the data when sent to backend.	Any text (e.g., name="cars").
multiple	<select>	Allows the user to select more than one option (hold Ctrl/Command).	multiple
value	<option>	The actual data sent to the server (often different from what the user sees).	value="volvo"
selected	<option>	Pre-selects an option by default.	selected
disabled	<option>	Shows the item but prevents clicking it.	disabled



The screenshot shows a web browser interface with a dropdown menu. The dropdown is titled "Choose a car:" and is open, displaying a list of car models: Audi, Volvo, Saab (Out of Stock), Mercedes, and Audi. The first "Audi" option is highlighted in blue, indicating it is the selected item. The browser's address bar shows "Result Size: 584 x 691" and a "Get your own website" button. The code editor on the left shows the following HTML code:

```
<!DOCTYPE html>
<html>
<body>

<label for="cars">Choose a car:</label>
<select name="cars" id="cars">
  <option value="volvo">Volvo</option>
  <option value="saab" disabled>Saab (Out of Stock)</option>
  <option value="mercedes">Mercedes</option>
  <option value="audi" selected>Audi</option>
</select>

</body>
</html>
```

2. The Data List (<datalist>)

Provides an "autocomplete" feature for an input field.

- **Tag:** `<input>` (Visible field) + `<datalist>` (Hidden list)
- **Child Tag:** `<option>` (Suggestions)
- **Difference from <select>:** The user is **flexible**. They can pick a suggestion OR type their own value.

Attribute	Placed In	Description	Important Values
list	<code><input></code>	Crucial: Links the input box to the datalist.	Must match the id of the datalist.
id	<code><datalist></code>	Identifies the list so the input can find it.	Must match the list attribute of input.
value	<code><option></code>	The suggested text.	Any text (e.g., Chrome).

```
<!DOCTYPE html>
<html>
<body>

<label for="browser">Choose your browser:</label>
<input list="browsers" name="browser" id="browser" placeholder="Type
or click...">

<datalist id="browsers">
  <option value="Edge">
  <option value="Firefox">
  <option value="Chrome">
  <option value="Opera">
  <option value="Safari">
</datalist>

</body>
</html>
```

3. The Text Area (<textarea>)

A multi-line text input control. Used for comments, reviews, or essays.

- **Tag:** <textarea>
- **Note:** Default text goes *between* the tags, not in a value attribute.

Attribute	Description	Important Values
rows	Specifies the visible height (number of lines).	Number (e.g., rows="4").
cols	Specifies the visible width (average character width).	Number (e.g., cols="50").
placeholder	Grey hint text shown when empty.	Text (e.g., Write review here...).
readonly	User can read but not edit the text.	readonly
maxlength	Limits the number of characters.	Number (e.g., maxlength="200").

The screenshot shows a web development environment with a code editor on the left and a preview window on the right. The code editor contains the following HTML code:

```
<!DOCTYPE html>
<html>
<body>

<p>Write your feedback:</p>
<textarea name="message" rows="4" cols="30" placeholder="Type here...">
This is default text.
</textarea>

</body>
</html>
```

The preview window displays the rendered output, which includes the text "Write your feedback:" followed by a text area containing the text "This is default text." and a green circular logo.

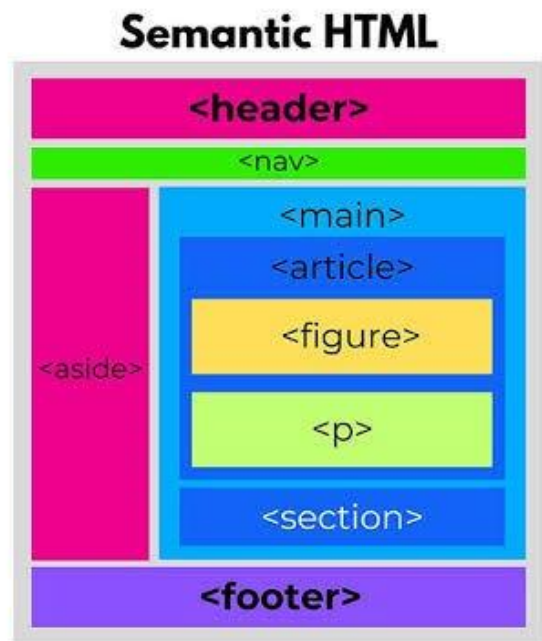
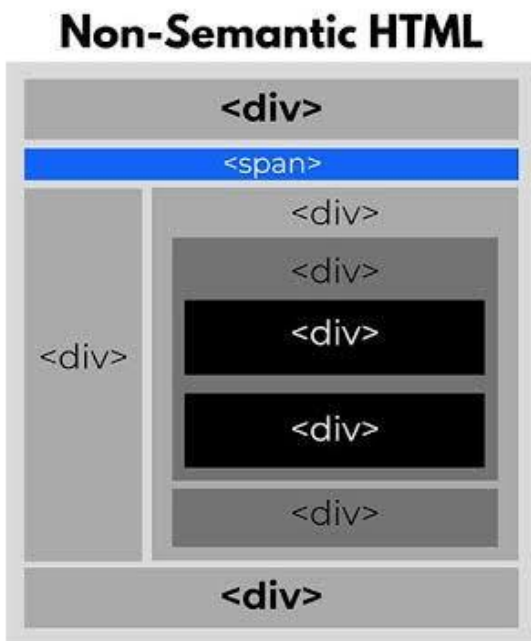
HTML Semantics

HTML **semantic elements** are tags that describe the meaning and purpose of the content they contain to both developers and browsers. This provides a clear, logical structure to a webpage, which improves accessibility for assistive technologies (like screen readers), enhances Search Engine Optimization (SEO), and makes code easier to read and maintain.

Why Use Semantic Elements?

Non-semantic elements like `<div>` and `` tell the browser nothing about their content. In contrast, semantic elements provide context:

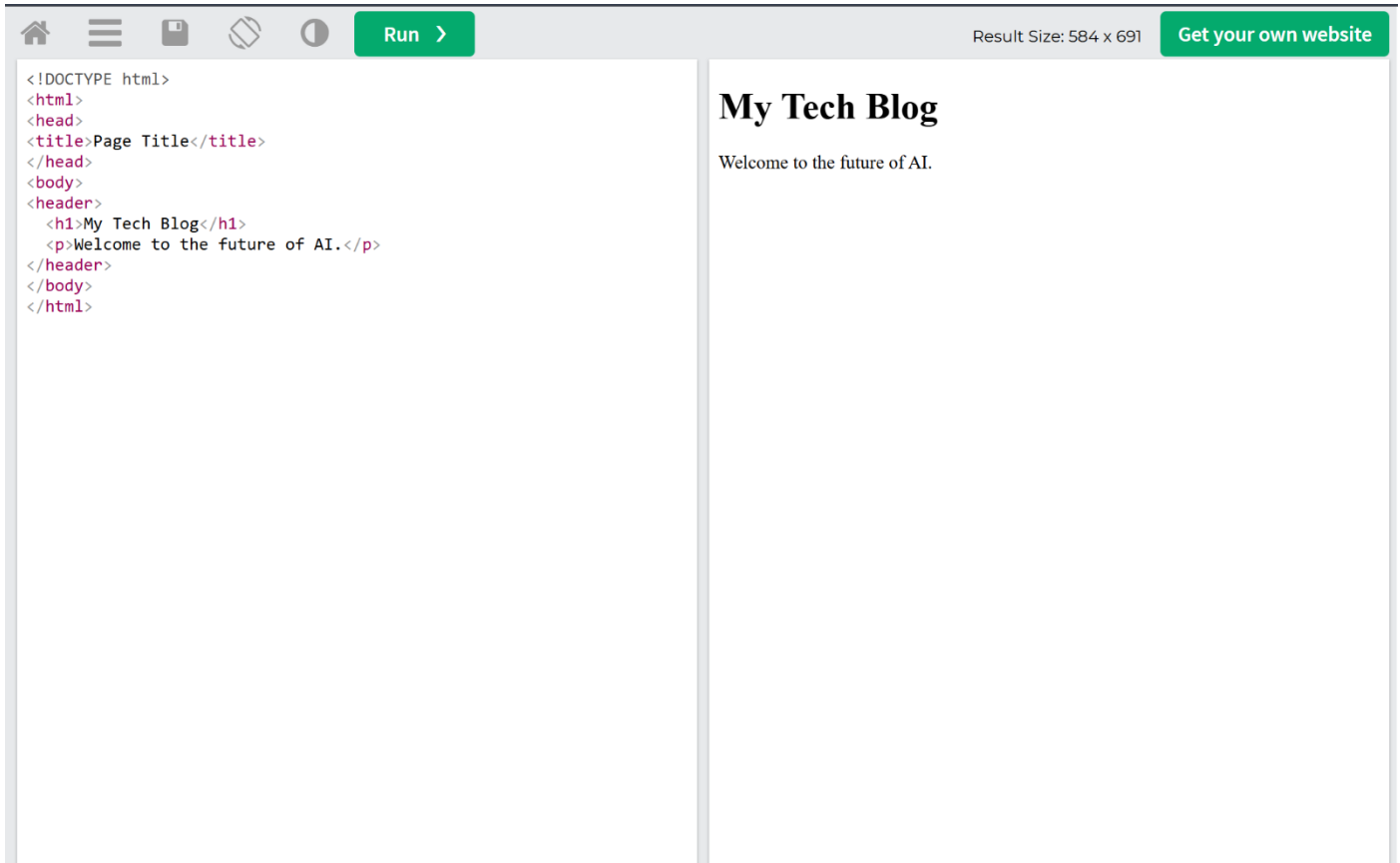
- **Accessibility:** Screen readers use semantic tags (`<nav>`, `<h1>`) as signposts to help visually impaired users navigate the page effectively.
- **SEO:** Search engines better understand the content and context of a page structured with semantic HTML, which can improve search rankings.
- **Maintainability:** Using meaningful tags instead of generic `<div>` with class names (e.g., `<div class="header">`) makes the code easier to read, debug, and update for other developers



1. The Header (<header>)

This is the introductory content of a page or a section. It often contains the logo, search bar, or main title.

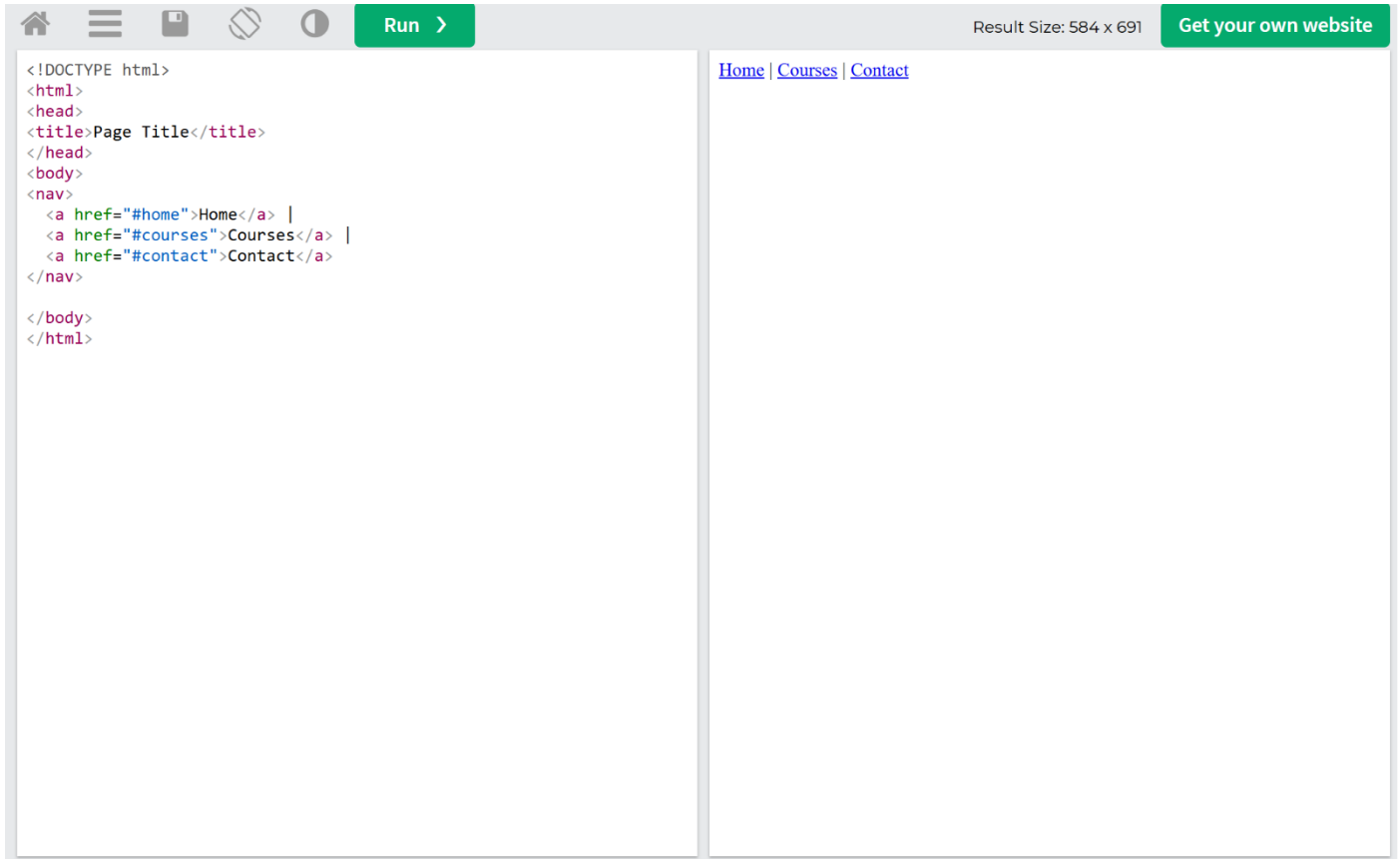
- **Tag:** <header>
- **Note:** You can have multiple headers (e.g., a main site header and a header for a specific article).
- **Global Attributes:** (id, class, style, etc.).



2. The Navigation (<nav>)

A container for major blocks of navigation links.

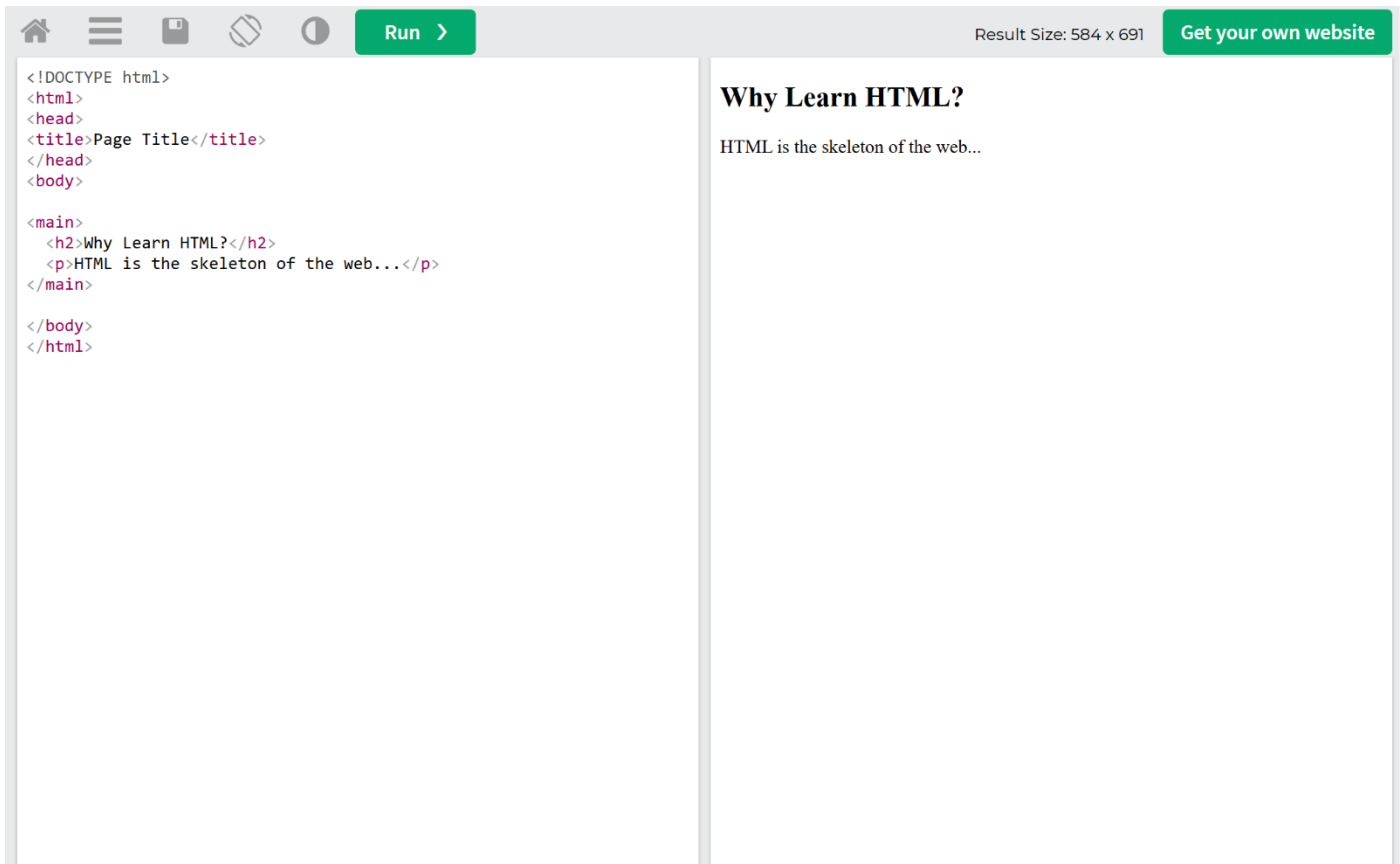
- **Tag:** `<nav>`
- **Usage:** Don't put every link inside `<nav>`, only the major ones.



3. The Main Content (<main>)

The dominant content of the <body>. There can be only **one** visible <main> element per page.

- **Tag:** <main>



The screenshot shows a web development tool interface. On the left, there is a code editor with the following HTML code:

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<main>
  <h2>Why Learn HTML?</h2>
  <p>HTML is the skeleton of the web...</p>
</main>

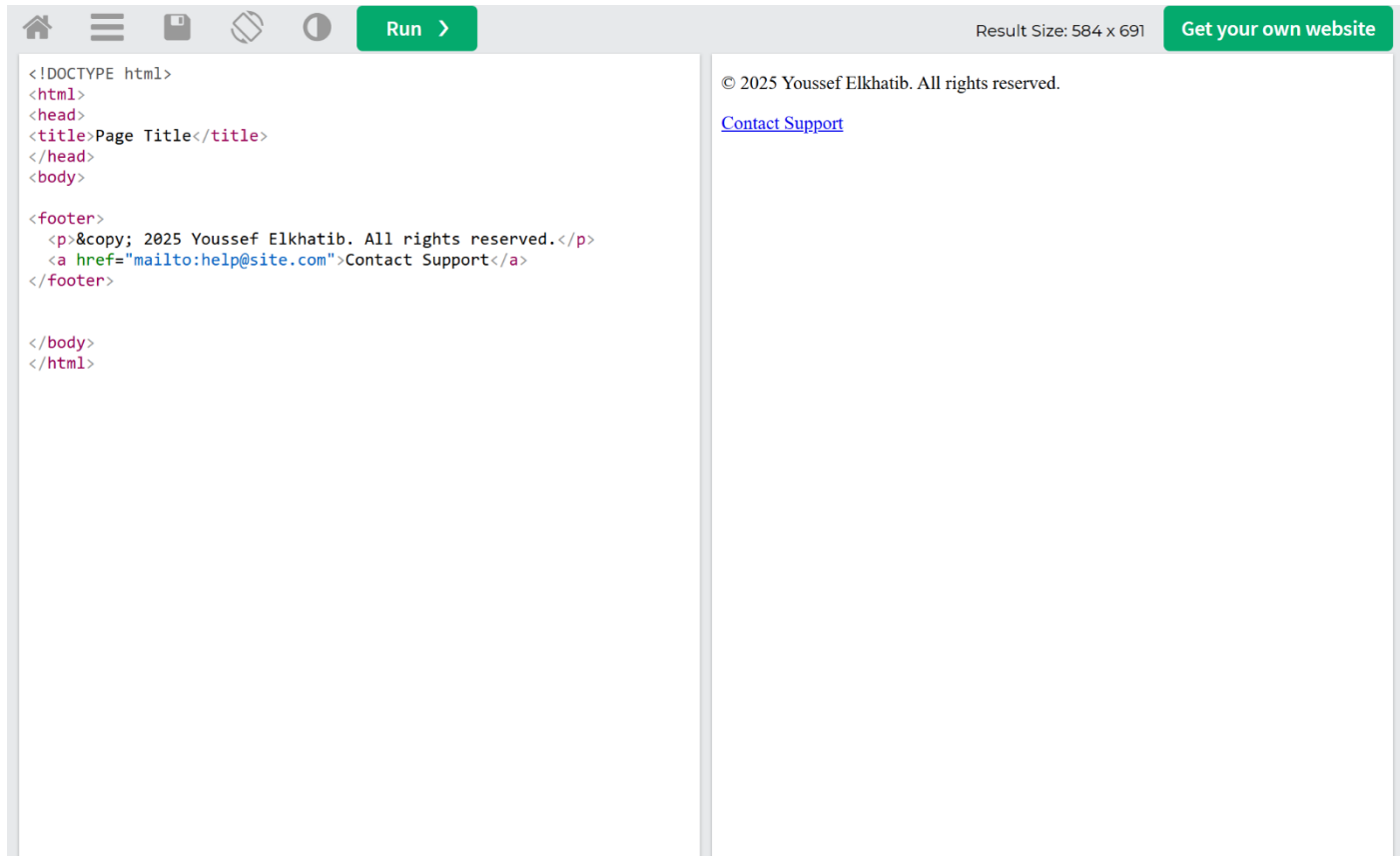
</body>
</html>
```

On the right, there is a preview window showing the rendered output. The preview displays the text "Why Learn HTML?" in a large, bold font, followed by the text "HTML is the skeleton of the web...". The preview window also shows a "Result Size: 584 x 691" and a "Get your own website" button.

4. The Footer (<footer>)

The bottom section of a page or article. It typically contains copyright info, contact details, or related links.

- **Tag:** <footer>



The screenshot shows a web editor interface. On the left, there is a code editor with the following HTML code:

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<footer>
  <p>&copy; 2025 Youssef Elkhatib. All rights reserved.</p>
  <a href="mailto:help@site.com">Contact Support</a>
</footer>

</body>
</html>
```

On the right, there is a preview window showing the rendered output:

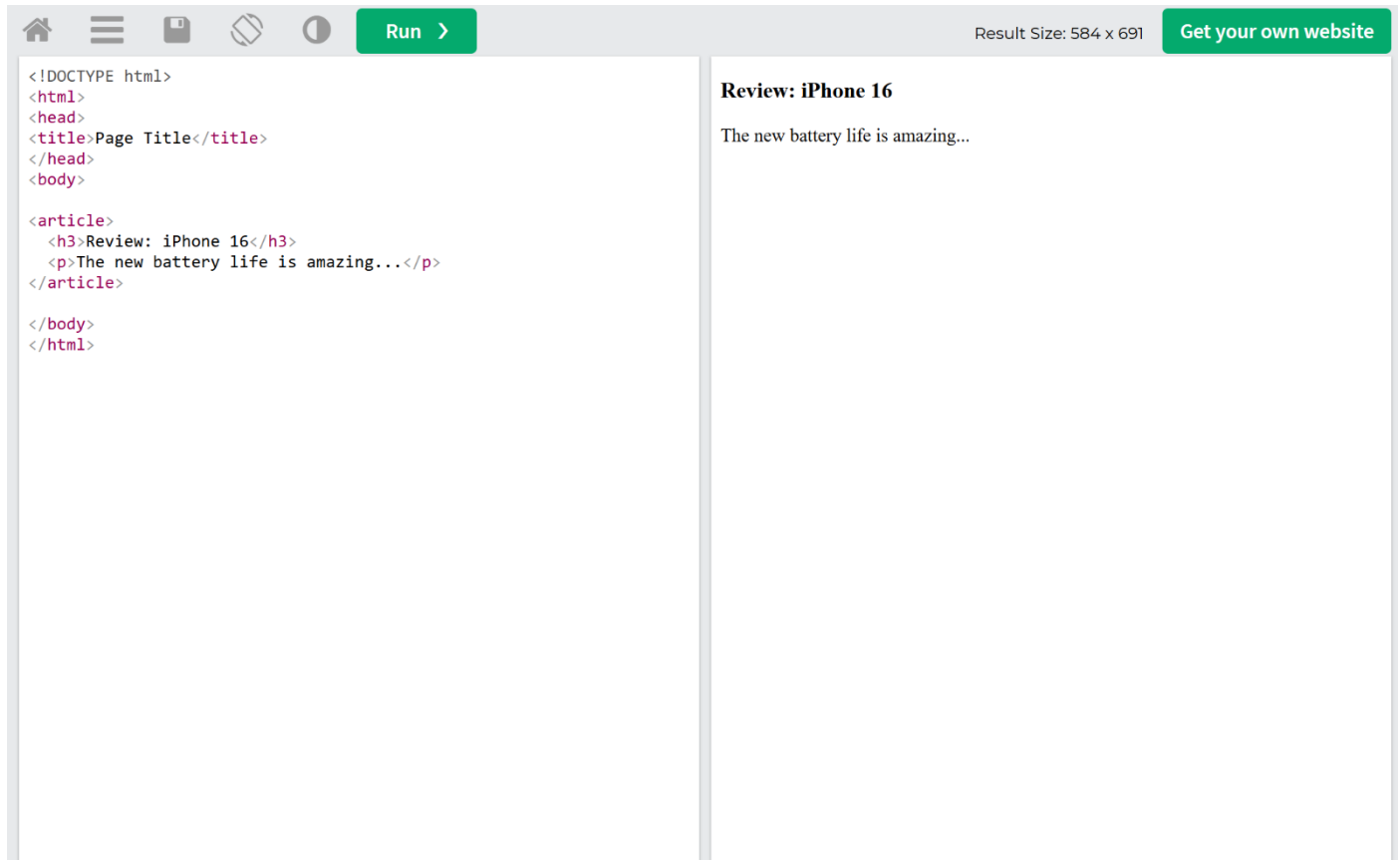
© 2025 Youssef Elkhatib. All rights reserved.
[Contact Support](mailto:help@site.com)

The editor interface includes a top bar with a 'Run' button and a 'Get your own website' button. The result size is indicated as 584 x 691.

5. The Article (<article>)

A self-contained piece of content that makes sense **on its own**. If you removed it from the site and put it elsewhere, it would still make perfect sense.

- **Tag:** <article>
- **Use Cases:** Blog posts, news stories, product cards, forum comments.



The screenshot shows a web browser interface. The top bar includes navigation icons (home, menu, save, refresh, stop) and a green 'Run >' button. The right side of the top bar shows 'Result Size: 584 x 691' and a green button that says 'Get your own website'.

The main area is split into two panels. The left panel displays the following HTML code:

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<article>
  <h3>Review: iPhone 16</h3>
  <p>The new battery life is amazing...</p>
</article>

</body>
</html>
```

The right panel shows the rendered output of this code. It features a bold heading 'Review: iPhone 16' followed by a paragraph of text: 'The new battery life is amazing...'.

6. The Section (<section>)

A thematic grouping of content, usually with a heading. It is part of a larger whole.

- **Tag:** <section>
- **Use Cases:** "Contact Us" section, "Features" section, "Testimonials" section.

Element	Independence	Example
<article>	High. Independent & Reusable.	A Weather Report.
<section>	Medium. Connected to context.	The "Sports" section of a newspaper.
<div>	None. Just a generic box.	A wrapper to turn text blue.

7. The Sidebar (<aside>)

Content that is tangentially related to the main content.

- **Tag:** <aside>
- **Use Cases:** Sidebars, advertising boxes, "Related Posts".

The screenshot shows a web browser interface. On the left, there is a code editor with the following HTML code:

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>
<aside>
  <h4>Did you know?</h4>
  <p>HTML was invented in 1993.</p>
</aside>
</body>
</html>
```

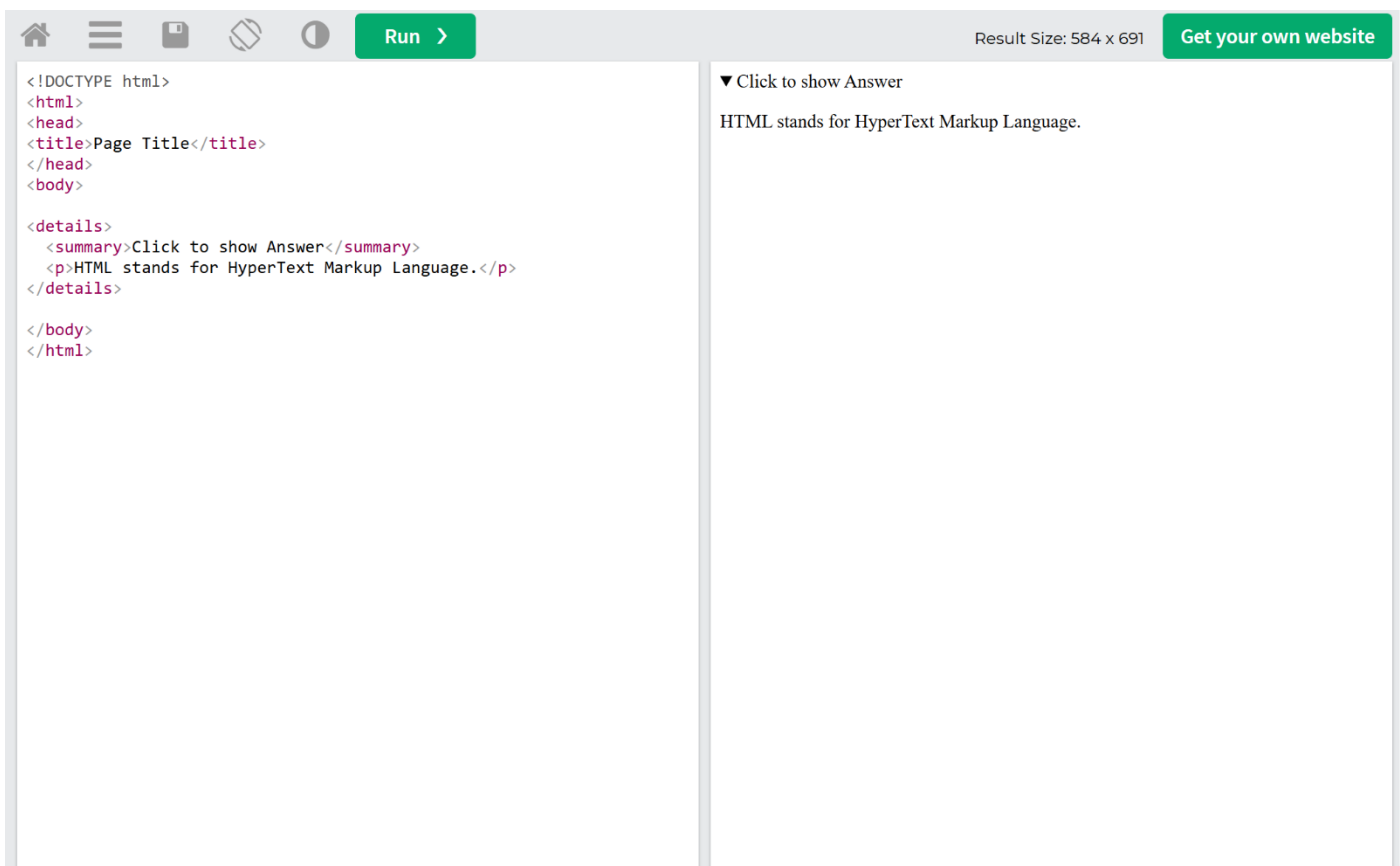
On the right, the rendered output is shown. It features a sidebar with the heading "Did you know?" and the text "HTML was invented in 1993." The browser's address bar shows "Result Size: 584 x 691" and a button that says "Get your own website".

8. The Interactive Toggle (<details> & <summary>)

Creates a collapsible widget (accordion) without JavaScript.

- **Wrapper:** <details>
- **Title:** <summary> (The visible text you click).

Attribute	Placed In	Description
open	<details>	If present, the widget is open by default.



The screenshot shows a web browser interface. On the left, there is a code editor with the following HTML code:

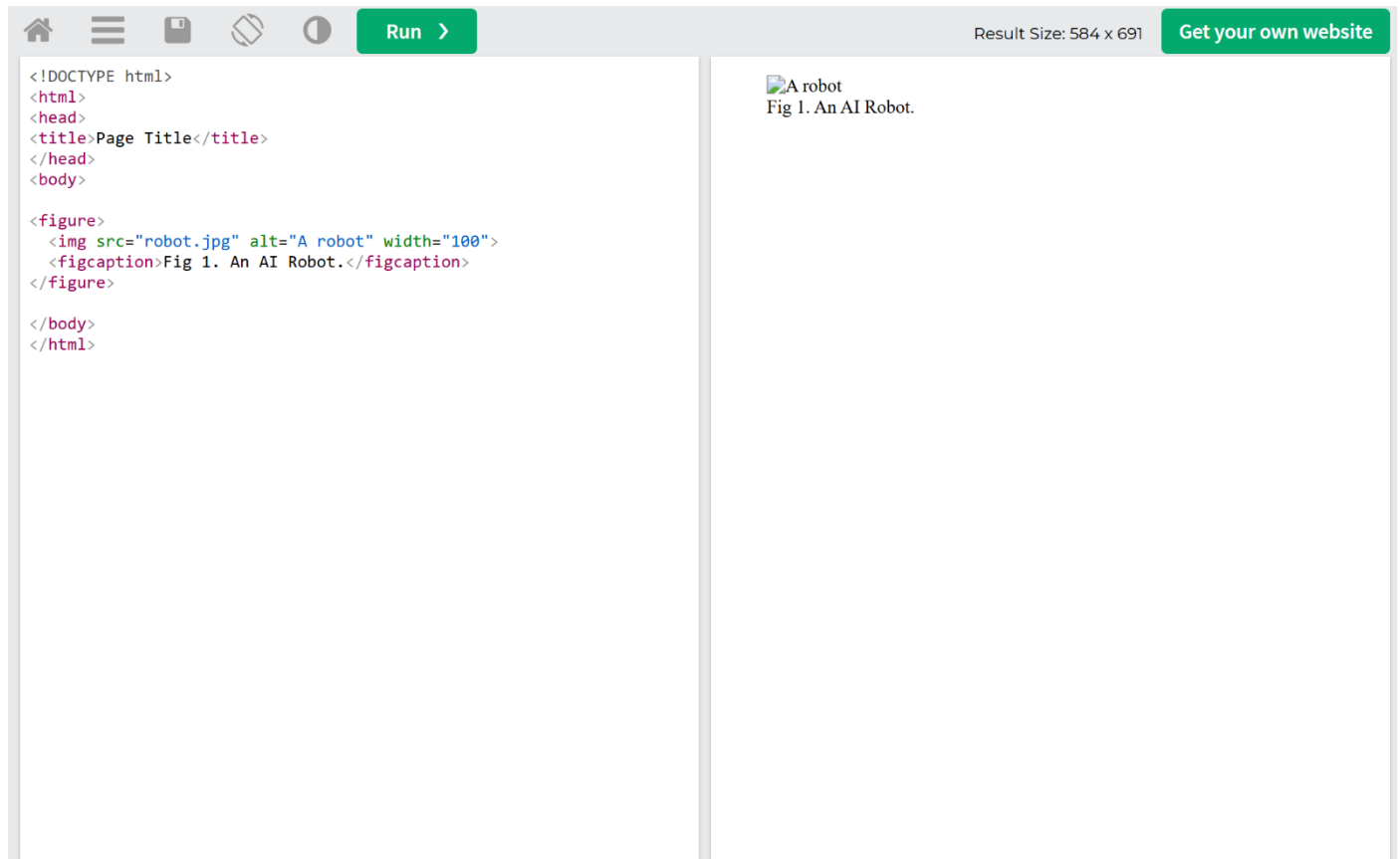
```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>
<details>
  <summary>Click to show Answer</summary>
  <p>HTML stands for HyperText Markup Language.</p>
</details>
</body>
</html>
```

On the right, the rendered output is shown. It features a green button with a downward arrow and the text "Click to show Answer". Below the button, the text "HTML stands for HyperText Markup Language." is displayed. The browser's top bar includes navigation icons, a "Run" button, and a "Get your own website" button. The result size is indicated as 584 x 691.

9. The Image Container (<figure> & <figcaption>)

Used to group an image with its caption.

- **Wrapper:** <figure>
- **Caption:** <figcaption>



10. The Highlighter (<mark>)

Highlights text for reference or emphasis.

- **Tag:** `<mark>`

11. The Time (<time>)

Defines a specific time (or date). This helps search engines add events to calendars.

- **Tag:** `<time>`

Attribute	Description	Important Values
datetime	Machine-readable format.	YYYY-MM-DD (e.g., 2025-12-25)

The screenshot shows a web browser interface. On the left, the HTML source code is displayed with syntax highlighting. On the right, the rendered page is shown. The code includes a title 'Page Title', a paragraph with a highlighted word 'milk', and another paragraph with a time '9:00 AM'.

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<p>Do not forget to buy <mark>milk</mark> today.</p>
<p>We open at <time datetime="09:00">9:00 AM</time>.</p>

</body>
</html>
```

Do not forget to buy **milk** today.
We open at 9:00 AM.

12. The Ultimate Semantic Layout Code

Here is a complete blog page structure using every tag above.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Semantic HTML Blog</title>
</head>
<body>

  <header style="background: #f4f4f4; padding: 20px;">
    <h1>The Daily Tech</h1>
    <p>News for Developers</p>

    <nav>
      <a href="#home">Home</a> |
      <a href="#news">News</a> |
      <a href="#about">About</a>
    </nav>
  </header>

  <div style="display: flex;">

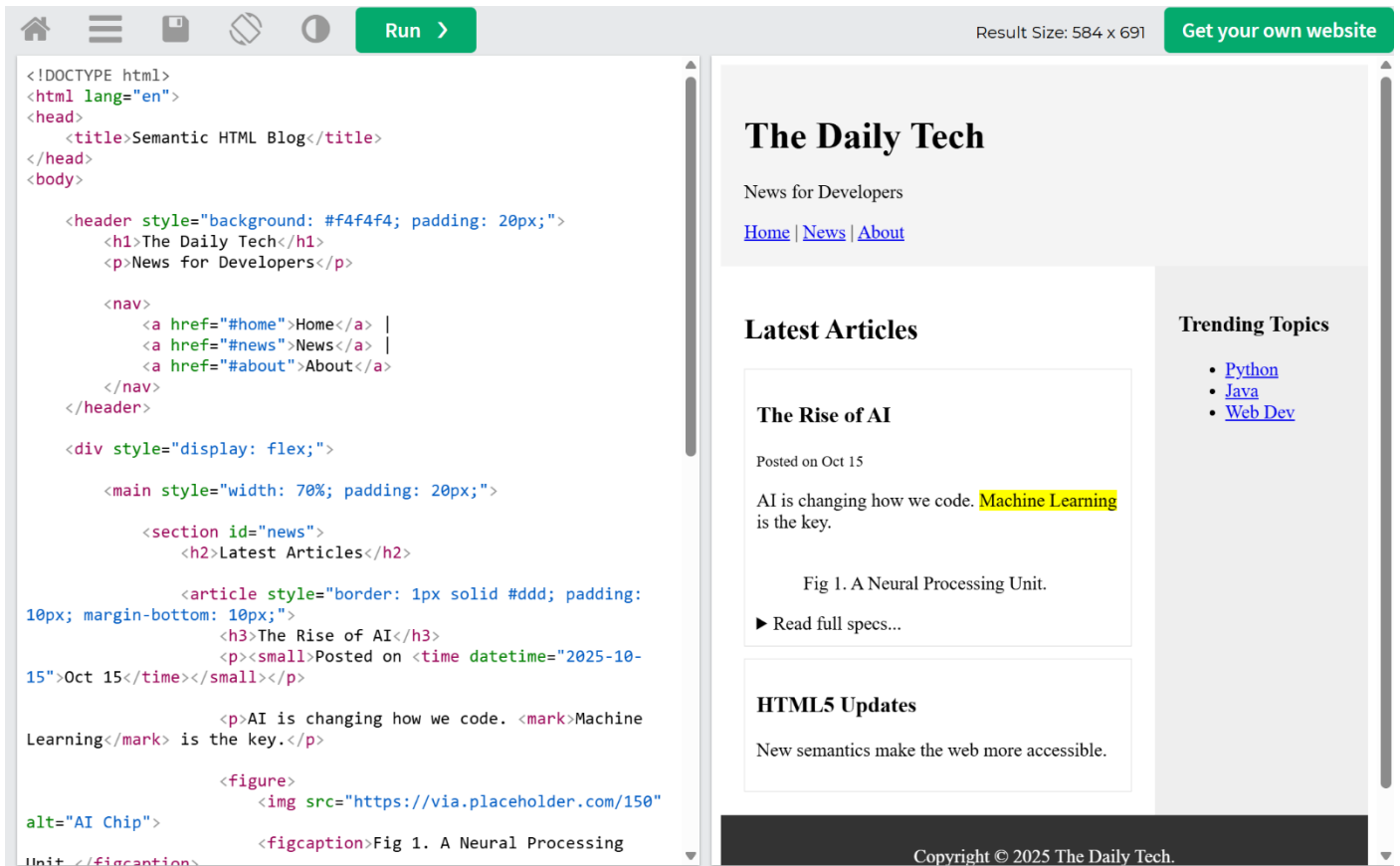
    <main style="width: 70%; padding: 20px;">

      <section id="news">
        <h2>Latest Articles</h2>

        <article style="border: 1px solid #ddd; padding:
10px; margin-bottom: 10px;">
          <h3>The Rise of AI</h3>
          <p><small>Posted on <time datetime="2025-10-
15">Oct 15</time></small></p>

          <p>AI is changing how we code. <mark>Machine
Learning</mark> is the key.</p>

          <figure>
            
            <figcaption>Fig 1. A Neural Processing
Unit</figcaption>
```



```
Unit.</figcaption>
  </figure>

  <details>
    <summary>Read full specs...</summary>
    <p>This chip has 50 Teraflops of power.</p>
  </details>
</article>

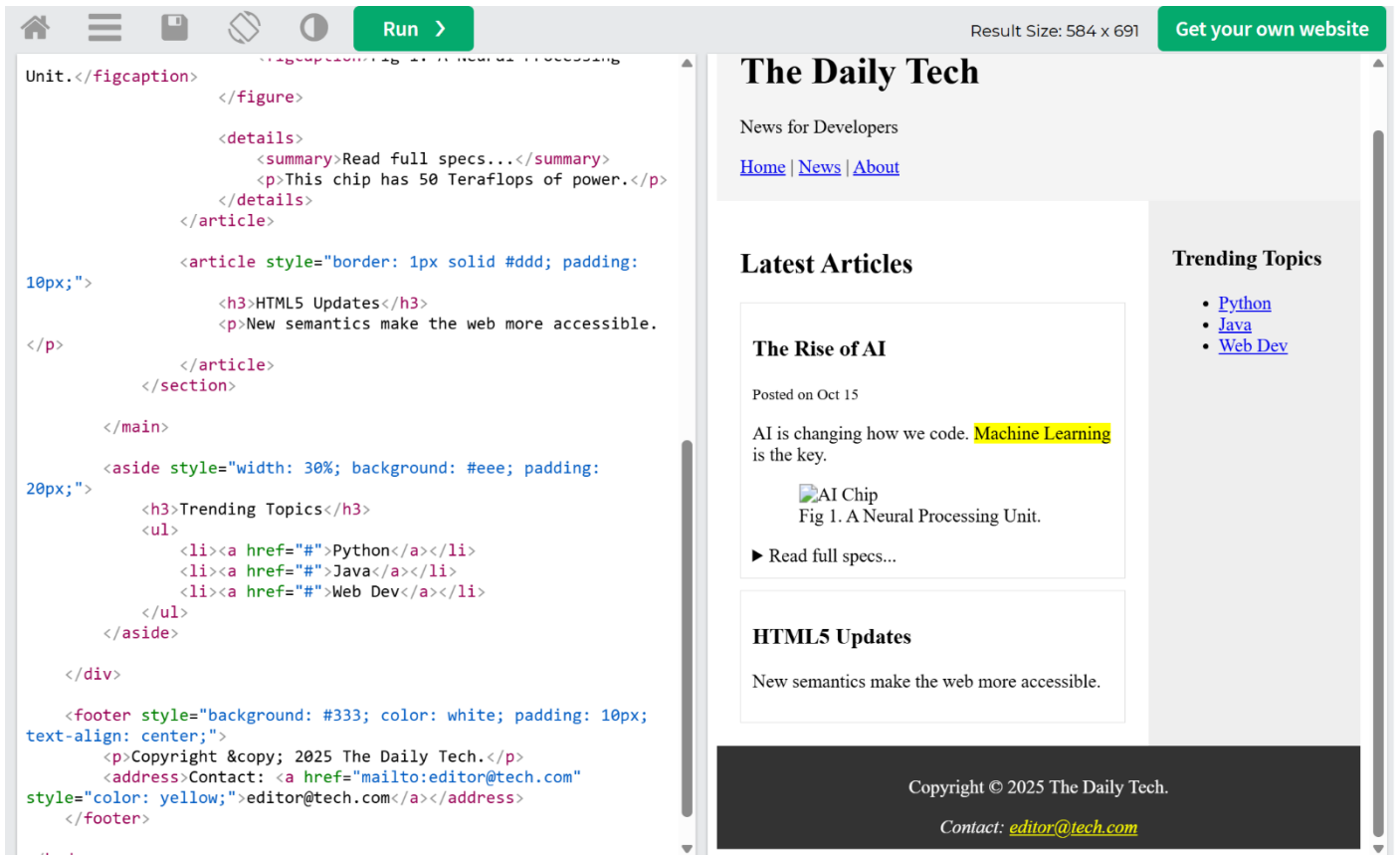
  <article style="border: 1px solid #ddd; padding:
10px;">
    <h3>HTML5 Updates</h3>
    <p>New semantics make the web more accessible.
</p>
  </article>
</section>

</main>

<aside style="width: 30%; background: #eee; padding:
20px;">
  <h3>Trending Topics</h3>
  <ul>
    <li><a href="#">Python</a></li>
    <li><a href="#">Java</a></li>
    <li><a href="#">Web Dev</a></li>
  </ul>
</aside>

</div>

<footer style="background: #333; color: white; padding: 10px;
text-align: center;">
  <p>Copyright &copy; 2025 The Daily Tech.</p>
  <address>Contact: <a href="mailto:editor@tech.com"
style="color: yellow;">editor@tech.com</a></address>
</footer>
</body>
```



Conclusion

In conclusion, this document has successfully met its objective of designing a structured, web-based digital document that transforms static educational content into a dynamic and accessible platform. By effectively utilizing HTML5 elements—including semantic text formatting, hyperlinked navigation, multimedia integration, and organized tables—the document solves the persistent issue of data fragmentation found in traditional note-taking methods. The result is a centralized, navigable architecture that not only streamlines the retrieval of information but also serves as a robust foundation for future frontend development, demonstrating the power of the web's core structural language in enhancing the user experience.

Feedback & Contribution

This is a living document. If you have feedback, suggestions, or additional insights that could improve it, I welcome your input. Sharing ideas helps keep this material accurate, relevant, and valuable for everyone.

Copyright & Usage

© 2025 [Youssef Amgad Elkhatib].

This document is intended **for learning and reference purposes only**.

You are free to read, use, and share the content for personal or educational use, but:

- **Do not copy or republish this document as your own.**
- Always provide proper credit when referencing.
- Commercial use or redistribution without permission is not allowed.

You may not reproduce this document in whole or in part without attribution. Suggestions and feedback are welcome, and contributors will be acknowledged, but copyright remains with the author.

Acknowledgments

This document is authored and copyrighted by **[Youssef Amgad Elkhatib]**.

Special thanks to the contributors who provided valuable feedback and suggestions for improvements:

Name	Contribution