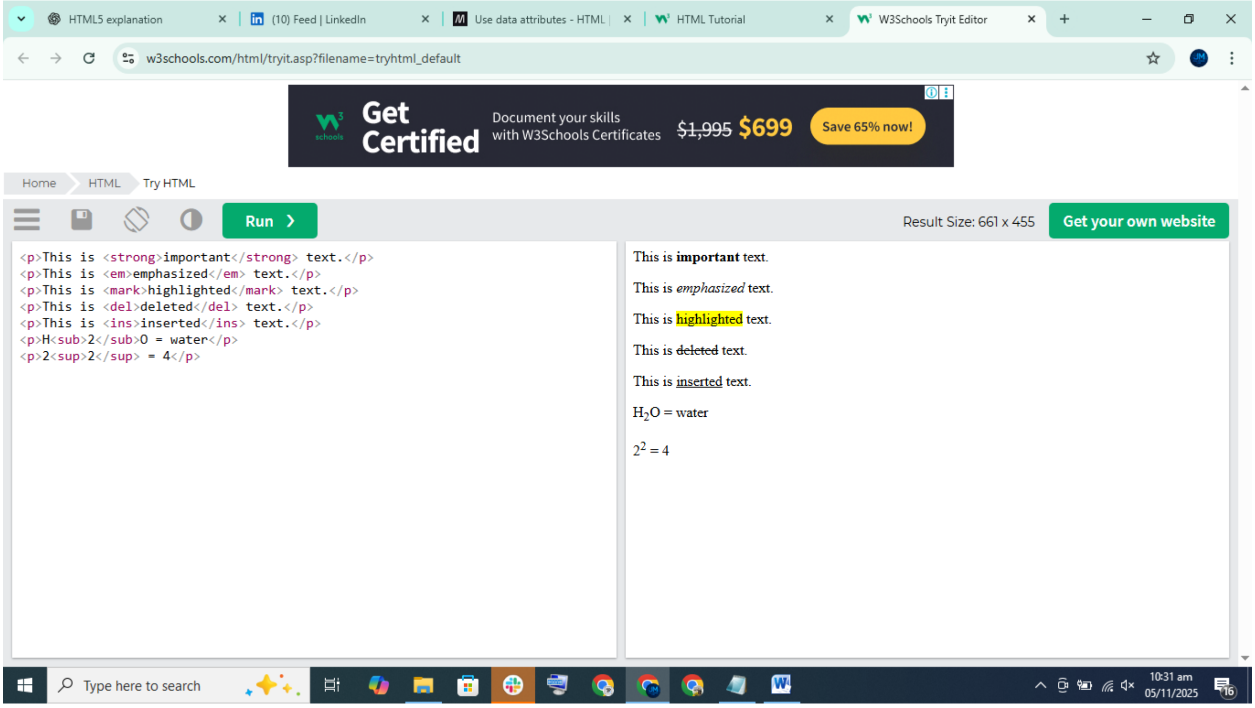
Full stack notes by jhangir:

HTML:

Basic tags: v13

Tag	Meaning	Looks
`<strong>`	Strong importance	Bold
`<em>`	Emphasized	Italic
`<mark>`	Highlight	Yellow highlight
`<del>`	Deleted text	Strikethrough
`<ins>`	Inserted	Underlined
`<sub>`	Subscript	H_2O ($_2$ goes down)
`<sup>`	Superscript	x^2 (^{2} goes up)



```
<p>This is <strong>important</strong> text.</p>
<p>This is <em>emphasized</em> text.</p>
<p>This is <mark>highlighted</mark> text.</p>
<p>This is <del>deleted</del> text.</p>
<p>This is <ins>inserted</ins> text.</p>
<p>H<sub>2</sub>O = water</p>
<p>2<sup>2</sup> = 4</p>
```

This is **important** text.

This is *emphasized* text.

This is highlighted text.

This is ~~deleted~~ text.

This is inserted text.

H_2O = water

$2^2 = 4$

⍰ Shortcuts Learned

Shortcut	Function
`! + Enter`	Create full HTML template
`Ctrl/Cmd + Enter`	New line from anywhere
`Ctrl/Cmd + /`	Add comment
`Shift + Alt + Down`	Duplicate line
`Ctrl/Cmd + +`	Zoom in browser

What Are DevTools?

DevTools are built-in tools in browsers (like Chrome) that help developers inspect and edit web pages. They help you see **HTML, CSS, and behavior** of a website.

You can open DevTools by:

- **Right-click → Inspect**
- Press **F12**
- **Ctrl + Shift + I** (Windows)
- **Cmd + Option + I** (Mac)

Ordered List Example

```
<h3>My Morning Routine</h3>
<ol>
  <li>Wake up</li>
  <li>Brush teeth</li>
  <li>Drink water</li>
  <li>Exercise</li>
</ol>
```

Output:
My Morning Routine

1. Wake up
2. Brush teeth
3. Drink water
4. Exercise

□ Unordered List Example

```
<h3>Shopping List</h3>
<ul>
  <li>Bread</li>
  <li>Milk</li>
  <li>Eggs</li>
  <li>Butter</li>
</ul>
```

Output:
Shopping List
• Bread
• Milk
• Eggs
• Butter

□ Description List Example

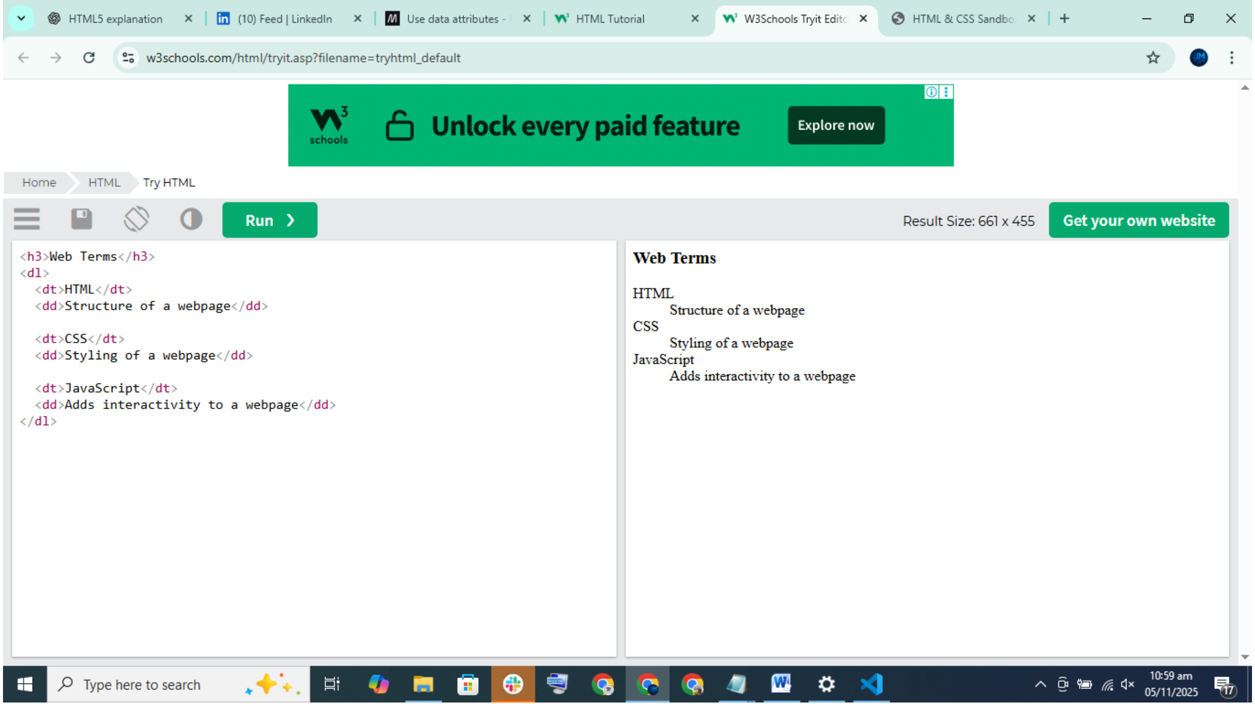
```
<h3>Web Terms</h3>
<dl>
  <dt>HTML</dt>
  <dd>Structure of a webpage</dd>

  <dt>CSS</dt>
  <dd>Styling of a webpage</dd>

  <dt>JavaScript</dt>
  <dd>Adds interactivity to a webpage</dd>
</dl>
```

Output:
Web Terms
HTML – Structure of a webpage
CSS – Styling of a webpage
JavaScript – Adds interactivity to a webpage

□ Tip

Use:

Tag	Purpose
`<ol>`	Ordered numbers
`<ul>`	Bullet points
`<dl>`	Definitions



```
<h3>Web Terms</h3>
<dl>
  <dt>HTML</dt>
  <dd>Structure of a webpage</dd>

  <dt>CSS</dt>
  <dd>Styling of a webpage</dd>

  <dt>JavaScript</dt>
  <dd>Adds interactivity to a webpage</dd>
</dl>
```

```
Web Terms

HTML
        Structure of a webpage
CSS
        Styling of a webpage
JavaScript
        Adds interactivity to a webpage
```

What are Anchor Tags?

Anchor tags (`<a>`) are used in HTML to create **links**. These links allow users to:

- Open other websites (external links)
- Move to other pages on the same website (relative/internal pages)
- Jump to a section on the same page
- Open email applications to send mail
- Download or open files like PDFs

□ Key Types of Links Explained

1□ External Link

Used to open other websites.

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

□ Opens Google in same tab

To open in a **new tab**:

```
<a href="https://google.com" target="_blank">Visit Google</a>
```

2□ Relative/Internal Page Link

Link to another page inside the same website folder.

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

3□ Internal Section Link (Jump to Section)

Move to a section on the same page.

```
<a href="#services">Go to Services</a>

<p id="services">This is services section.</p>
```

4□ Email Link

Open the default email app to send email.

```
<a href="mailto:example@example.com">Send Email</a>
```

5□ File Download Link

Link to open/download a file like PDF.

```
<a href="invoice.pdf">Download Invoice</a>
```

□ Title Attribute

Shows tooltip when hovering.

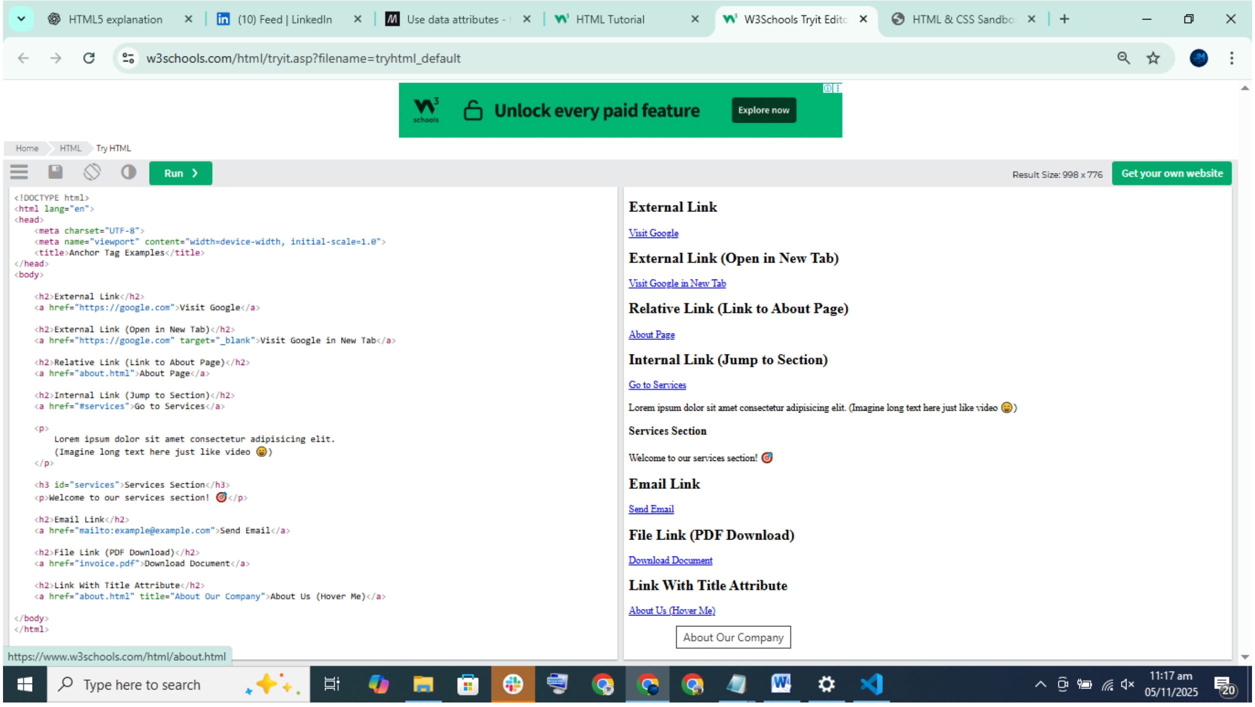
```
<a href="about.html" title="About Our Company">About Page</a>
```

□ Extra Notes

- `href` = link location
- `target="_blank"` opens link in new tab
- Internal navigation uses `#sectionID`
- Use anchor tags for menus, buttons, and navigation links

□ Real-World Example Navigation Menu

```
<nav>
  <a href="index.html">Home</a>
  <a href="about.html">About</a>
  <a href="services.html">Services</a>
  <a href="contact.html">Contact</a>
</nav>
```



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Anchor Tag Examples</title>
</head>
<body>

    <h2>External Link</h2>
    <a href="https://google.com">Visit Google</a>

    <h2>External Link (Open in New Tab)</h2>
    <a href="https://google.com" target="_blank">Visit Google in New Tab</a>

    <h2>Relative Link (Link to About Page)</h2>
    <a href="about.html">About Page</a>

    <h2>Internal Link (Jump to Section)</h2>
    <a href="#services">Go to Services</a>

    <p>
        Lorem ipsum dolor sit amet consectetur adipisicing elit.
        (Imagine long text here just like video 😀)
    </p>

    <h3 id="services">Services Section</h3>
    <p>Welcome to our services section! 🎯</p>

    <h2>Email Link</h2>
    <a href="mailto:example@example.com">Send Email</a>

    <h2>File Link (PDF Download)</h2>
    <a href="invoice.pdf">Download Document</a>

    <h2>Link With Title Attribute</h2>
    <a href="about.html" title="About Our Company">About Us (Hover Me)</a>

</body>
</html>
```

External Link

Visit Google

External Link (Open in New Tab)

Visit Google in New Tab

Relative Link (Link to About Page)

About Page

Internal Link (Jump to Section)

Go to Services

Lorem ipsum dolor sit amet consectetur adipisicing elit. (Imagine long text here just like video 😀)

Services Section

Welcome to our services section! 🎯

Email Link

Send Email

File Link (PDF Download)

Download Document

Link With Title Attribute

About Us (Hover Me)

About Our Company

□ HTML Image Tag (`<img>`) Explained

In HTML, we use the **`<img>` tag** to show images on a webpage.
This tag is **self-closing** (means it doesn't need a closing tag).

□ Basic Syntax

```
<img src="image.jpg" alt="My Image">
```

□ Important Attributes

Attribute	Meaning
`src`	The **path/location** of your image
`alt`	Text shown if image can’t load (also used for accessibility)
`width` & `height`	Set image size (but usually done in CSS)
`title`	Shows text when you hover over image

□ Example: Display Local Image

Folder structure:

```
project/
 └── images/
      └── landscape.jpg
```

Code:

```
<img src="images/landscape.jpg" alt="Landscape">
```

□ Broken Image Example (Alt Text Shows)

```
<img src="images/notfound.jpg" alt="Landscape">
```

If file doesn't exist → browser will show the **alt** text.

□ Absolute Image URL Example

```
<img src="https://picsum.photos/200/300" alt="Random Image">
```

This loads a random image from the web.

□ Setting Width & Height (not recommended in HTML)

```
<img src="images/landscape.jpg" width="400" height="300" alt="Landscape">
```

Better to use **CSS** later for styling.

□ Add Title Tooltip

```
<img src="images/landscape.jpg" alt="Landscape" title="Beautiful Landscape">
```

Hover → shows tooltip text

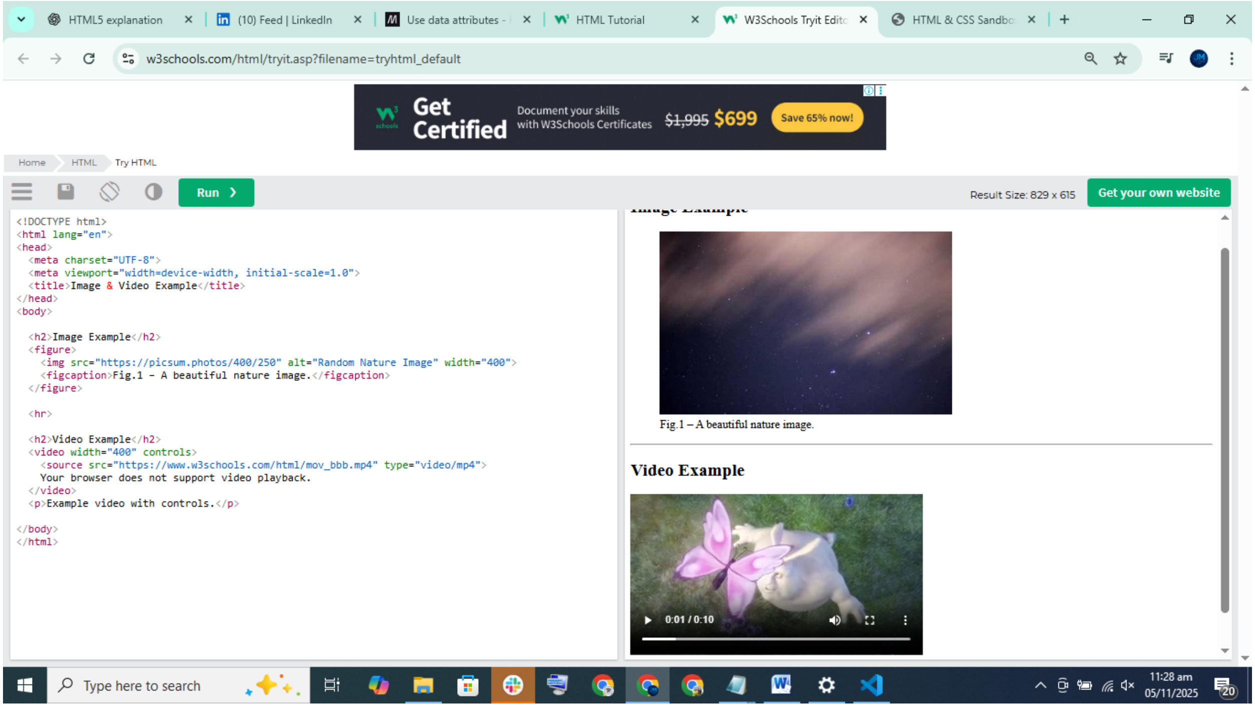
□ `figure` & `figcaption`

Used for images **with descriptions/captions**

```
<figure>
  <img src="images/landscape.jpg" width="400" height="300" alt="Landscape">
  <figcaption>Fig.1 – Beautiful Landscape</figcaption>
</figure>
```

□ Key Points to Remember

- `<img>` = shows an image
- Must include `src` & `alt`
- Broken image → alt text appears
- Use CSS for styling images
- `figure` + `figcaption` = image with caption



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta viewport="width=device-width, initial-scale=1.0">
  <title>Image & Video Example</title>
</head>
<body>

  <h2>Image Example</h2>
  <figure>
    <img src="https://picsum.photos/400/250" alt="Random Nature Image" width="400">
    <figcaption>Fig.1 - A beautiful nature image.</figcaption>
  </figure>

  <hr>

  <h2>Video Example</h2>
  <video width="400" controls>
    <source src="https://www.w3schools.com/html/mov_bbb.mp4" type="video/mp4">
    Your browser does not support video playback.
  </video>
  <p>Example video with controls.</p>

</body>
</html>
```

Fig.1 – A beautiful nature image.

Video Example

□ Block-Level vs Inline Elements (Easy Explanation)

Block-Level Elements

✔ Always start on a **new line**
✔ Take up the **full width** available
✔ Push the next element to the **next line**
✔ Allow **top & bottom margin**

Examples:
`<div>`, `<p>`, `<h1> - <h6>`, `<ul>`, `<ol>`, `<li>`, `<section>`, `<article>`, `<header>`, `<footer>`, `<form>`, `<table>`

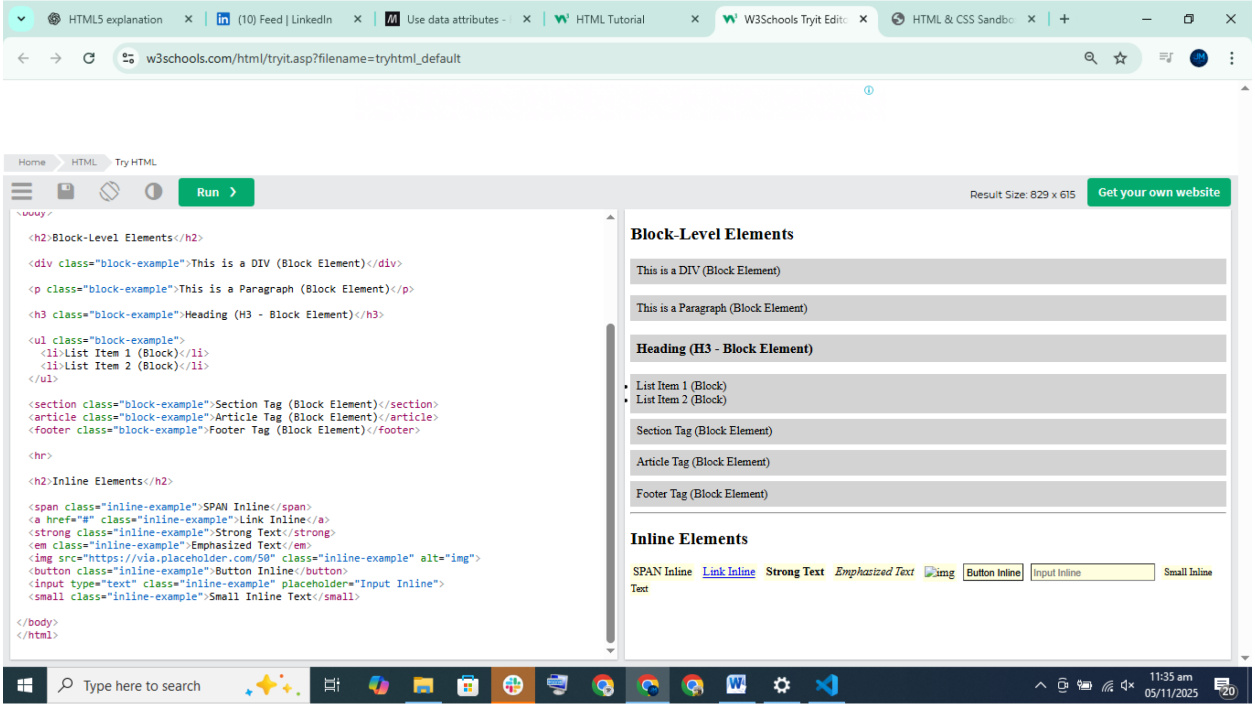
□ Think of block elements like **big blocks** that take the full row.

Inline Elements

✔ Do **not** start a new line
✔ Only take the **needed width**
✔ Sit **side-by-side**
✔ Can only have **left/right margin**, not top/bottom

Examples:
`<span>`, `<a>`, `<img>`, `<strong>`, `<em>`, `<button>`, `<input>`, `<label>`, `<small>`

□ Think of inline elements like **text characters** — they appear in the same line.



```html
<body>

  <h2>Block-Level Elements</h2>

  <div class="block-example">This is a DIV (Block Element)</div>

  <p class="block-example">This is a Paragraph (Block Element)</p>

  <h3 class="block-example">Heading (H3 - Block Element)</h3>

  <ul class="block-example">
    <li>List Item 1 (Block)</li>
    <li>List Item 2 (Block)</li>
  </ul>

  <section class="block-example">Section Tag (Block Element)</section>
  <article class="block-example">Article Tag (Block Element)</article>
  <footer class="block-example">Footer Tag (Block Element)</footer>

  <hr>

  <h2>Inline Elements</h2>

  <span class="inline-example">SPAN Inline</span>
  <a href="#" class="inline-example">Link Inline</a>
  <strong class="inline-example">Strong Text</strong>
  <em class="inline-example">Emphasized Text</em>
  <img src="https://via.placeholder.com/50" class="inline-example" alt="img">
  <button class="inline-example">Button Inline</button>
  <input type="text" class="inline-example" placeholder="Input Inline">
  <small class="inline-example">Small Inline Text</small>

</body>
</html>
```

Block-Level Elements

This is a DIV (Block Element)

This is a Paragraph (Block Element)

Heading (H3 - Block Element)

- List Item 1 (Block)
- List Item 2 (Block)

Section Tag (Block Element)

Article Tag (Block Element)

Footer Tag (Block Element)

Inline Elements

SPAN Inline Link Inline **Strong Text** *Emphasized Text* img Button Inline Input Inline Small Inline Text

1️⃣ Line Break (`<br>`)

- Used to move text to a new line.
- Example:

```
Hello<br>World
```

- **Tip:** Only use `<br>` for text line break — *not for spacing* (use CSS padding/margin for space).

2️⃣ Horizontal Line (`<hr>`)

- Creates a horizontal divider line between content.

```
<p>First paragraph</p>
<hr>
<p>Second paragraph</p>
```

- **Tip:** Don’t use `<hr>` for layout, use CSS instead.

3️⃣ `<pre>` Tag (Preformatted Text)

- Shows text **exactly** how you type it, including spaces, tabs, and new lines.

```
<pre>
Hello       World
    Indented text
</pre>
```

4️⃣ `<code>` Tag (Code Snippets)

- Used to display code in HTML.

```
<code>console.log("Hello")</code>
```

5️⃣ HTML Entities

Used to show special characters in browser.

Character Entity

< `&lt;`

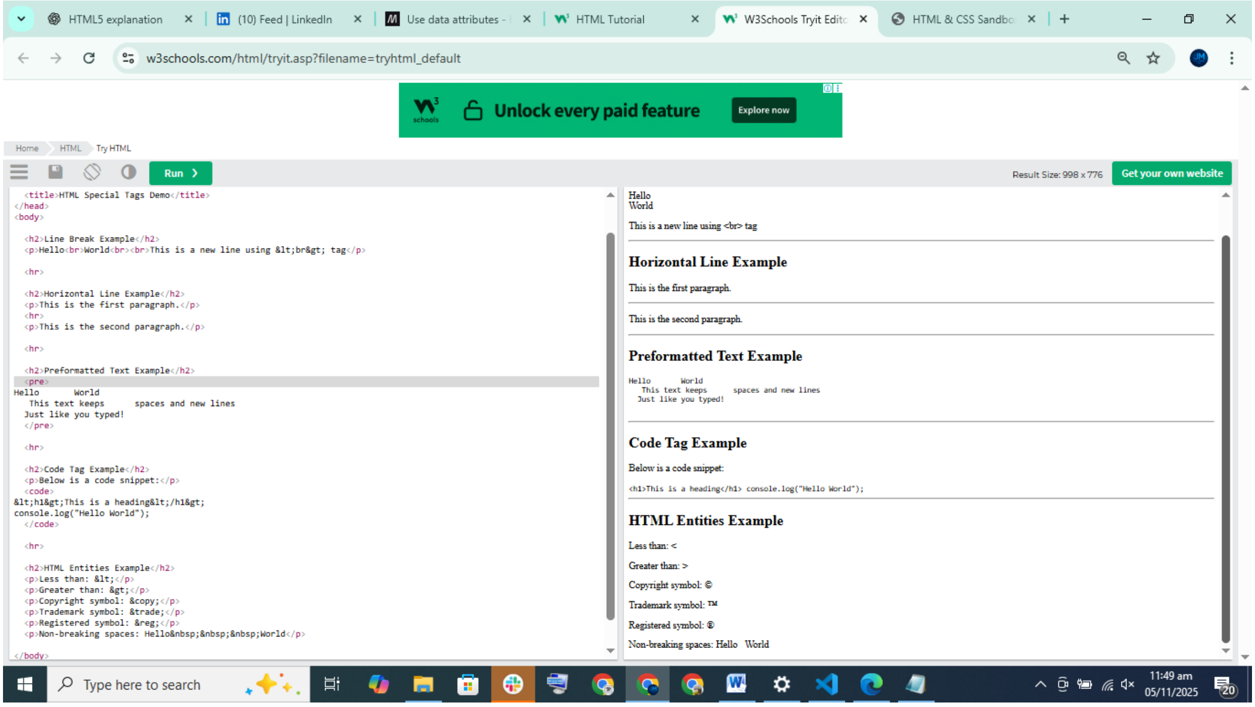
| `&gt;`
Copyright © | `&copy;`
Trademark ™ | `&trade;`
Non-breaking space | `&nbsp;`

Example — showing `<h1>` as text:

```
&lt;h1&gt;This is a heading&lt;/h1&gt;
```

□ Main Points to Remember

- `<br>` → new line
- `<hr>` → horizontal line
- `<pre>` → preserves text formatting
- `<code>` → shows code
- **Entities** allow you to display HTML characters like < > ©



```
  <title>HTML Special Tags Demo</title>
</head>
<body>

  <h2>Line Break Example</h2>
  <p>Hello<br>World<br><br>This is a new line using &lt;br&gt; tag</p>

  <hr>

  <h2>Horizontal Line Example</h2>
  <p>This is the first paragraph.</p>
  <hr>
  <p>This is the second paragraph.</p>

  <hr>

  <h2>Preformatted Text Example</h2>
  <pre>
Hello       World
   This text keeps      spaces and new lines
  Just like you typed!
  </pre>

  <hr>

  <h2>Code Tag Example</h2>
  <p>Below is a code snippet:</p>
  <code>
&lt;h1&gt;This is a heading&lt;/h1&gt;
console.log("Hello World");
  </code>

  <hr>

  <h2>HTML Entities Example</h2>
  <p>Less than: &lt;</p>
  <p>Greater than: &gt;</p>
  <p>Copyright symbol: &copy;</p>
  <p>Trademark symbol: &trade;</p>
  <p>Registered symbol: &reg;</p>
  <p>Non-breaking spaces: Hello&nbsp;&nbsp;&nbsp;World</p>

</body>
```

Hello
World

This is a new line using <br> tag

Horizontal Line Example

This is the first paragraph.

This is the second paragraph.

Preformatted Text Example

```
Hello       World
   This text keeps      spaces and new lines
  Just like you typed!
```

Code Tag Example

Below is a code snippet:

`<h1>This is a heading</h1> console.log("Hello World");`

HTML Entities Example

Less than: <

Greater than: >

Copyright symbol: ©

Trademark symbol: ™

Registered symbol: ®

Non-breaking spaces: Hello   World

Video 20

What You Learned in This Video

□ DIV (Block-level container)

- `<div>` is used to group big blocks of content.
- It always starts on a **new line**.
- Used to create sections like header, footer, blocks, cards, sections on websites.
- Most websites use **divs** everywhere to structure pages.

Example use:

- Wrap header
- Wrap navigation menu
- Wrap sections and paragraphs
- Wrap footer

□ SPAN (Inline container)

- `<span>` is for **small text styling inside a line**.
- It does **not break the line**.
- Used to style only part of text (color, bold, highlight)

□ Simple Rule

Tag	Use for
`<div>`	Big sections / Layout blocks
`<span>`	Styling small words inside a sentence

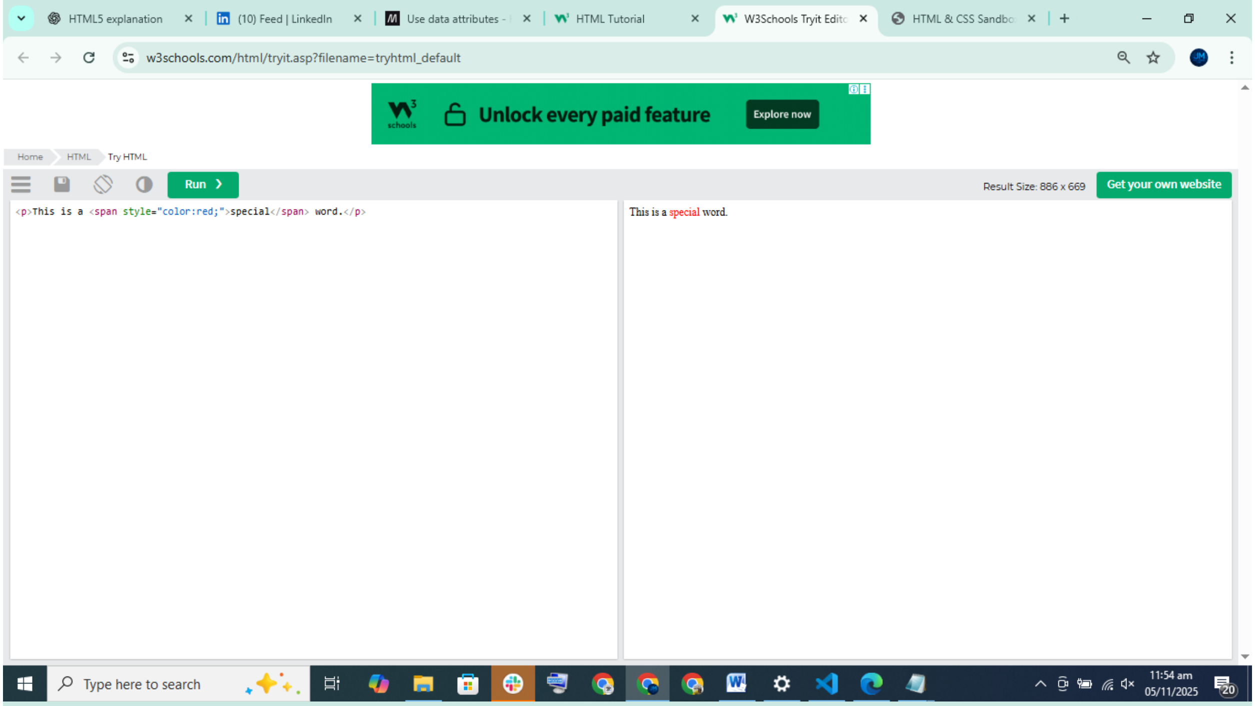
□ Example Websites

They showed how websites like Netlify wrap boxes, cards, sections using **DIV**.

□ Why we use them

To organize HTML and later apply:

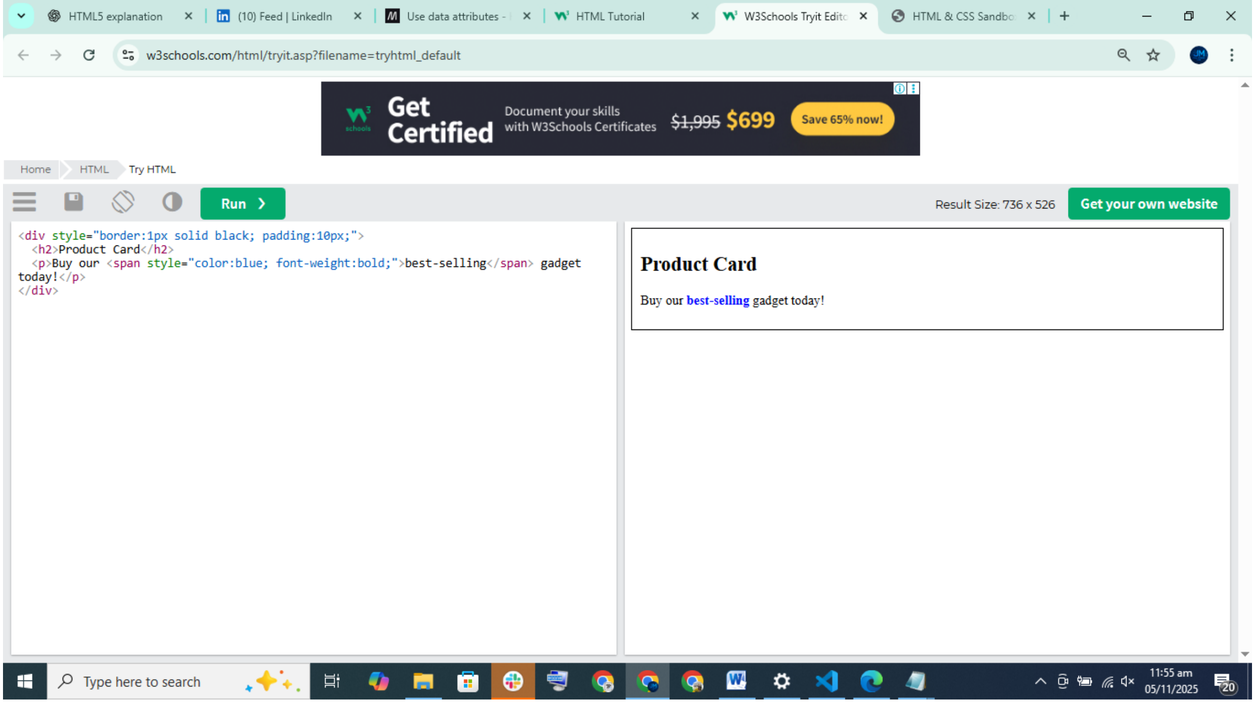
- CSS
- Layout styles
- Colors, spacing, borders



w3schools.com/html/tryit.asp?filename=tryhtml_default
Unlock every paid feature
Explore now
Home
HTML
Try HTML
Run
Result Size: 886 x 669
Get your own website

```
<p>This is a <span style="color:red;">special</span> word.</p>
```

This is a special word.



w3schools.com/html/tryit.asp?filename=tryhtml_default
Get Certified
Document your skills with W3Schools Certificates
$1,995 $699
Save 65% now!
Home
HTML
Try HTML
Run
Result Size: 736 x 526
Get your own website

```
<div style="border:1px solid black; padding:10px;">
  <h2>Product Card</h2>
  <p>Buy our <span style="color:blue; font-weight:bold;">best-selling</span> gadget
today!</p>
</div>
```

Product Card

Buy our **best-selling** gadget today!

Memory Trick

Think like this

DIV = Room in a house □

SPAN = Word highlighted in a book ✏□

□ Pro Tip

In modern development we also use:

- `<section>`
- `<header>`
- `<footer>`

- `<article>`

But **DIV is still used everywhere**, especially in CSS frameworks (Bootstrap, Tailwind, etc.).

Classes & IDs help us identify HTML elements

HTML elements can have:

Attribute	Purpose
`class`	Used to style multiple elements (repeatable)
`id`	Used for **unique** element on the page (one-time)

□ Class (`class=""`)

✔ Used when you want to style or target **many elements**
✔ Reusable — you can apply the same class to multiple elements
✔ Mostly used for CSS styling

Example:

```
<div class="card text-center">Content Box</div>
<div class="card text-center">Another Box</div>
```

Here `card` and `text-center` can be reused many times.

□ ID (`id=""`)

✔ Used when you want to target **ONE specific element**
✔ Should be **unique** (only one per page)
✔ Mostly used for JavaScript or page anchor links

Example:

```
<div id="header">Website Header</div>
```

This ID will be used **only once**.

□ Developer Best Practice (from video)

For	Use
CSS	Classes □
JavaScript	IDs □ or Classes □ but ID preferred
One-time element	ID □

For	Use
Repeat elements	Class □

□ Real Examples (like instructor showed)

□ Classes for reusable styles

```
<div class="card text-center"></div>
<div class="card text-center"></div>
```

□ ID for unique sections

Used for scrolling to a section (like About page link)

```
<a href="#about">About</a>

<section id="about">About section</section>
```

□ Simple Memory Trick

Feature	CLASS	ID
Use-case	Repeatable	Unique
Count	Many	Only One
Used for	CSS Styling	JavaScript / Links
Symbol in CSS	. class	# id

□ Example With Both

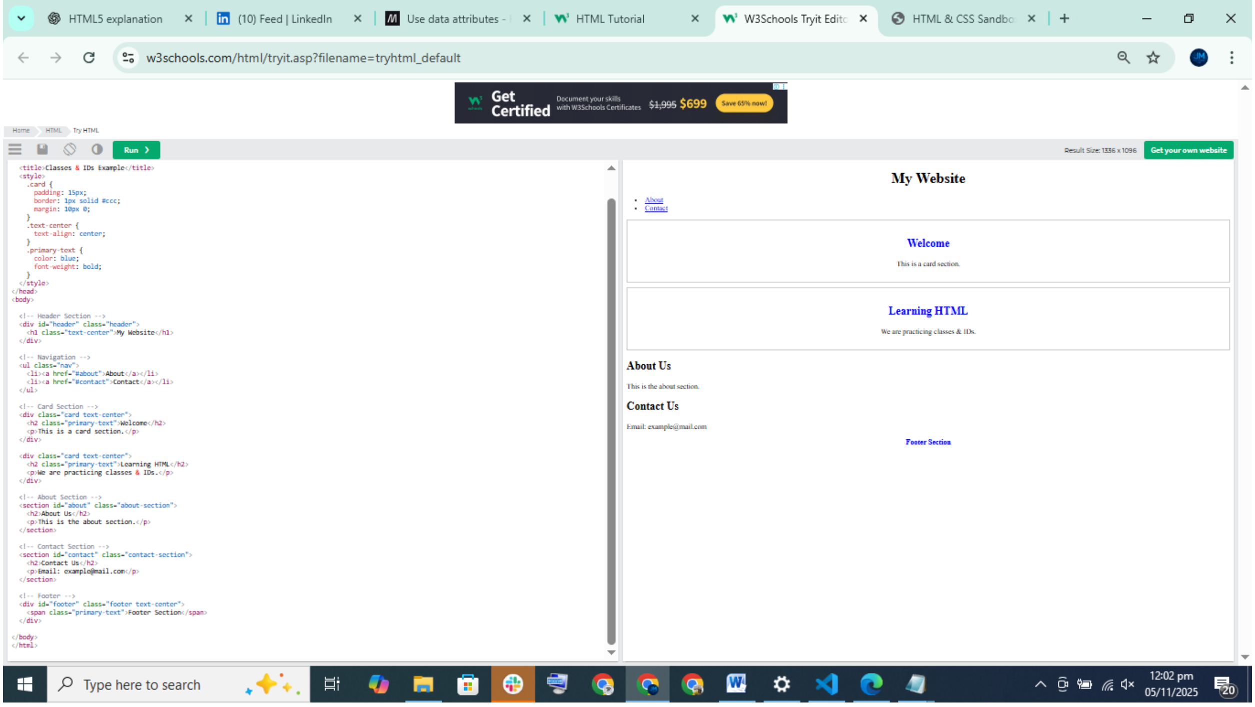
```
<div class="card" id="header">
  <h2 class="text-lg">Welcome</h2>
  <p class="primary-text">Hello User</p>
</div>
```

□ Extra Tip from video

- You **can** use class + id together on same element (but not always needed)
- You **can** bring elements to JavaScript with class or id
- **Semantic tags** (like `<header>`, `<footer>`, `<nav>`) coming next topic

□ Summary in Simple English

Classes = Many elements, reusable, for styling
ID = One element, unique, for JavaScript & page navigation



```
  <title>Classes & IDs Example</title>
  <style>
    .card {
      padding: 15px;
      border: 1px solid #ccc;
      margin: 10px 0;
    }
    .text-center {
      text-align: center;
    }
    .primary-text {
      color: blue;
      font-weight: bold;
    }
  </style>
</head>
<body>

  <!-- Header Section -->
  <div id="header" class="header">
    <h1 class="text-center">My Website</h1>
  </div>

  <!-- Navigation -->
  <ul class="nav">
    <li><a href="#about">About</a></li>
    <li><a href="#contact">Contact</a></li>
  </ul>

  <!-- Card Section -->
  <div class="card text-center">
    <h2 class="primary-text">Welcome</h2>
    <p>This is a card section.</p>
  </div>

  <div class="card text-center">
    <h2 class="primary-text">Learning HTML</h2>
    <p>We are practicing classes & IDs.</p>
  </div>

  <!-- About Section -->
  <section id="about" class="about-section">
    <h2>About Us</h2>
    <p>This is the about section.</p>
  </section>

  <!-- Contact Section -->
  <section id="contact" class="contact-section">
    <h2>Contact Us</h2>
    <p>Email: example@mail.com</p>
  </section>

  <!-- Footer -->
  <div id="footer" class="footer text-center">
    <span class="primary-text">Footer Section</span>
  </div>

</body>
</html>
```

My Website

- About
- Contact

Welcome

This is a card section.

Learning HTML

We are practicing classes & IDs.

About Us

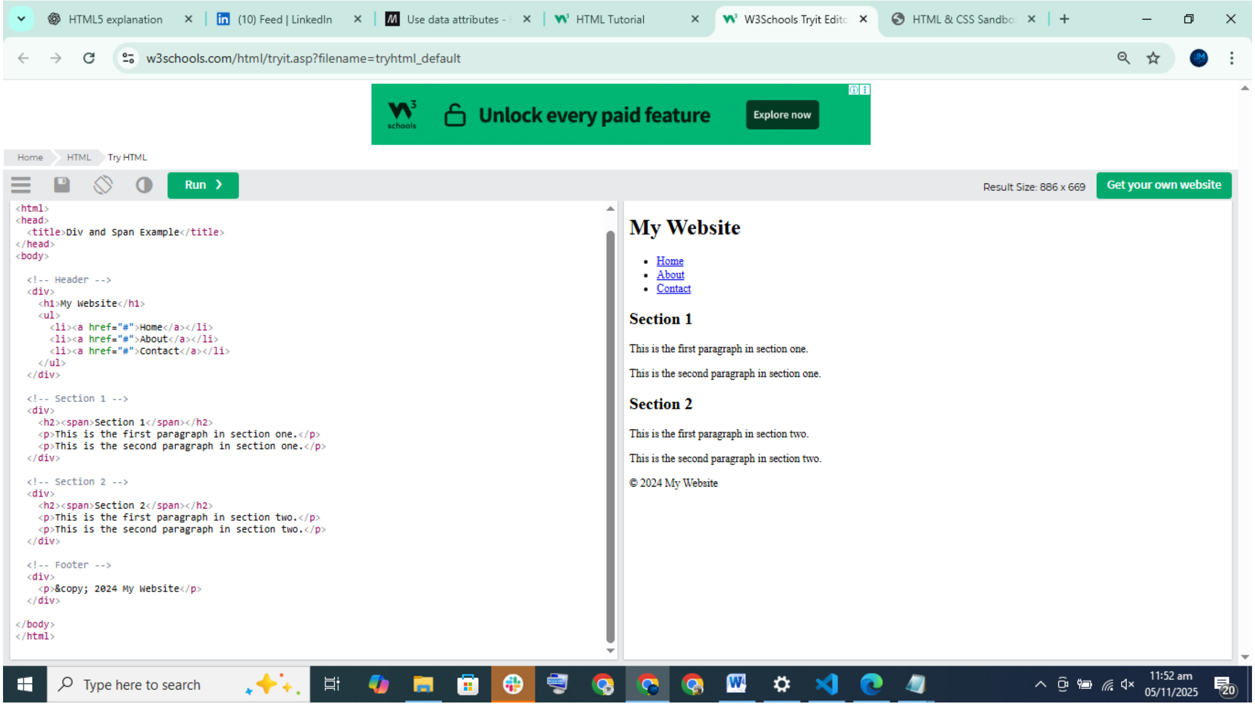
This is the about section.

Contact Us

Email: example@mail.com

Footer Section

Vid 21



```
<html>
<head>
  <title>Div and Span Example</title>
</head>
<body>

  <!-- Header -->
  <div>
    <h1>My Website</h1>
    <ul>
      <li><a href="#">Home</a></li>
      <li><a href="#">About</a></li>
      <li><a href="#">Contact</a></li>
    </ul>
  </div>

  <!-- Section 1 -->
  <div>
    <h2><span>Section 1</span></h2>
    <p>This is the first paragraph in section one.</p>
    <p>This is the second paragraph in section one.</p>
  </div>

  <!-- Section 2 -->
  <div>
    <h2><span>Section 2</span></h2>
    <p>This is the first paragraph in section two.</p>
    <p>This is the second paragraph in section two.</p>
  </div>

  <!-- Footer -->
  <div>
    <p>&copy; 2024 My Website</p>
  </div>

</body>
</html>
```

My Website

- Home
- About
- Contact

Section 1

This is the first paragraph in section one.

This is the second paragraph in section one.

Section 2

This is the first paragraph in section two.

This is the second paragraph in section two.

© 2024 My Website

Vid 22

⍰ What is Semantic HTML?

Semantic HTML means using HTML tags that clearly describe the meaning of content.

Instead of using `<div>` everywhere, HTML5 introduced special tags like:

Tag	Meaning
`<header>`	Top section / intro area
`<nav>`	Navigation / menu
`<main>`	Main content of page
`<section>`	Section of content
`<article>`	Independent content (like blog post)
`<aside>`	Sidebar content
`<footer>`	Bottom of page

□ Why Semantic HTML is Important?

✔ Makes code descriptive & readable
✔ Helps **search engines (SEO)** understand your page
✔ Improves **screen readers** for accessibility
✔ Better for structure and web standards

□ Example (Non-Semantic)

```
<div>
  <div>Menu</div>
  <div>Content</div>
  <div>Footer</div>
</div>
```

This doesn't tell us what each part really is.

□ Semantic Version

```
<header>Menu</header>
<main>Content</main>
<footer>Footer</footer>
```

Now it's clear □

□ Key Points From Your Text

- HTML5 brings semantic tags to replace just `<div>`
- `<div>` & `<span>` = general containers (no meaning)

- Semantic tags = meaningful labels
- You can still use classes with semantic tags for styling
- Using semantic tags = **good practice**

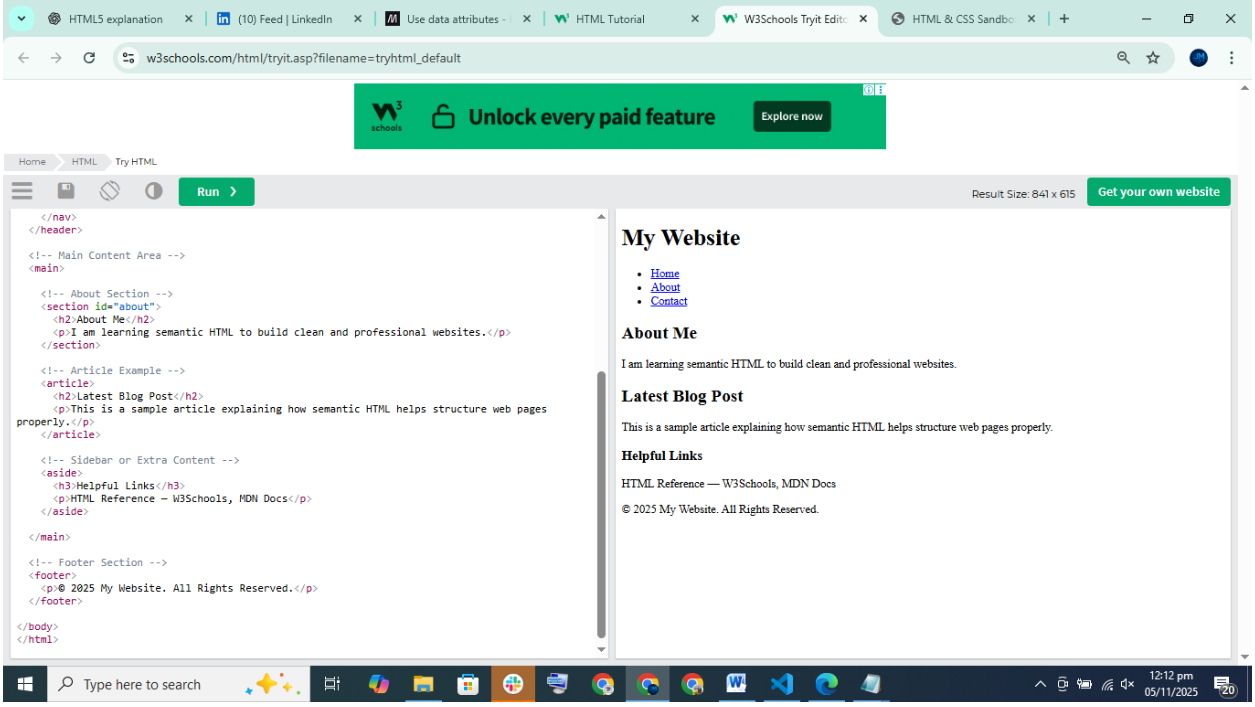
☐ Simple Real-Life Example

Bad:

```
<div id="top">My Website</div>
```

Better:

```
<header>My Website</header>
```



```
    </nav>
  </header>

  <!-- Main Content Area -->
  <main>

    <!-- About Section -->
    <section id="about">
      <h2>About Me</h2>
      <p>I am learning semantic HTML to build clean and professional websites.</p>
    </section>

    <!-- Article Example -->
    <article>
      <h2>Latest Blog Post</h2>
      <p>This is a sample article explaining how semantic HTML helps structure web pages
properly.</p>
    </article>

    <!-- Sidebar or Extra Content -->
    <aside>
      <h3>Helpful Links</h3>
      <p>HTML Reference — W3Schools, MDN Docs</p>
    </aside>

  </main>

  <!-- Footer Section -->
  <footer>
    <p>© 2025 My Website. All Rights Reserved.</p>
  </footer>

</body>
</html>
```

My Website

- Home
- About
- Contact

About Me

I am learning semantic HTML to build clean and professional websites.

Latest Blog Post

This is a sample article explaining how semantic HTML helps structure web pages properly.

Helpful Links

HTML Reference — W3Schools, MDN Docs

© 2025 My Website. All Rights Reserved.

Vid 23 □ What You Learned (Emmet Crash Course — Easy Explanation)

Emmet is a **tool in VS Code** that helps you write HTML & CSS **super fast** using shortcuts instead of typing full code.

Example:

- `p` + **Enter** → `<p></p>`
- `.title` + **Enter** → `<div class="title"></div>`
- `#main` + **Enter** → `<div id="main"></div>`
- `ul>li*5` + **Enter** → Creates a list with 5 `<li>` items

You can also:

- Add **multiple classes** → `div.red.big.center`
- Create **nested tags** → `div>p`
- Add **text inside tags** → `p{Hello World}`
- Auto numbering → `li.item$*4` → item1, item2, item3, item4
- `lorem` → dummy text generator (e.g., `lorem10` = 10 words)
- Add attributes quickly → `a:link`, `script:src`
- The `!` shortcut generates full HTML boilerplate — also from Emmet

---

□ Why Emmet is Useful

- Saves time typing HTML
- Helps build structure fast
- Great for practice & real projects
- Perfect for creating lists, forms, layouts quickly

□ Memory Shortcut Table

Shortcut	Output
`p`	`<p></p>`
`.box`	`<div class="box"></div>`
`#header`	`<div id="header"></div>`
`ul>li*3`	3 list items
`p{Hello}`	`<p>Hello</p>`
`lorem5`	5-word dummy text
`!`	Full HTML template

□ Summary (Easy Line)

"Emmet helps write HTML quickly using shortcuts like `.class`, `#id`, `ul>li*5`, and `p{Hello}` — saving time and improving workflow."

□ Memory Shortcut Table

Shortcut	Output
p	<p></p>
.box	<div class="box"></div>
#header	<div id="header"></div>
ul>li*3	3 list items
p{Hello}	<p>Hello</p>
lorem5	5-word dummy text
!	

Key board short cut for vscode vid 24

Here are the **most useful VS Code keyboard shortcuts** for daily coding ⍰

⍰ Basic Editing Shortcuts

Shortcut	Action
Ctrl + C	Copy line (if nothing selected)
Ctrl + X	Cut line (if nothing selected)
Ctrl + V	Paste
Ctrl + S	Save file
Ctrl + Z	Undo
Ctrl + Shift + Z	Redo
Ctrl + A	Select all
Ctrl + /	Toggle comment (single line comment)

⍰ Multi-cursor & Selection

Shortcut	Action
Alt + Click	Add cursor
Ctrl + Alt + ↑ / ↓	Add multiple cursors above/below
Ctrl + D	Select next match
Ctrl + Shift + L	Select all matches

Shortcut	Action
Shift + Alt + Right Arrow	Expand selection
Shift + Alt + Left Arrow	Shrink selection

▯ Moving & Editing Lines

Shortcut	Action
Alt + ↑ / ↓	Move line up/down
Shift + Alt + ↓ / ↑	Copy line up/down
Ctrl + Shift + K	Delete line
Ctrl + Enter	New line below
Ctrl + Shift + Enter	New line above

▯ File & Window Navigation

Shortcut	Action
Ctrl + P	Quick open files
Ctrl + B	Toggle sidebar
Ctrl + Shift + E	Show Explorer
Ctrl + Shift + F	Search files
Ctrl + Tab	Switch between open tabs
Ctrl + Shift + T	Reopen closed file tab

▯ Command & Settings

Shortcut	Action
Ctrl + Shift + P	Command palette
Ctrl + ,	Open Settings
Ctrl + K + Ctrl + S	Keyboard shortcut settings

⍰ Terminal Shortcuts

Shortcut	Action
Ctrl + `	Toggle terminal
Ctrl + Shift + `	New terminal
Ctrl + C	Stop running terminal process

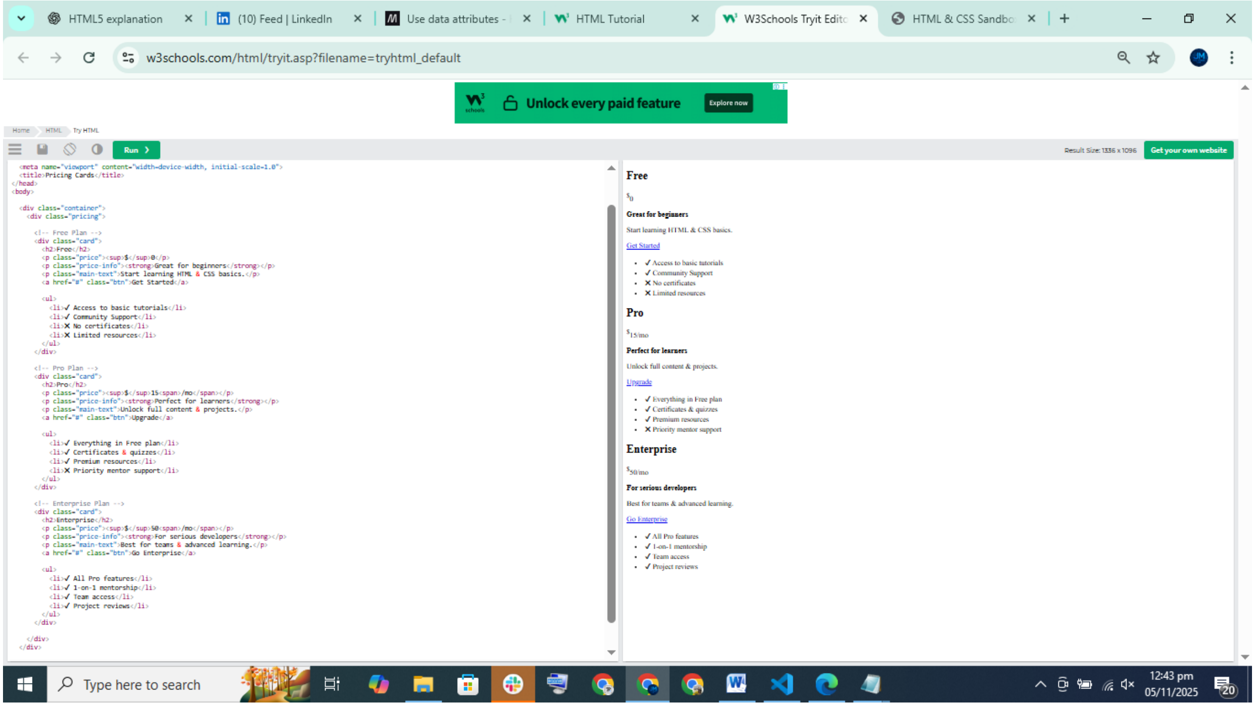
⍰ Zoom / View

Shortcut	Action
Ctrl + =	Zoom in
Ctrl + -	Zoom out
Ctrl + 0	Reset zoom
F11	Toggle full screen

⍰ Bonus: Emmet (HTML Shortcut Examples)

Type	Output
!	Full HTML boilerplate
div.container	`<div class="container"></div>`
ul>li*5	Create list with 5 items

Vid 25 project:



```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Pricing Cards</title>
</head>
<body>

  <div class="container">
    <div class="pricing">

      <!-- Free Plan -->
      <div class="card">
        <h2>Free</h2>
        <p class="price"><sup>$</sup>0</p>
        <p class="price-info"><strong>Great for beginners</strong></p>
        <p class="main-text">Start learning HTML & CSS basics.</p>
        <a href="#" class="btn">Get Started</a>

        <ul>
          <li>✓ Access to basic tutorials</li>
          <li>✓ Community Support</li>
          <li>✕ No certificates</li>
          <li>✕ Limited resources</li>
        </ul>
      </div>

      <!-- Pro Plan -->
      <div class="card">
        <h2>Pro</h2>
        <p class="price"><sup>$</sup>15<span>/mo</span></p>
        <p class="price-info"><strong>Perfect for learners</strong></p>
        <p class="main-text">Unlock full content & projects.</p>
        <a href="#" class="btn">Upgrade</a>

        <ul>
          <li>✓ Everything in Free plan</li>
          <li>✓ Certificates & quizzes</li>
          <li>✓ Premium resources</li>
          <li>✕ Priority mentor support</li>
        </ul>
      </div>

      <!-- Enterprise Plan -->
      <div class="card">
        <h2>Enterprise</h2>
        <p class="price"><sup>$</sup>50<span>/mo</span></p>
        <p class="price-info"><strong>For serious developers</strong></p>
        <p class="main-text">Best for teams & advanced learning.</p>
        <a href="#" class="btn">Go Enterprise</a>

        <ul>
          <li>✓ All Pro features</li>
          <li>✓ 1-on-1 mentorship</li>
          <li>✓ Team access</li>
          <li>✓ Project reviews</li>
        </ul>
      </div>

    </div>
  </div>
```

```
Free
$0
Great for beginners
Start learning HTML & CSS basics.
Get Started
  • ✓ Access to basic tutorials
  • ✓ Community Support
  • ✕ No certificates
  • ✕ Limited resources
Pro
$15/mo
Perfect for learners
Unlock full content & projects.
Upgrade
  • ✓ Everything in Free plan
  • ✓ Certificates & quizzes
  • ✓ Premium resources
  • ✕ Priority mentor support
Enterprise
$50/mo
For serious developers
Best for teams & advanced learning.
Go Enterprise
  • ✓ All Pro features
  • ✓ 1-on-1 mentorship
  • ✓ Team access
  • ✓ Project reviews
```

Next folder forms vid 27

This section explains **HTML form tags and input elements**, which are used to collect information from users on a webpage.

□ Key Points Explained

- **Forms in HTML** are used to **collect data** from users
  Examples: Login form, signup form, contact form, survey form
- **HTML builds the form structure only**
  HTML collects & displays inputs, but **does NOT process** the data
  (Processing is done via JavaScript or backend languages like PHP, Node.js, etc.)
- **CSS styles the form**
  Example: button colors, input box design, spacing

□ What You Will Learn in This Section

Topic	Description
`<form>` tag	Creates a form on a webpage
`<input>` types	Text, email, password, etc.
Form attributes	placeholder, disabled, required
Select box	Dropdown menu using `<select>`
Textarea	Large text field for messages
Checkbox	Allows multiple selections
Radio button	Choose **one** option from group
Date, color inputs	Special input types
Data lists	Auto-suggest drop-down list

`<form>` Tag Attributes

Attribute	Meaning
`action=""`	URL/endpoint where form data goes
`method=""`	How data is sent — `GET` or `POST`

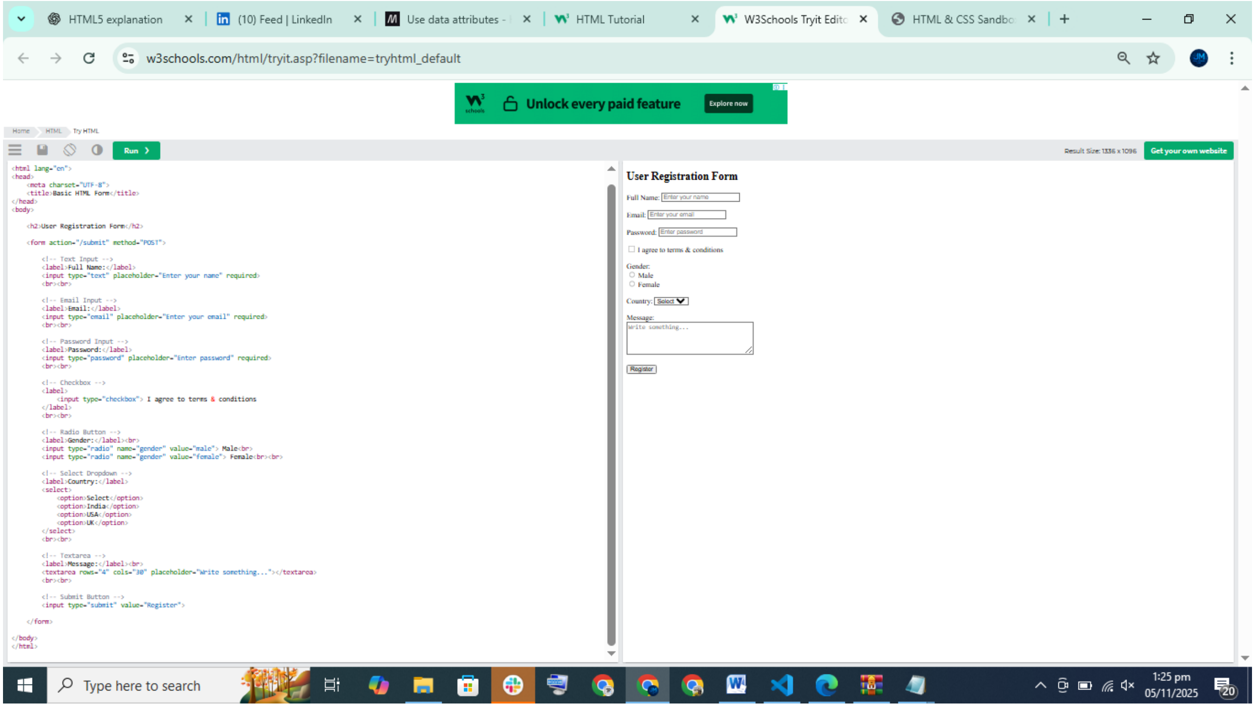
GET vs POST

Method	Purpose
GET	Fetch data (URL shows data)
POST	Send form data securely to server

Forms can only use **GET** or **POST**

□ Form Input Types Explained

Input Type	Use
`text`	Single-line text
`email`	Only accepts emails (browser checks format)
`password`	Hidden characters
`number`	Only numbers
`textarea`	Multi-line text (for long messages or blog body)
`select`	Dropdown list
`date`	Date picker
`checkbox`	Multiple select options
`radio`	Select only ONE option
`file`	Upload a file
`submit`	Submit form
`range`	Slider
`color`	Color picker



```html
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Basic HTML Form</title>
</head>
<body>

    <h2>User Registration Form</h2>

    <form action="/submit" method="POST">

        <!-- Text Input -->
        <label>Full Name:</label>
        <input type="text" placeholder="Enter your name" required>
        <br><br>

        <!-- Email Input -->
        <label>Email:</label>
        <input type="email" placeholder="Enter your email" required>
        <br><br>

        <!-- Password Input -->
        <label>Password:</label>
        <input type="password" placeholder="Enter password" required>
        <br><br>

        <!-- Checkbox -->
        <label>
            <input type="checkbox"> I agree to terms & conditions
        </label>
        <br><br>

        <!-- Radio Button -->
        <label>Gender:</label><br>
        <input type="radio" name="gender" value="male"> Male<br>
        <input type="radio" name="gender" value="female"> Female<br><br>

        <!-- Select Dropdown -->
        <label>Country:</label>
        <select>
            <option>Select</option>
            <option>India</option>
            <option>USA</option>
            <option>UK</option>
        </select>
        <br><br>

        <!-- Textarea -->
        <label>Message:</label><br>
        <textarea rows="4" cols="30" placeholder="Write something..."></textarea>
        <br><br>

        <!-- Submit Button -->
        <input type="submit" value="Register">

    </form>

</body>
</html>
```

User Registration Form

Full Name: Enter your name

Email: Enter your email

Password: Enter password

□ I agree to terms & conditions

Gender:
○ Male
○ Female

Country: Select

Message:
Write something...

Register

⍰ Understanding HTML Forms (Easy Explanation)

In this lesson, we learned how to create basic **HTML forms** to collect user information like name, email, and password.

- The `<form>` tag is used to create a form.
- The form can send data to a server using the `action` attribute (but we didn't use it here because we are just learning HTML first).

□ Input Fields

We used:

Type	Purpose
`type="text"`	For entering text like name
`type="email"`	For email input (checks valid email format)
`type="password"`	Hides text while typing (for passwords)

□ Label Tag

- `<label>` describes the input field.
- The **for** attribute connects label to input using `id`.
- Clicking the label focuses the input field — good for accessibility.

□ Block & Inline Behavior

- Labels and inputs are **inline** by default = same line.
- To move them on new lines, we used `<br>` OR wrapped each in a `<div>` (better practice).

□ Submit Button

Two ways to submit the form:

```
<input type="submit" value="Register">
```

OR

```
<button type="submit">Register</button>
```

Both submit the form data.

□ Extra Learning Tips

- `email` input validates email automatically.
- Use `id` for every input and match with `label for=""`.
- Better form spacing is done using CSS instead of `<br>`.

Vid 29

⍰ 1. `placeholder`

Shows grey text inside the input before the user types.

```
<input type="text" placeholder="Enter name">
```

Purpose:
Helps users know what to type, especially when there is no label.

□ 2. `value`

Pre-fills the input with default text.

```
<input type="text" value="John Doe">
```

Used when editing existing data (like editing a profile or blog post).

□ 3. `name`

Very important for sending form data to the server.

```
<input type="text" name="username">
```

When you submit the form, this becomes the key:

```
?username=John+Doe
```

Without `name`, the input isn't included in the form submission.

□ 4. GET vs POST

Forms default to **GET** (data shows in URL):

```
?name=John&email=test@test.com
```

▶ **GET** — visible in URL, not secure
▶ **POST** — hidden, used for passwords/user data

Use POST for login, signup, payment forms:

```
<form action="process.php" method="post">
```

□ 5. `required`

Makes input mandatory.

```
<input required>
```

Browser prevents form submission if empty.

□ 6. `minlength` and `maxlength`

Restrict input length.

```
<input minlength="3" maxlength="10">
```

Useful for usernames, passwords, etc.

□ 7. `disabled`

Locks the field — user can't change it.

```
<input value="test@email.com" disabled>
```

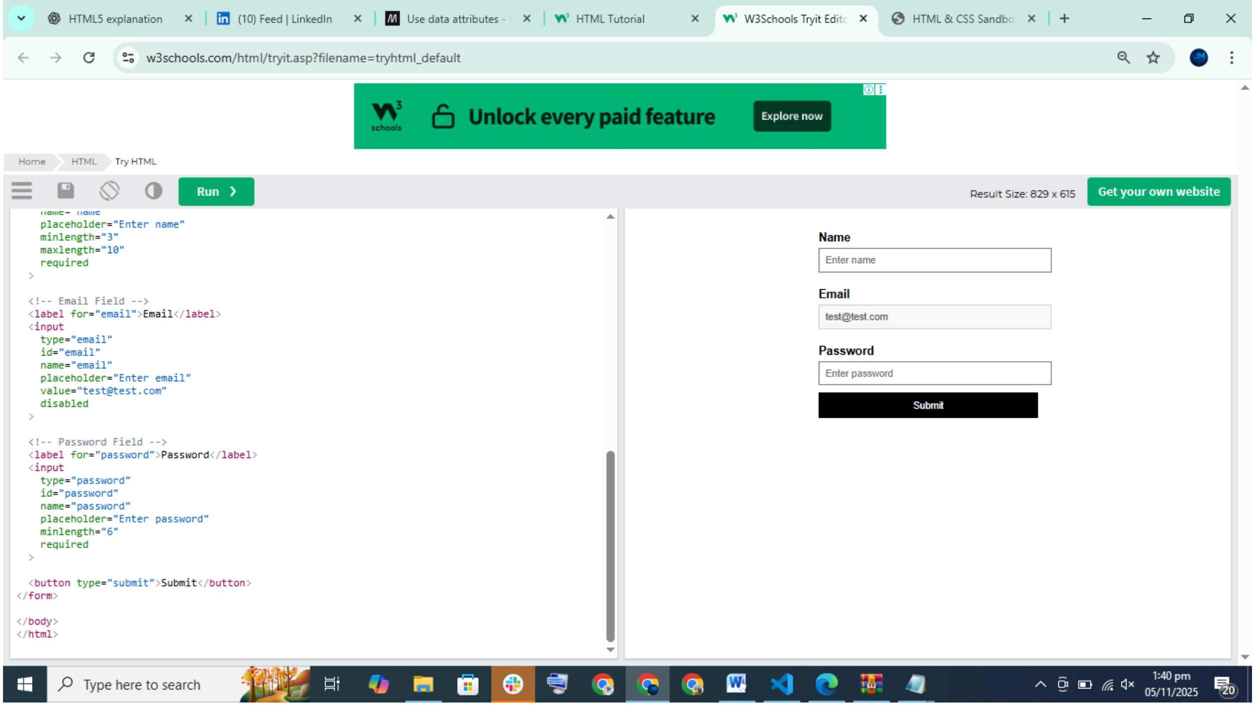
Greyed out → not sent with form.

□ Summary Table

Attribute	Purpose
placeholder	Hint text inside input
value	Default text inside input
name	Key used when sending data
method="post"	Secure form submission
required	Must fill before submit
minlength / maxlength	Text length limits
disabled	Field can't be edited

□ Coming Next Lesson

- Select dropdowns `<select>`
- Multi-line text input `<textarea>`



```
    name="name"
    placeholder="Enter name"
    minlength="3"
    maxlength="10"
    required
  >

  <!-- Email Field -->
  <label for="email">Email</label>
  <input
    type="email"
    id="email"
    name="email"
    placeholder="Enter email"
    value="test@test.com"
    disabled
  >

  <!-- Password Field -->
  <label for="password">Password</label>
  <input
    type="password"
    id="password"
    name="password"
    placeholder="Enter password"
    minlength="6"
    required
  >

  <button type="submit">Submit</button>
</form>

</body>
</html>
```

Name

Enter name

Email

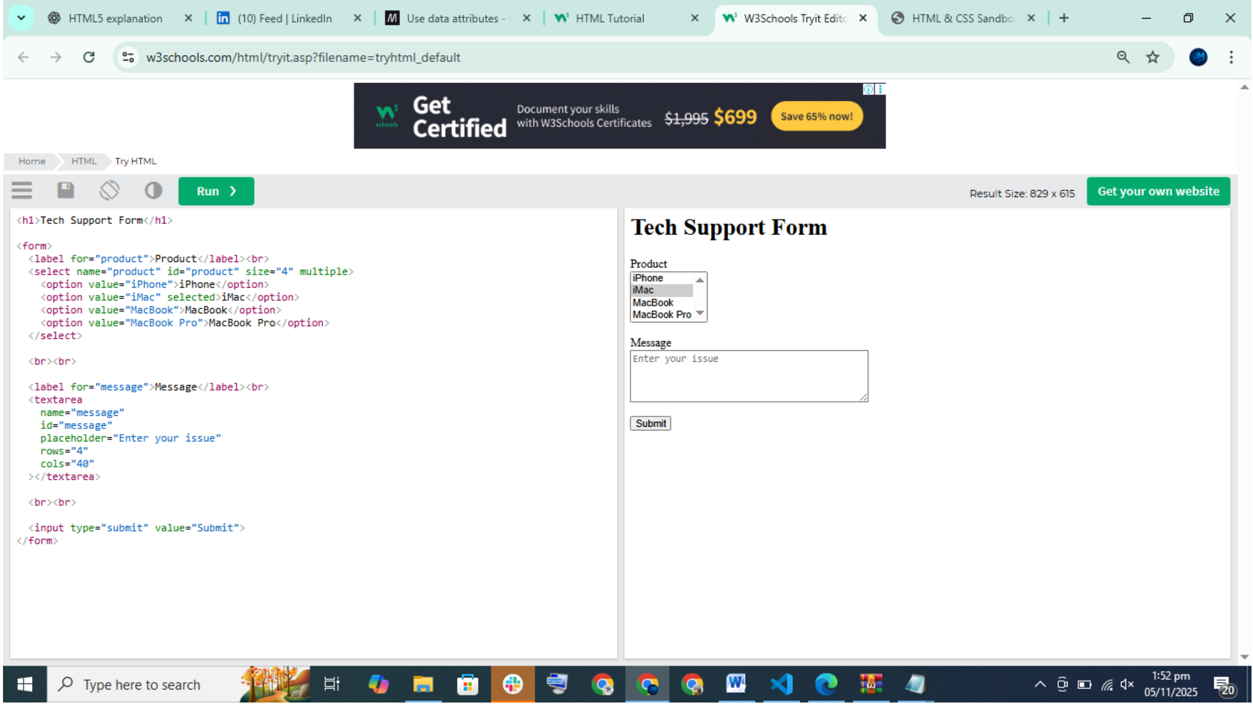
test@test.com

Password

Enter password

Submit

Vid 30



```
<h1>Tech Support Form</h1>

<form>
  <label for="product">Product</label><br>
  <select name="product" id="product" size="4" multiple>
    <option value="iPhone">iPhone</option>
    <option value="iMac" selected>iMac</option>
    <option value="MacBook">MacBook</option>
    <option value="MacBook Pro">MacBook Pro</option>
  </select>

  <br><br>

  <label for="message">Message</label><br>
  <textarea
    name="message"
    id="message"
    placeholder="Enter your issue"
    rows="4"
    cols="40"
  ></textarea>

  <br><br>

  <input type="submit" value="Submit">
</form>
```

Tech Support Form

Product

iPhone
iMac
MacBook
MacBook Pro

Message

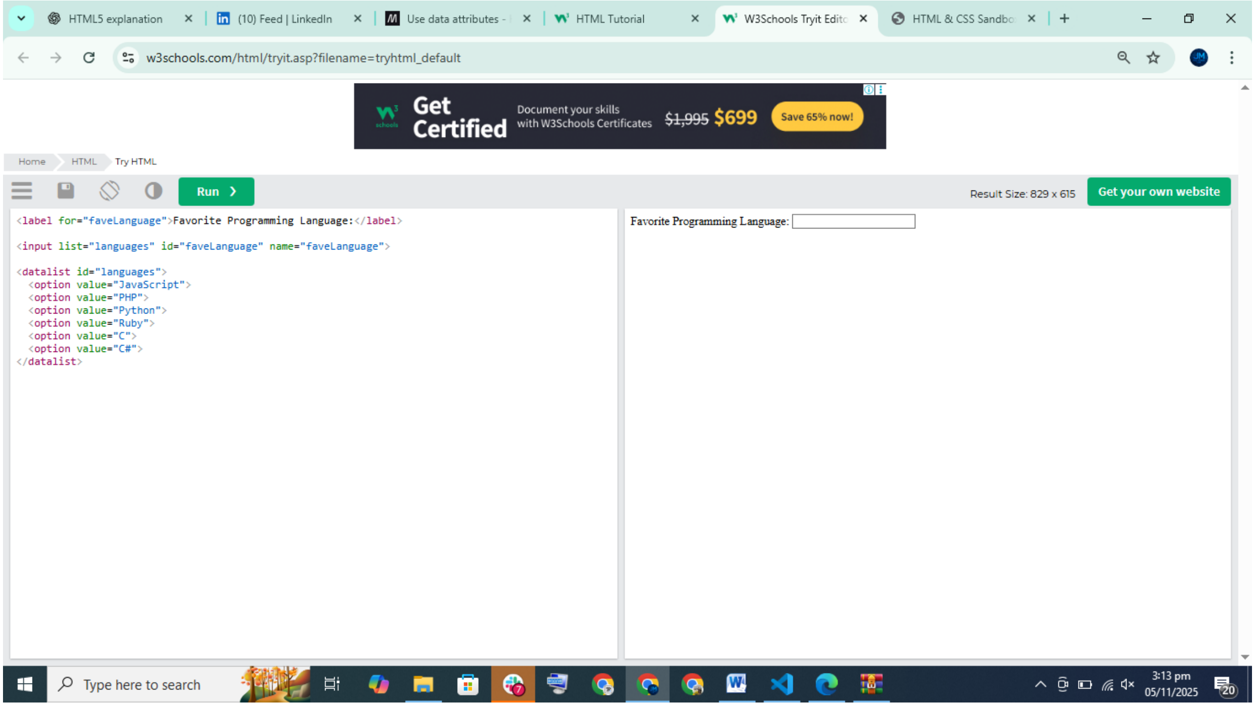
Enter your issue

Submit

Key Learning Points

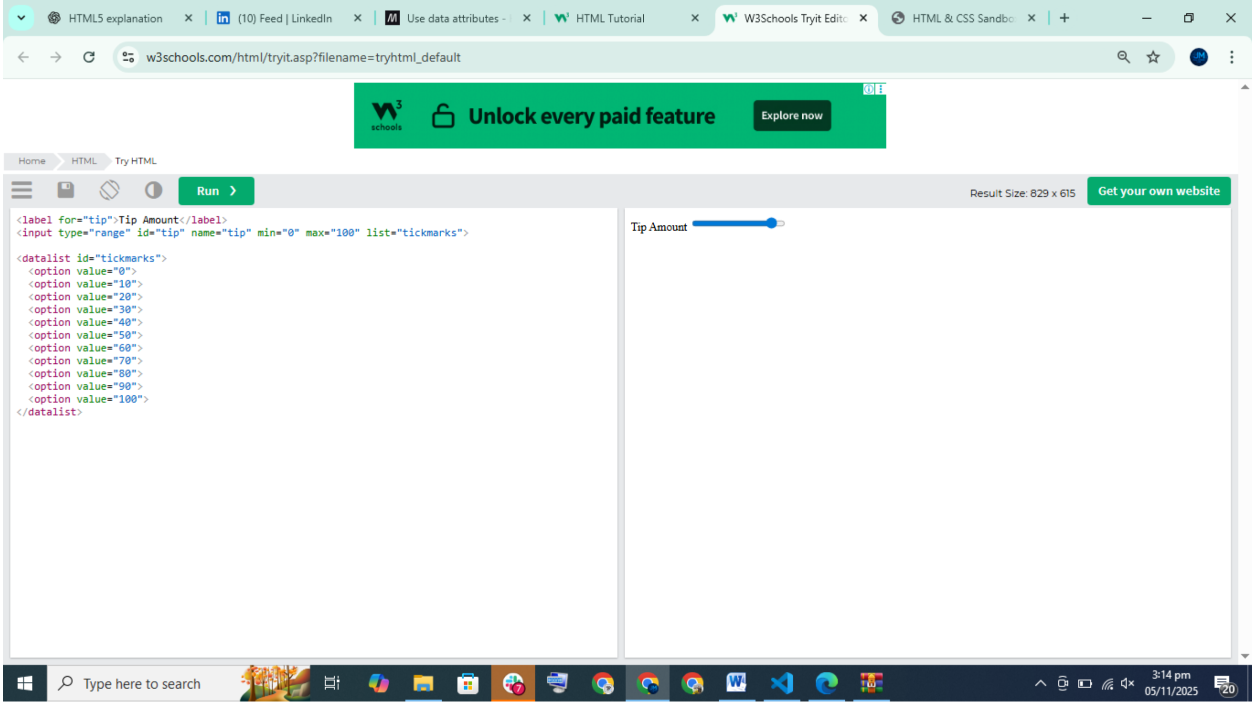
Concept	Meaning
`<select>`	dropdown input
`<option>`	each item inside dropdown
`multiple`	select multiple options
`size`	how many options visible

Concept	Meaning
`textarea`	multi-line input area
`rows / cols`	size control



```
<label for="faveLanguage">Favorite Programming Language:</label>

<input list="languages" id="faveLanguage" name="faveLanguage">

<datalist id="languages">
  <option value="JavaScript">
  <option value="PHP">
  <option value="Python">
  <option value="Ruby">
  <option value="C">
  <option value="C#">
</datalist>
```

Favorite Programming Language:



```
<label for="tip">Tip Amount</label>
<input type="range" id="tip" name="tip" min="0" max="100" list="tickmarks">

<datalist id="tickmarks">
  <option value="0">
  <option value="10">
  <option value="20">
  <option value="30">
  <option value="40">
  <option value="50">
  <option value="60">
  <option value="70">
  <option value="80">
  <option value="90">
  <option value="100">
</datalist>
```

Tip Amount

<!DOCTYPE html>

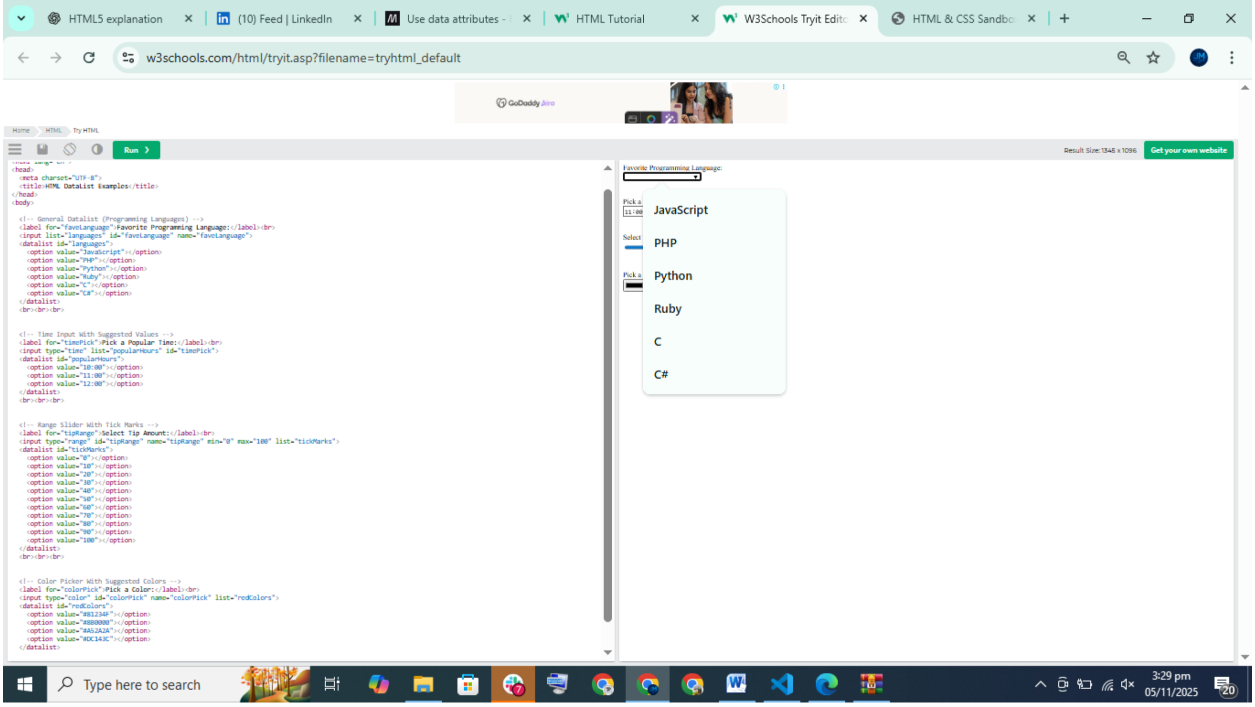
<html lang="en">

<head>

```
  <meta charset="UTF-8">
  <title>HTML DataList Examples</title>
</head>
<body>

  <!-- General Datalist (Programming Languages) -->
  <label for="faveLanguage">Favorite Programming Language:</label><br>
  <input list="languages" id="faveLanguage" name="faveLanguage">
  <datalist id="languages">
    <option value="JavaScript"></option>
    <option value="PHP"></option>
    <option value="Python"></option>
    <option value="Ruby"></option>
    <option value="C"></option>
    <option value="C#"></option>
  </datalist>
  <br><br><br>


  <!-- Time Input With Suggested Values -->
  <label for="timePick">Pick a Popular Time:</label><br>
  <input type="time" list="popularHours" id="timePick">
  <datalist id="popularHours">
    <option value="10:00"></option>
    <option value="11:00"></option>
    <option value="12:00"></option>
  </datalist>
  <br><br><br>
```

```
 <!-- Range Slider With Tick Marks -->

 <label for="tipRange">Select Tip Amount:</label><br>

 <input type="range" id="tipRange" name="tipRange" min="0" max="100" list="tickMarks">

 <datalist id="tickMarks">

   <option value="0"></option>

   <option value="10"></option>

   <option value="20"></option>

   <option value="30"></option>

   <option value="40"></option>

   <option value="50"></option>

   <option value="60"></option>

   <option value="70"></option>

   <option value="80"></option>

   <option value="90"></option>

   <option value="100"></option>

 </datalist>

 <br><br><br>



 <!-- Color Picker With Suggested Colors -->

 <label for="colorPick">Pick a Color:</label><br>

 <input type="color" id="colorPick" name="colorPick" list="redColors">

 <datalist id="redColors">

   <option value="#81234F"></option>

   <option value="#8B0000"></option>

   <option value="#A52A2A"></option>

   <option value="#DC143C"></option>

 </datalist>


</body>
```

</html>



```html
<head>
  <meta charset="UTF-8">
  <title>HTML DataList Examples</title>
</head>
<body>

  <!-- General Datalist (Programming Languages) -->
  <label for="faveLanguage">Favorite Programming Language:</label><br>
  <input list="languages" id="faveLanguage" name="faveLanguage">
  <datalist id="languages">
    <option value="JavaScript"></option>
    <option value="PHP"></option>
    <option value="Python"></option>
    <option value="Ruby"></option>
    <option value="C"></option>
    <option value="C#"></option>
  </datalist>
  <br><br><br>


  <!-- Time Input With Suggested Values -->
  <label for="timePick">Pick a Popular Time:</label><br>
  <input type="time" list="popularHours" id="timePick">
  <datalist id="popularHours">
    <option value="10:00"></option>
    <option value="11:00"></option>
    <option value="12:00"></option>
  </datalist>
  <br><br><br>


  <!-- Range Slider With Tick Marks -->
  <label for="tipRange">Select Tip Amount:</label><br>
  <input type="range" id="tipRange" name="tipRange" min="0" max="100" list="tickMarks">
  <datalist id="tickMarks">
    <option value="0"></option>
    <option value="10"></option>
    <option value="20"></option>
    <option value="30"></option>
    <option value="40"></option>
    <option value="50"></option>
    <option value="60"></option>
    <option value="70"></option>
    <option value="80"></option>
    <option value="90"></option>
    <option value="100"></option>
  </datalist>
  <br><br><br>


  <!-- Color Picker With Suggested Colors -->
  <label for="colorPick">Pick a Color:</label><br>
  <input type="color" id="colorPick" name="colorPick" list="redColors">
  <datalist id="redColors">
    <option value="#81234F"></option>
    <option value="#8B0000"></option>
    <option value="#A52A2A"></option>
    <option value="#DC143C"></option>
  </datalist>
```

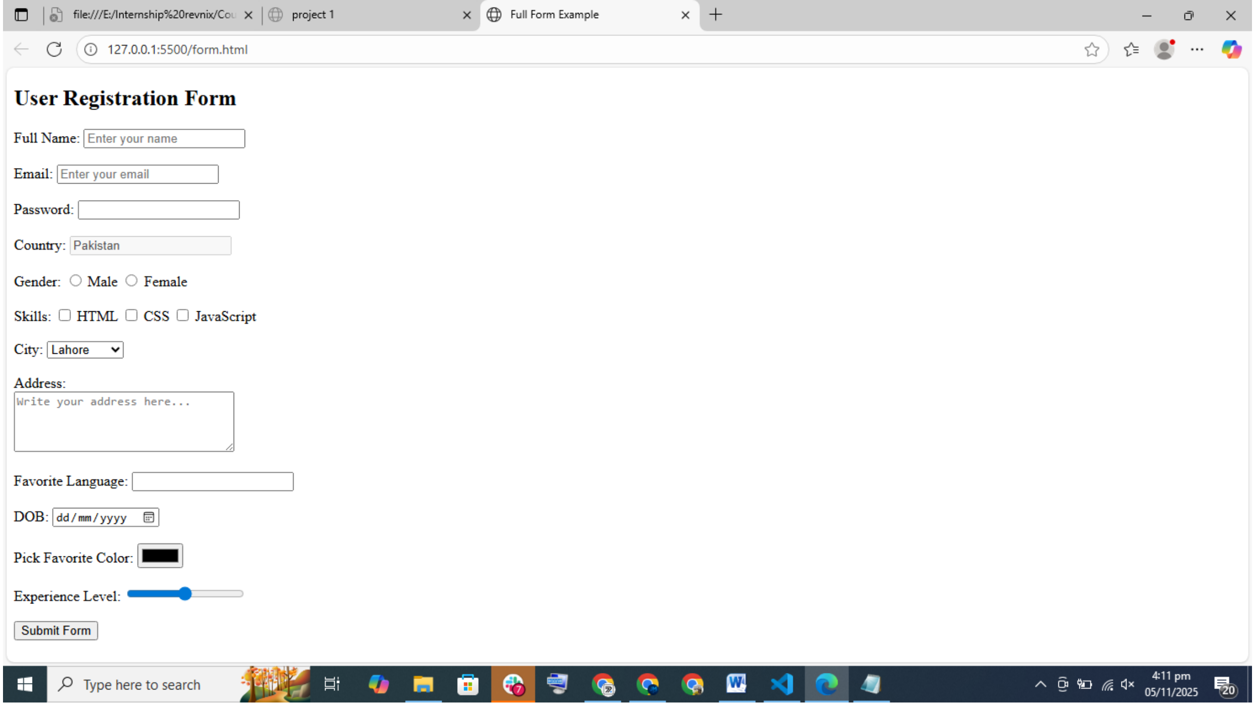
Favorite Programming Language:

JavaScript
PHP
Python
Ruby
C
C#

Pick a
11:00

Select

Pick a



User Registration Form

Full Name: Enter your name

Email: Enter your email

Password:

Country: Pakistan

Gender: ○ Male ○ Female

Skills: □ HTML □ CSS □ JavaScript

City: Lahore

Address:
Write your address here...

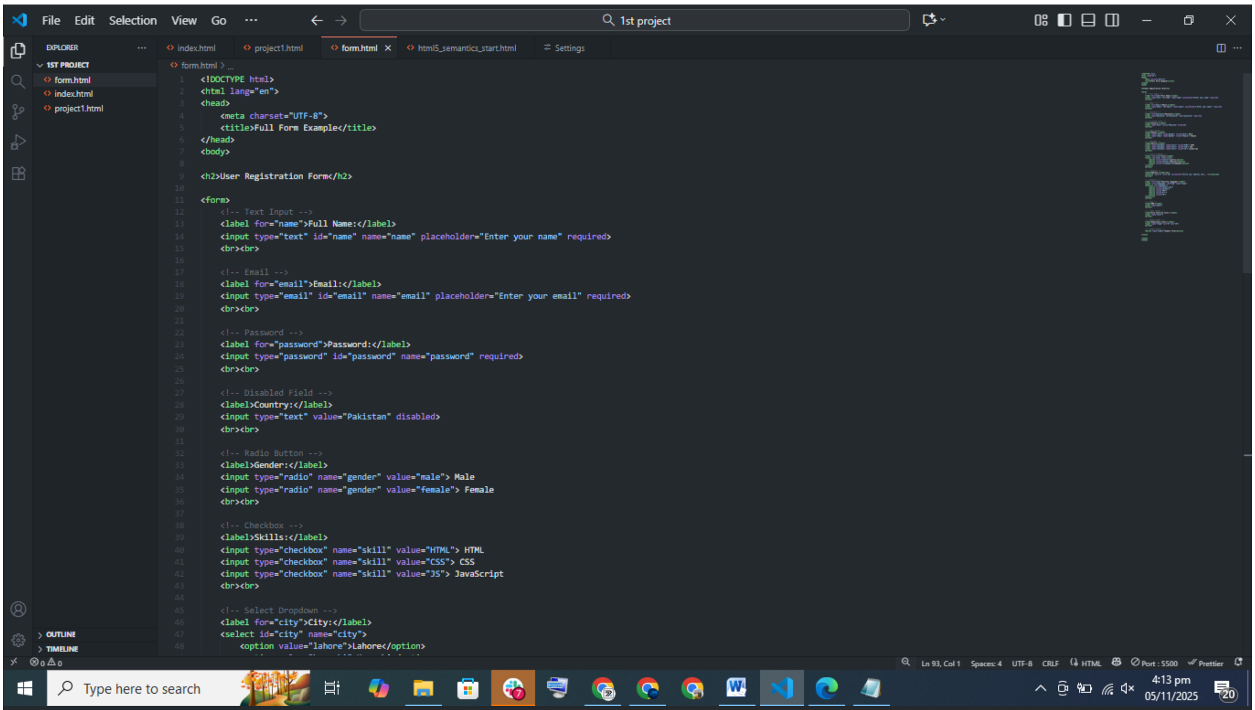
Favorite Language:

DOB: dd/mm/yyyy

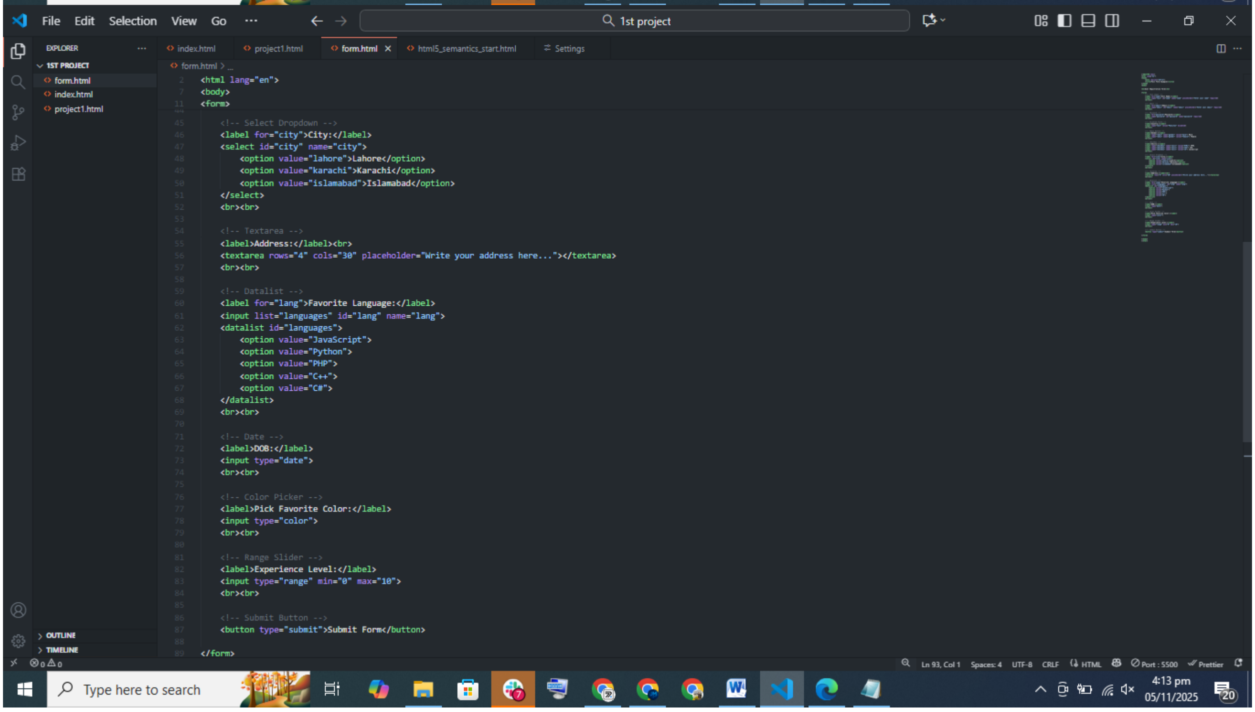
Pick Favorite Color:

Experience Level:

Submit Form



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Full Form Example</title>
</head>
<body>

<h2>User Registration Form</h2>

<form>
    <!-- Text Input -->
    <label for="name">Full Name:</label>
    <input type="text" id="name" name="name" placeholder="Enter your name" required>
    <br><br>

    <!-- Email -->
    <label for="email">Email:</label>
    <input type="email" id="email" name="email" placeholder="Enter your email" required>
    <br><br>

    <!-- Password -->
    <label for="password">Password:</label>
    <input type="password" id="password" name="password" required>
    <br><br>

    <!-- Disabled Field -->
    <label>Country:</label>
    <input type="text" value="Pakistan" disabled>
    <br><br>

    <!-- Radio Button -->
    <label>Gender:</label>
    <input type="radio" name="gender" value="male"> Male
    <input type="radio" name="gender" value="female"> Female
    <br><br>

    <!-- Checkbox -->
    <label>Skills:</label>
    <input type="checkbox" name="skill" value="HTML"> HTML
    <input type="checkbox" name="skill" value="CSS"> CSS
    <input type="checkbox" name="skill" value="JS"> JavaScript
    <br><br>

    <!-- Select Dropdown -->
    <label for="city">City:</label>
    <select id="city" name="city">
        <option value="lahore">Lahore</option>
```



```html
<html lang="en">
<body>
<form>
    <!-- Select Dropdown -->
    <label for="city">City:</label>
    <select id="city" name="city">
        <option value="lahore">Lahore</option>
        <option value="karachi">Karachi</option>
        <option value="islamabad">Islamabad</option>
    </select>
    <br><br>

    <!-- Textarea -->
    <label>Address:</label><br>
    <textarea rows="4" cols="30" placeholder="Write your address here..."></textarea>
    <br><br>

    <!-- Datalist -->
    <label for="lang">Favorite Language:</label>
    <input list="languages" id="lang" name="lang">
    <datalist id="languages">
        <option value="JavaScript">
        <option value="Python">
        <option value="PHP">
        <option value="C++">
        <option value="C#">
    </datalist>
    <br><br>

    <!-- Date -->
    <label>DOB:</label>
    <input type="date">
    <br><br>

    <!-- Color Picker -->
    <label>Pick Favorite Color:</label>
    <input type="color">
    <br><br>

    <!-- Range Slider -->
    <label>Experience Level:</label>
    <input type="range" min="0" max="10">
    <br><br>

    <!-- Submit Button -->
    <button type="submit">Submit Form</button>

</form>
```