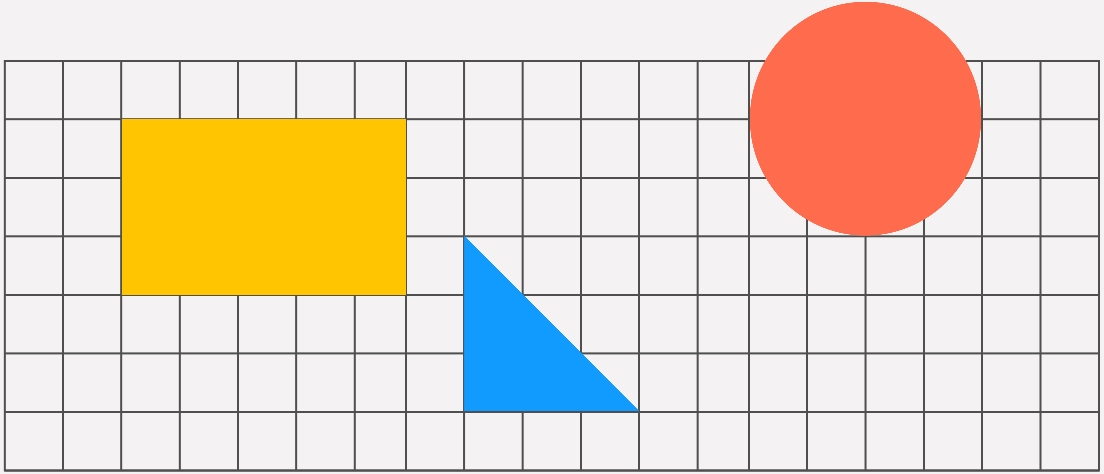
The Building Blocks of the Web

Ali Abrishami



In This Lecture You Will...

a

Learn What HTML is

b

Learn How HTML is used to transfer content

c

Get started coding!

The Communication Age

- We live in the most connected era in human history.
- The idea of a global network began with ARPANET, created by the U.S. Department of Defense.
- The idea was to build a decentralized network for secure military communications, but eventually, the breakthrough evolved into a framework for interconnecting universities, research centers, and ultimately, the global internet.
- Researchers and universities started linking computers to share data globally (1980s).
- Tim Berners-Lee invented the World Wide Web, introducing HTML and revolutionizing communication. (1991-1992)
- At the core of this revolution: the Web; and it all starts with HTML.

The Three Layers of a Website

Structure

What the **content** is

Style

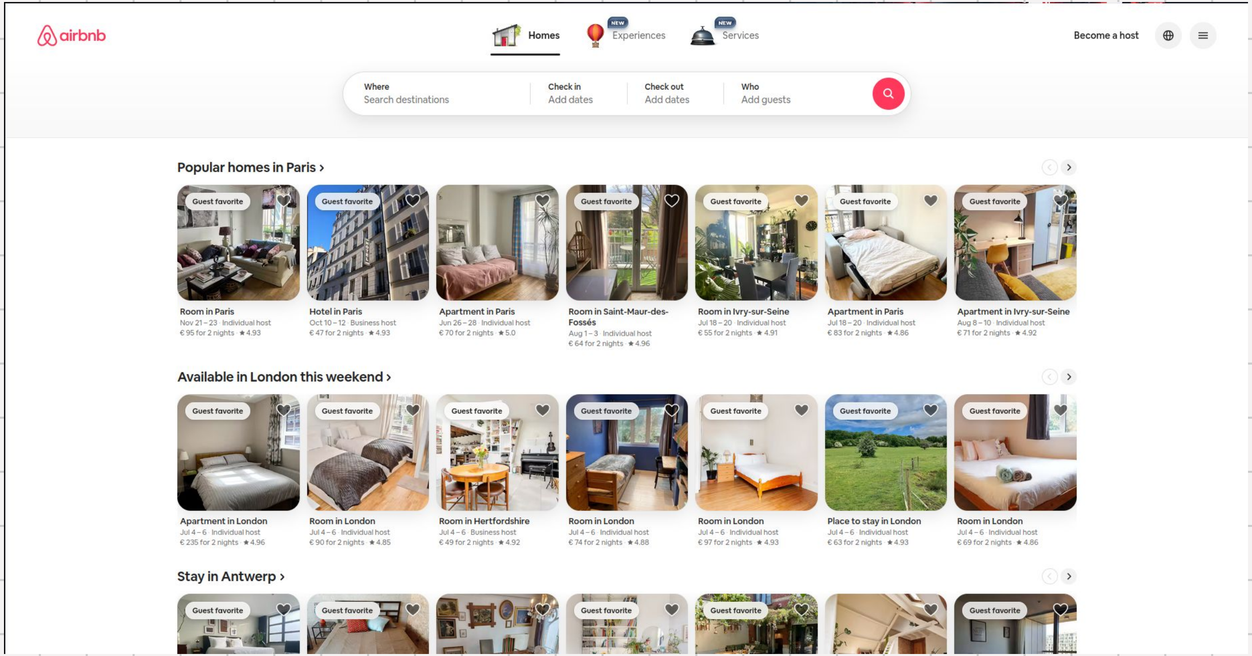
How it **looks**

Behaviour

How it **reacts**

Class Activity Time

Based on the screenshot of Airbnb's homepage below, what are some elements that belong to the Structure, Style, and Behavior layers?



Popular homes in Paris ›

Room in Paris
Nov 21 – 23 · Individual host
€ 95 for 2 nights · ★ 4.93

Hotel in Paris
Oct 10 – 12 · Business host
€ 47 for 2 nights · ★ 4.93

Apartment in Paris
Jun 26 – 28 · Individual host
€ 70 for 2 nights · ★ 5.0

Room in Saint-Maur-des-Fossés
Aug 1 – 3 · Individual host
€ 64 for 2 nights · ★ 4.96

Room in Ivry-sur-Seine
Jul 18 – 20 · Individual host
€ 55 for 2 nights · ★ 4.91

Apartment in Paris
Jul 18 – 20 · Individual host
€ 83 for 2 nights · ★ 4.86

Apartment in Ivry-sur-Seine
Aug 8 – 10 · Individual host
€ 71 for 2 nights · ★ 4.92

Available in London this weekend ›

Apartment in London
Jul 4 – 6 · Individual host
€ 235 for 2 nights · ★ 4.96

Room in London
Jul 4 – 6 · Individual host
€ 90 for 2 nights · ★ 4.85

Room in Hertfordshire
Jul 4 – 6 · Business host
€ 49 for 2 nights · ★ 4.92

Room in London
Jul 4 – 6 · Individual host
€ 74 for 2 nights · ★ 4.88

Room in London
Jul 4 – 6 · Individual host
€ 97 for 2 nights · ★ 4.93

Place to stay in London
Jul 4 – 6 · Individual host
€ 63 for 2 nights · ★ 4.93

Room in London
Jul 4 – 6 · Individual host
€ 69 for 2 nights · ★ 4.86

Stay in Antwerp ›

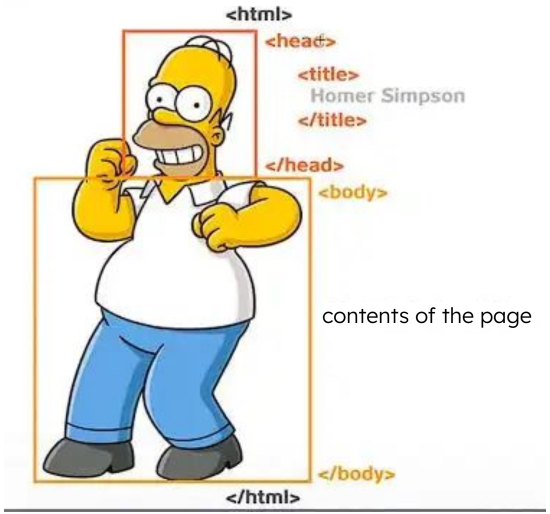
What is HTML?

Introduction

- HTML is short for HyperText Markup Language.
- Web developers use HTML to **structure** and **describe** the **content** of a webpage.
- HTML consists of **elements** that describe **different types**.
- Web browsers **understand** HTML and **render** HTML code as **websites.**
- HTML is **not** a programming language.

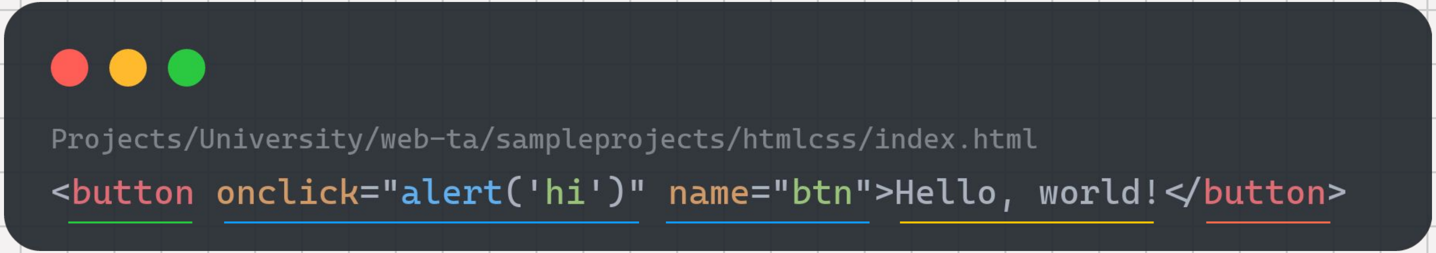
HTML Document Parts

- An HTML5 document starts with <!DOCTYPE html> to define the HTML version.
- The `<html>` element wraps the entire document content.
- The `<head>` section contains meta information like title, charset, and links to scripts or icons.
- The `<body>` section holds all the visible content of the webpage.
- Tags are structured hierarchically and must be properly opened and closed.



Source: DIO

Someone Said Elements?



```
Projects/University/web-ta/sampleprojects/htmlcss/index.html
<button onclick="alert('hi')" name="btn">Hello, world!</button>
```

- What you are seeing is an HTML element:
    - It has an opening tag <button>
    - A closing tag </button>
    - Key-value properties
    - Child element(s) or content

The Things You Can’t See!

- The `<head>` element contains information about the page, not displayed directly.
- It helps browsers, search engines, and devices understand your page.
- Common elements inside `<head>` include:
    - `<title>` sets the page title (shown on browser tab)
    - `<meta>` defines metadata like charset, author, and description
    - `<link>` connects external resources like stylesheets or icons

Class Activity Time

What exactly is SEO?

What is `<meta>` tag good for?

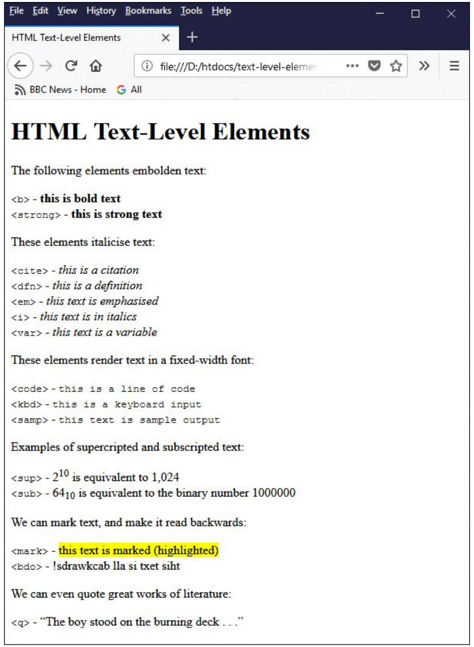
Why would you want to include things that don’t appear to user?

What Elements Can We Use?

- HTML gives us a bunch of elements to build everything we see on a website.
- Some elements are used to structure the page, like headers, sections, and footers.
- Others control content, like text, links, images, and buttons.
- And some elements help us interact, like forms and inputs.
- We will learn many of them as we progress through the course

Text Elements

- Almost everything on the web starts with text.
- HTML provides **semantic elements** to describe the purpose of each piece of text.
- These elements help both browsers and developers understand how content should be displayed or interpreted.
- **Important note:** Always choose elements that reflect the content's intended meaning, not just how it looks.


File Edit View History Bookmarks Tools Help
HTML Text-Level Elements
file:///D:/htdocs/text-level-elemer
BBC News - Home
All

HTML Text-Level Elements

The following elements embolden text:

<b> - **this is bold text**
<strong> - **this is strong text**

These elements italicise text:

<cite> - *this is a citation*
<dfn> - *this is a definition*
<em> - *this text is emphasised*
<i> - *this text is in italics*
<var> - *this text is a variable*

These elements render text in a fixed-width font:

```
<code> - this is a line of code
<kbd> - this is a keyboard input
<samp> - this text is sample output
```

Examples of supercripted and subscripted text:

<sup> - 2^{10} is equivalent to 1,024
<sub> - 64_{10} is equivalent to the binary number 1000000

We can mark text, and make it read backwards:

<mark> - this text is marked (highlighted)
<bdo> - !sdrawkcab lla si txet siht

We can even quote great works of literature:

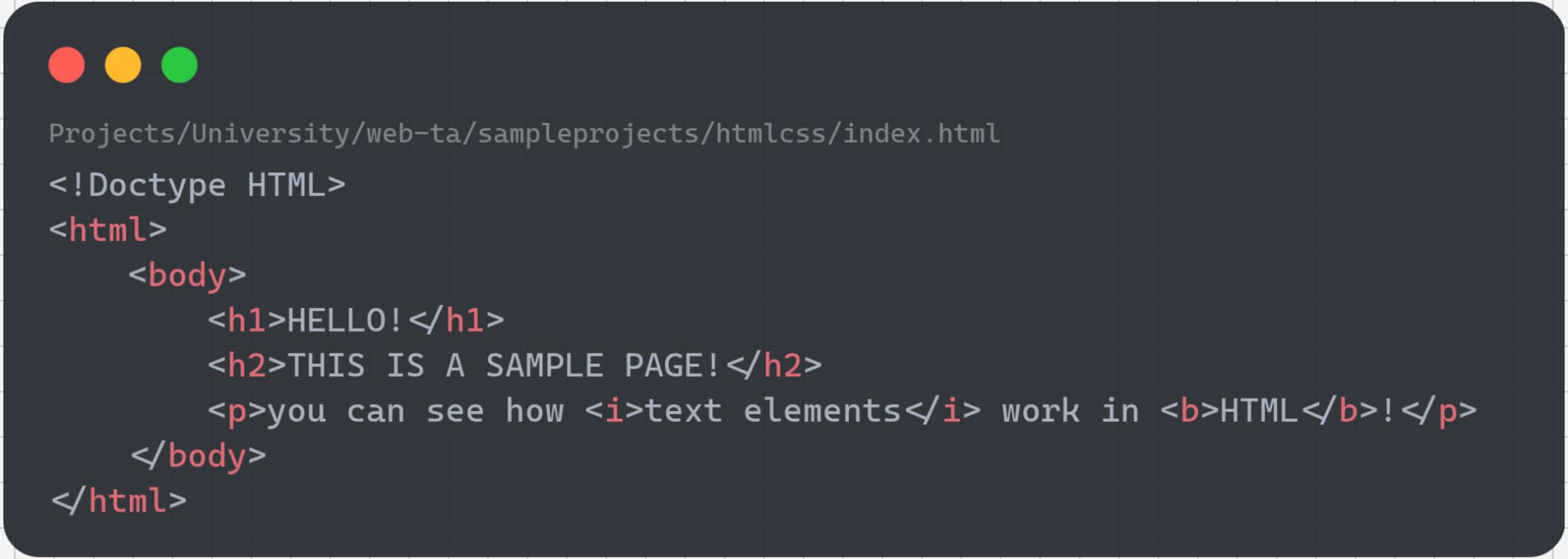
<q> - "The boy stood on the burning deck . . ."

source: technologyuk.net

Text Element Tags

- `<p>` Used commonly for paragraphs
- `<h1>~~<h6>` Headings
- `<strong>` or `<b>` Bold font
- `<em>` or `<i>` Italic font
- `<code>` Displays a piece of computer code

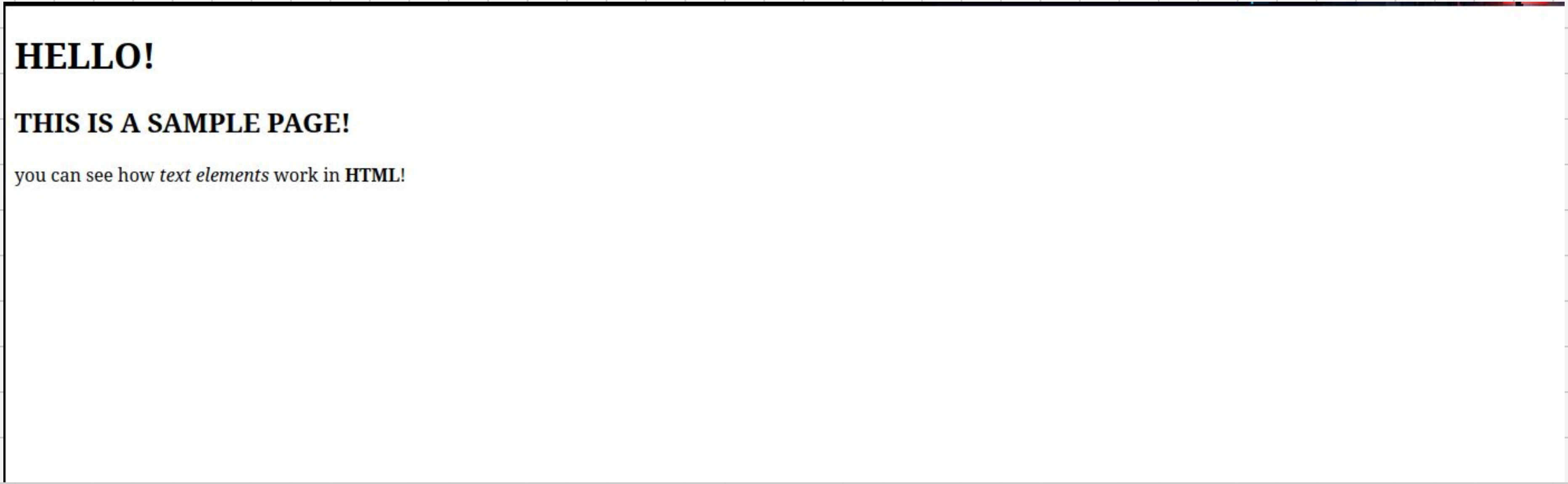
Sample Text-Only Page



Projects/University/web-ta/sampleprojects/htmlcss/index.html

```
<!Doctype HTML>
<html>
    <body>
        <h1>HELLO!</h1>
        <h2>THIS IS A SAMPLE PAGE!</h2>
        <p>you can see how <i>text elements</i> work in <b>HTML</b>!</p>
    </body>
</html>
```

And The Result:



HELLO!

THIS IS A SAMPLE PAGE!

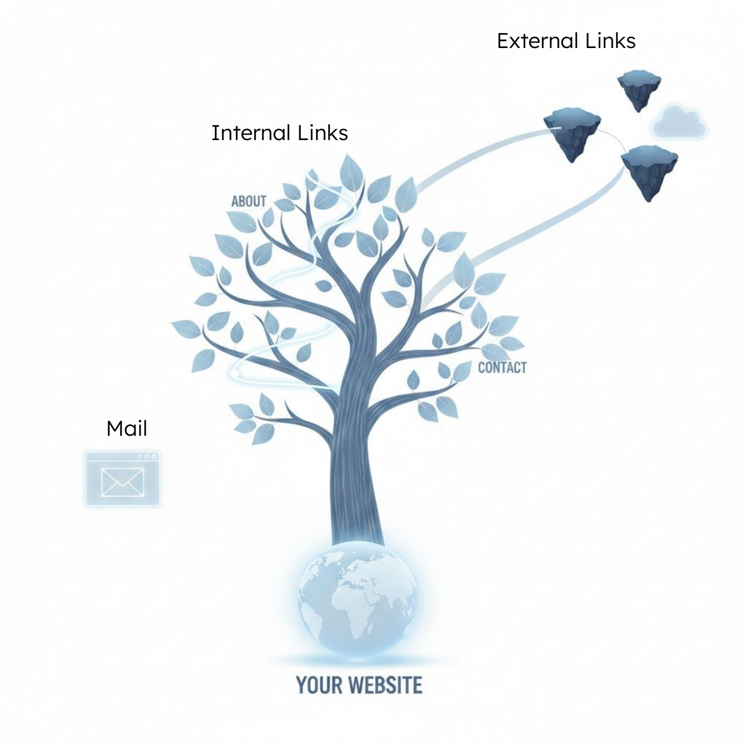
you can see how *text elements* work in **HTML**!

Hyperlinks

- Hyperlinks are created using the `<a>` (anchor) tag.
- The `href` attribute specifies the URL or location to link to.
- Links can point to external sites, internal pages, or specific page sections.
- You can open links in a new tab using the `target="_blank"` attribute.
- You can use hyperlinks to trigger file downloads by adding the `download` attribute.
- You can link to email addresses using `mailto:`, e.g.,
  `<a href="mailto:info@example.com">Email Us</a>`.

hrefs Point to What?

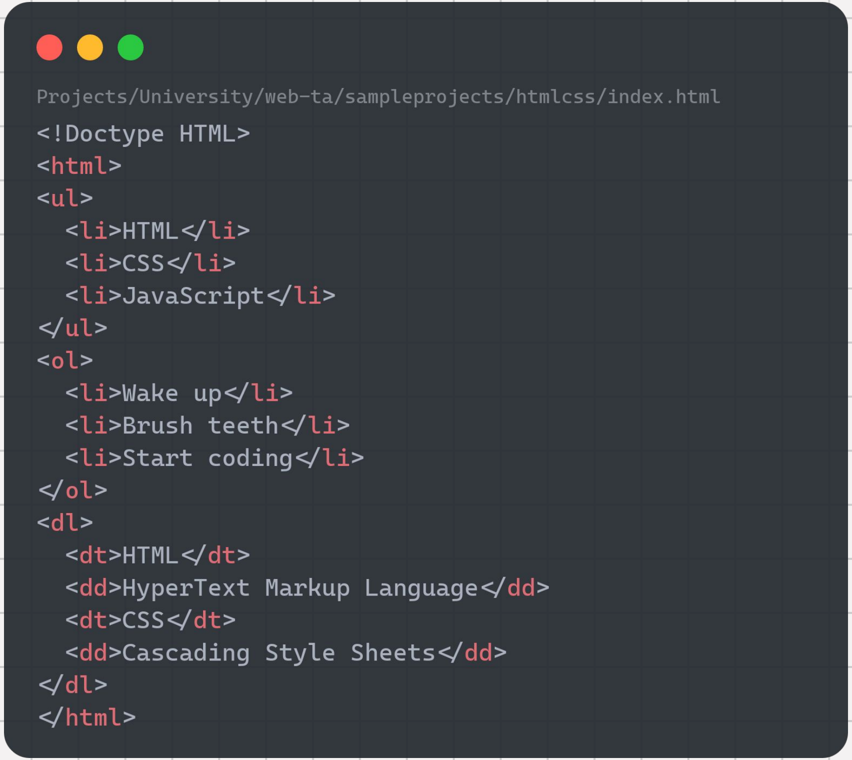
- Internal links: point to pages within the same website (e.g. /about)
- External links: point to other websites (e.g. https://wikipedia.org)
- Anchor links: jump to a section within the same page using `id` (e.g. #top)
- Mail links: open the user’s email client to send email (e.g. mailto:sharif@sharif.ir)
- Tel links: allow user to call a phone number (e.g. tel:0215421234)



List Elements

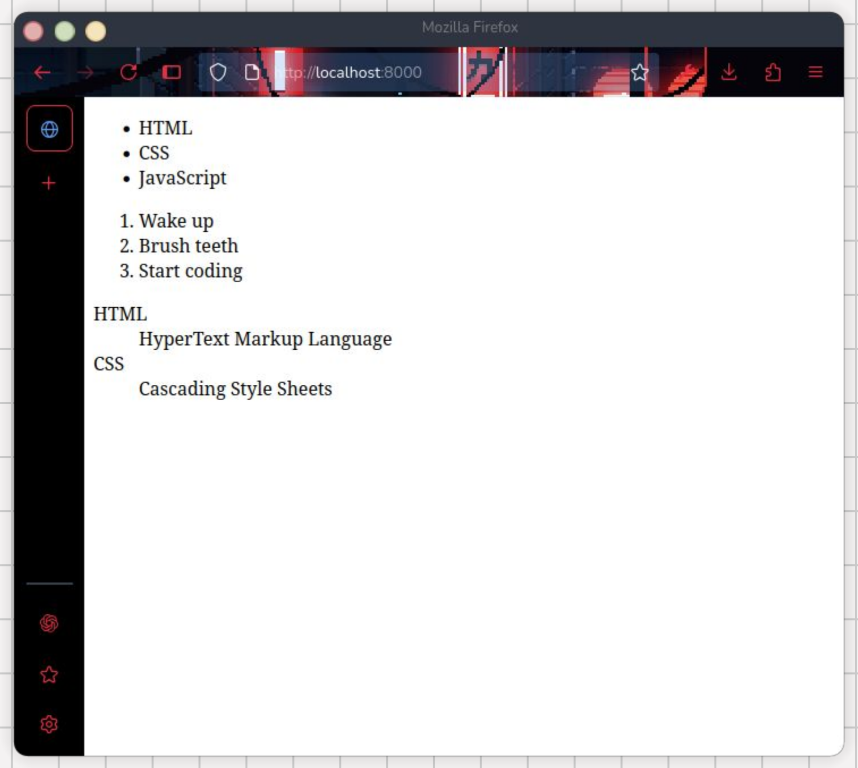
- HTML supports three main types of lists: ordered, unordered, and description lists.
- Use `<ol>` for ordered lists (numbered).
- Use `<ul>` for unordered lists (bulleted).
- Use `<dl>` with `<dt>` and `<dd>` for description lists (terms and definitions).

List Elements Example



Projects/University/web-ta/sampleprojects/htmlcss/index.html

```
<!Doctype HTML>
<html>
<ul>
  <li>HTML</li>
  <li>CSS</li>
  <li>JavaScript</li>
</ul>
<ol>
  <li>Wake up</li>
  <li>Brush teeth</li>
  <li>Start coding</li>
</ol>
<dl>
  <dt>HTML</dt>
  <dd>HyperText Markup Language</dd>
  <dt>CSS</dt>
  <dd>Cascading Style Sheets</dd>
</dl>
</html>
```



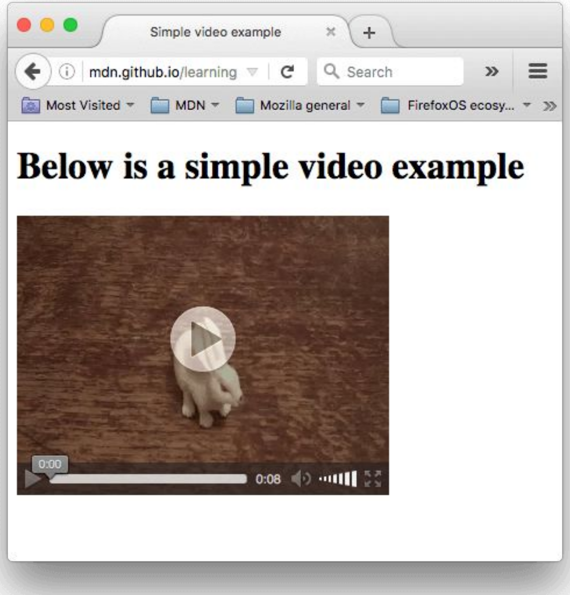
```
    • HTML
    • CSS
    • JavaScript

   1. Wake up
   2. Brush teeth
   3. Start coding

HTML
        HyperText Markup Language
CSS
        Cascading Style Sheets
```

Comments

- Use `<!-- comment -->` to add notes in your code.
- Comments are not displayed in the browser.
- Useful for leaving reminders or explanations for yourself or others.
- Can also be used to temporarily hide parts of the code during development.

Media Elements

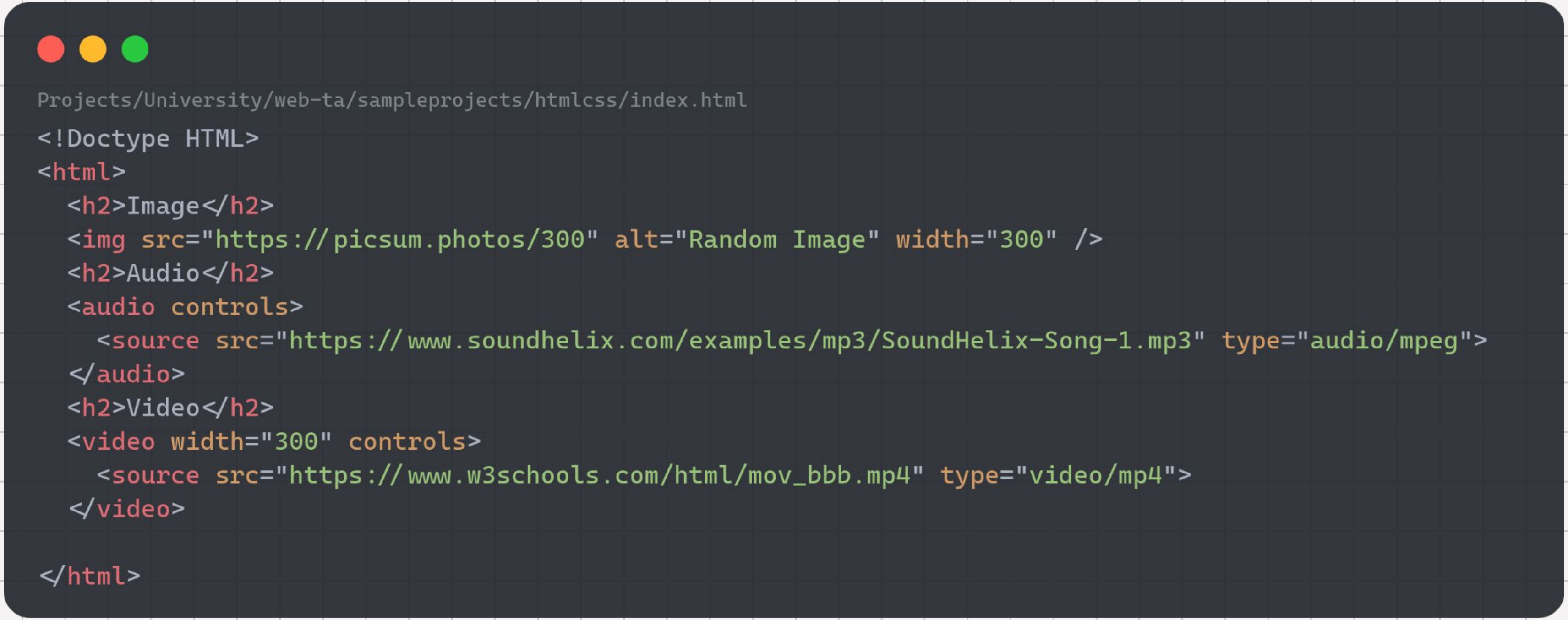
- HTML provides elements to embed audio, video, and images directly into web pages.
- Common media tags include `<img>`, `<audio>`, and `<video>`.
- You can control playback with built-in attributes like controls, autoplay, and loop.
- Media elements support fallback content for unsupported browsers.



Below is a simple video example

source: MDN Web Docs

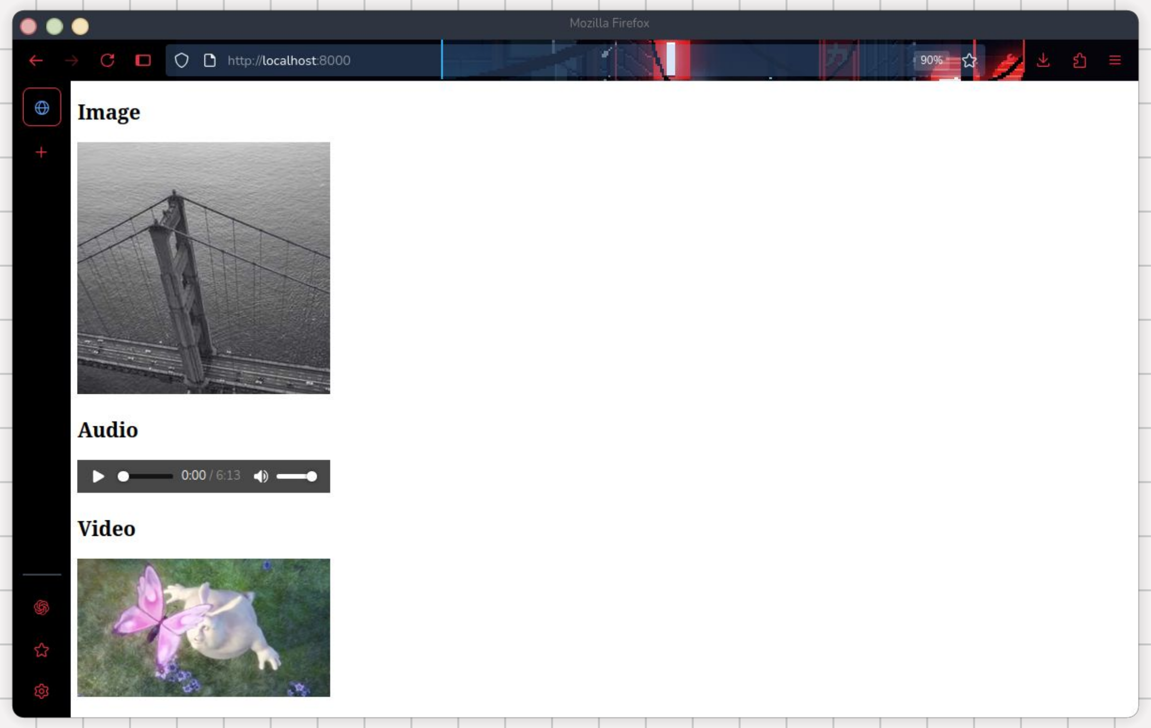
Media Elements Example



Projects/University/web-ta/sampleprojects/htmlcss/index.html

```
<!Doctype HTML>
<html>
  <h2>Image</h2>
  <img src="https://picsum.photos/300" alt="Random Image" width="300" />
  <h2>Audio</h2>
  <audio controls>
    <source src="https://www.soundhelix.com/examples/mp3/SoundHelix-Song-1.mp3" type="audio/mpeg">
  </audio>
  <h2>Video</h2>
  <video width="300" controls>
    <source src="https://www.w3schools.com/html/mov_bbb.mp4" type="video/mp4">
  </video>

</html>
```

And The Result!



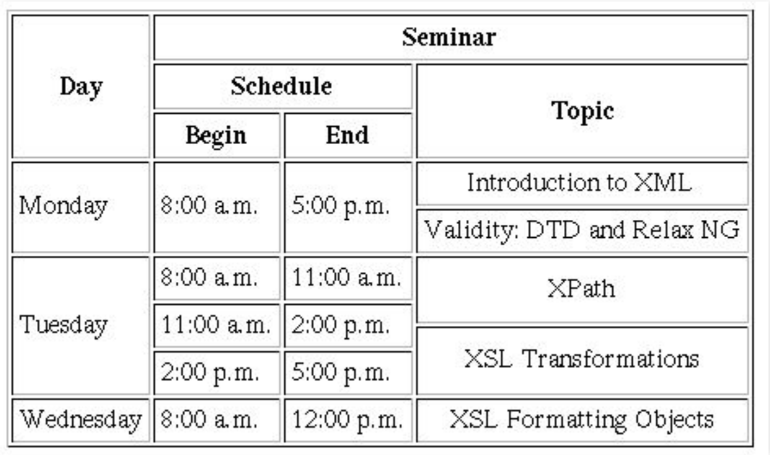
Image

Audio

Video

Tables

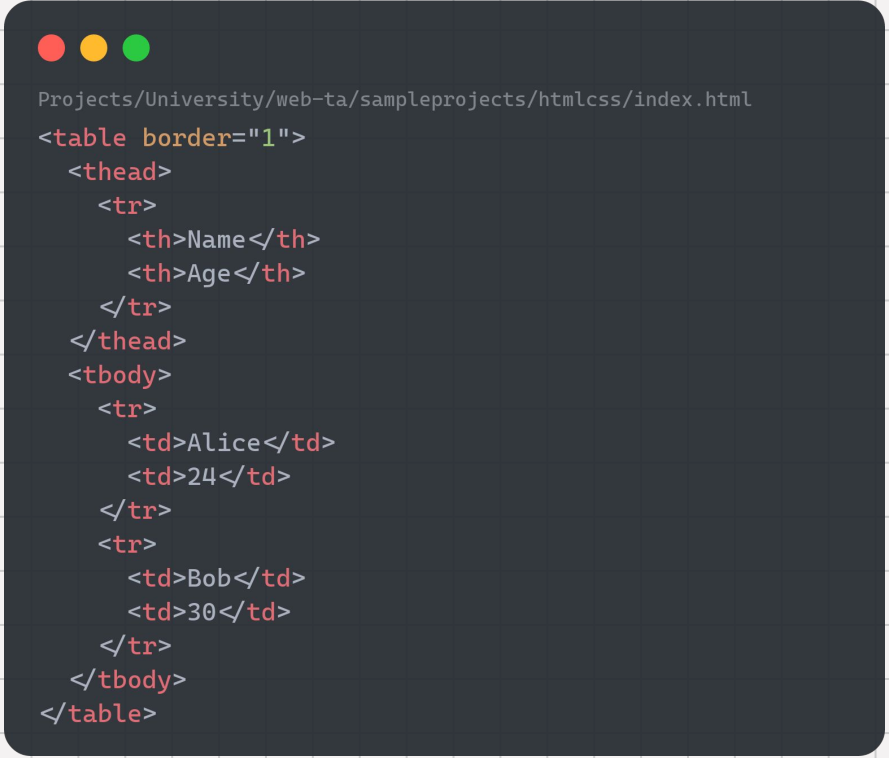
- Tables are used to display structured data in rows and columns.
- Use `<table>` to define the table structure.
- `<tr>` defines a table row, `<td>` a data cell, and `<th>` a header cell.
- Optional tags like `<thead>`, `<tbody>`, and `<tfoot>` help organize **large tables**.



Day	Seminar		
	Schedule		Topic
	Begin	End	
Monday	8:00 a.m.	5:00 p.m.	Introduction to XML
			Validity: DTD and Relax NG
Tuesday	8:00 a.m.	11:00 a.m.	XPath
	11:00 a.m.	2:00 p.m.	XSL Transformations
	2:00 p.m.	5:00 p.m.	
Wednesday	8:00 a.m.	12:00 p.m.	XSL Formatting Objects

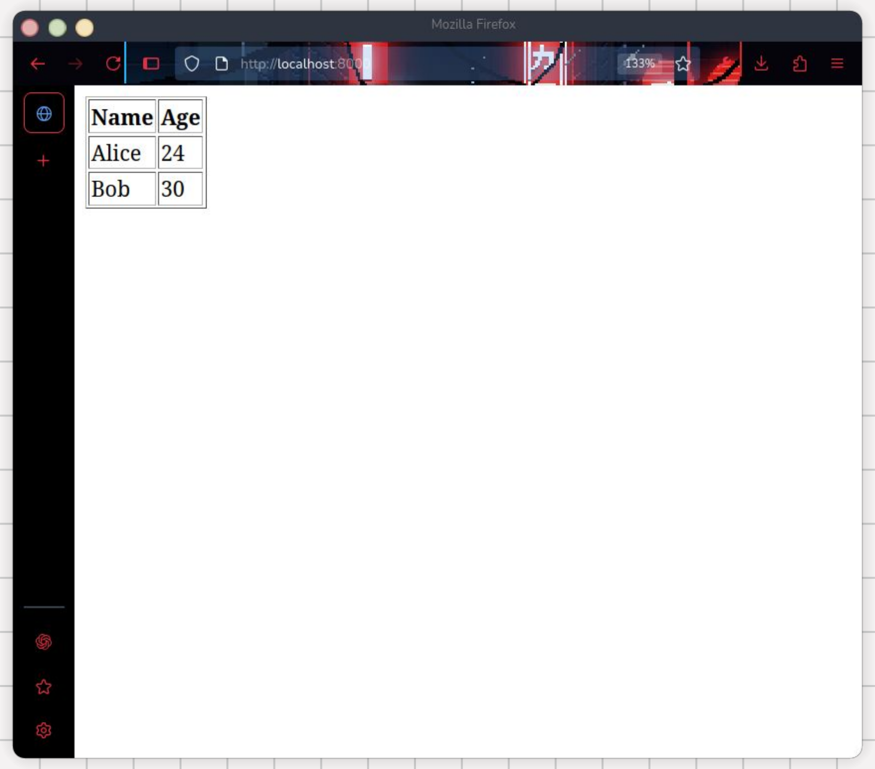
source: stackoverflow

A Sample Table



Projects/University/web-ta/sampleprojects/htmlcss/index.html

```
<table border="1">
  <thead>
    <tr>
      <th>Name</th>
      <th>Age</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Alice</td>
      <td>24</td>
    </tr>
    <tr>
      <td>Bob</td>
      <td>30</td>
    </tr>
  </tbody>
</table>
```



Name	Age
Alice	24
Bob	30

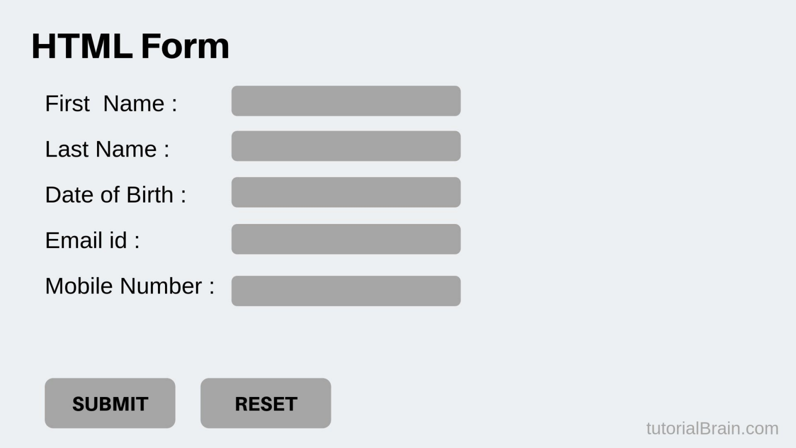
Forms

- Forms collect user input and send it to a server.

- Use the `<form>` tag with attributes like action and method.

- Input elements include `<input>`, `<textarea>`, `<select>`, and `<button>`.

- Use name attributes to identify the data sent to the server.



HTML Form

First Name :

Last Name :

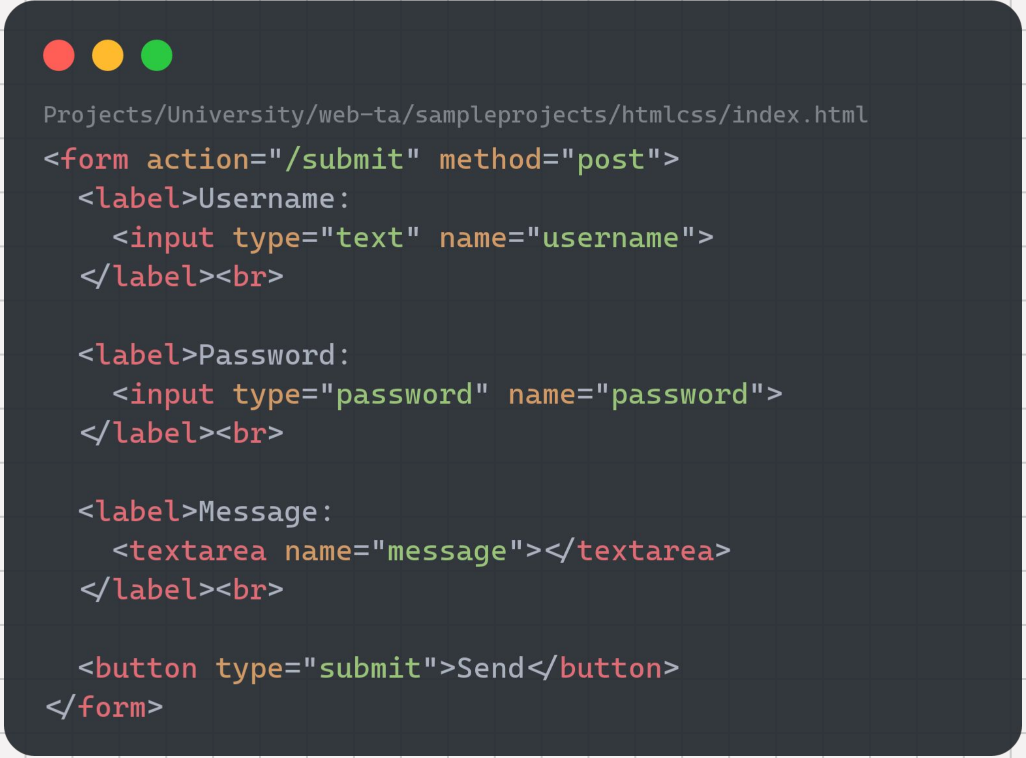
Date of Birth :

Email id :

Mobile Number :

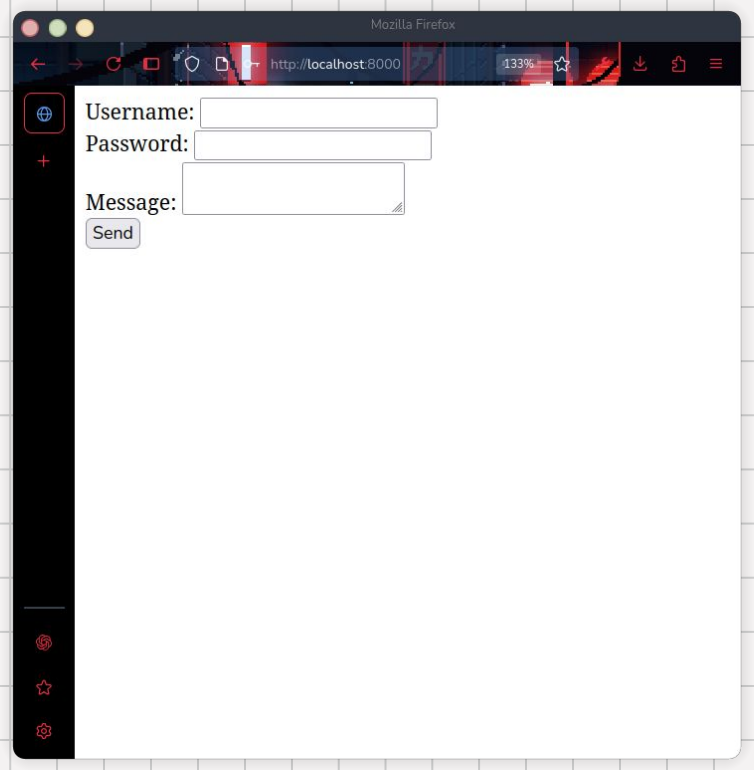
source: TutorialBrain

A Sample Form



Projects/University/web-ta/sampleprojects/htmlcss/index.html

```html
<form action="/submit" method="post">
  <label>Username:
    <input type="text" name="username">
  </label><br>

  <label>Password:
    <input type="password" name="password">
  </label><br>

  <label>Message:
    <textarea name="message"></textarea>
  </label><br>

  <button type="submit">Send</button>
</form>
```



Username:

Password:

Message:

Class Activity Time

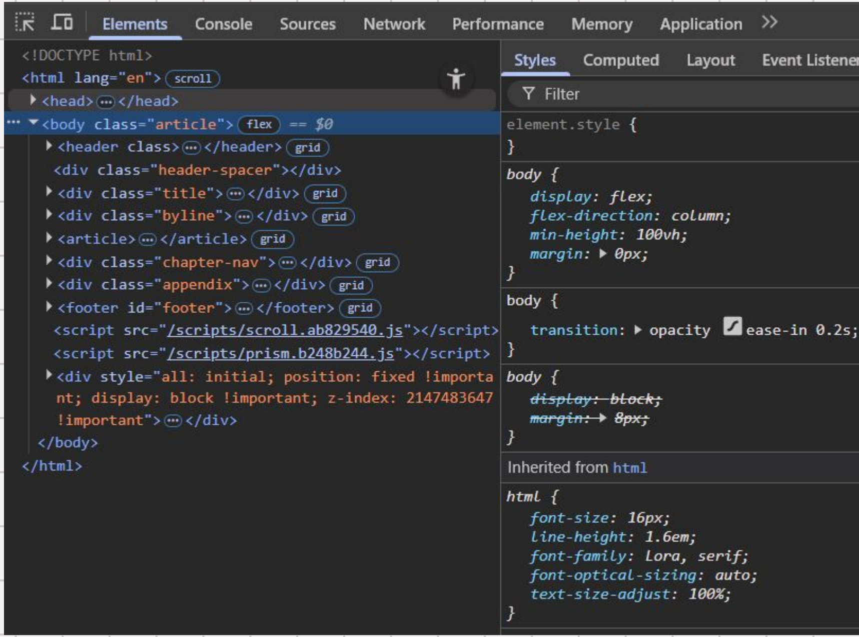
What other elements can we have in an HTML form?

How do you merge two cells in a table?

What other elements are there in HTML?

Inspect Element

- To **analyze** how a web page is structured.
- To **debug** issues related to design, layout, or functionality.
- To **experiment** with code changes without affecting the live website.
- To **learn** front-end web development by exploring real examples.



```
<!DOCTYPE html>
<html lang="en"> scroll
 ▶<head>⋯</head>
⋯ ▼<body class="article"> flex == $0
   ▶<header class>⋯</header> grid
    <div class="header-spacer"></div>
   ▶<div class="title">⋯</div> grid
   ▶<div class="byline">⋯</div> grid
   ▶<article>⋯</article> grid
   ▶<div class="chapter-nav">⋯</div> grid
   ▶<div class="appendix">⋯</div> grid
   ▶<footer id="footer">⋯</footer> grid
    <script src="/scripts/scroll.ab829540.js"></script>
    <script src="/scripts/prism.b248b244.js"></script>
   ▶<div style="all: initial; position: fixed !importa
    nt; display: block !important; z-index: 2147483647
    !important">⋯</div>
  </body>
</html>
```

```
element.style {
}
body {
  display: flex;
  flex-direction: column;
  min-height: 100vh;
  margin: ▶ 0px;
}
body {
  transition: ▶ opacity ease-in 0.2s;
}
body {
  ~~display: block;~~
  ~~margin: ▶ 8px;~~
}
Inherited from html
html {
  font-size: 16px;
  line-height: 1.6em;
  font-family: Lora, serif;
  font-optical-sizing: auto;
  text-size-adjust: 100%;
}
```

Debugging

- It is a painful thing
  To look at your own trouble and know
  That you yourself and no one else has made it
  Sophocles, Ajax

- No one writes perfect software, so it's a given that debugging will take up a major portion of your day.

- Fix the Problem, Not the Blame.

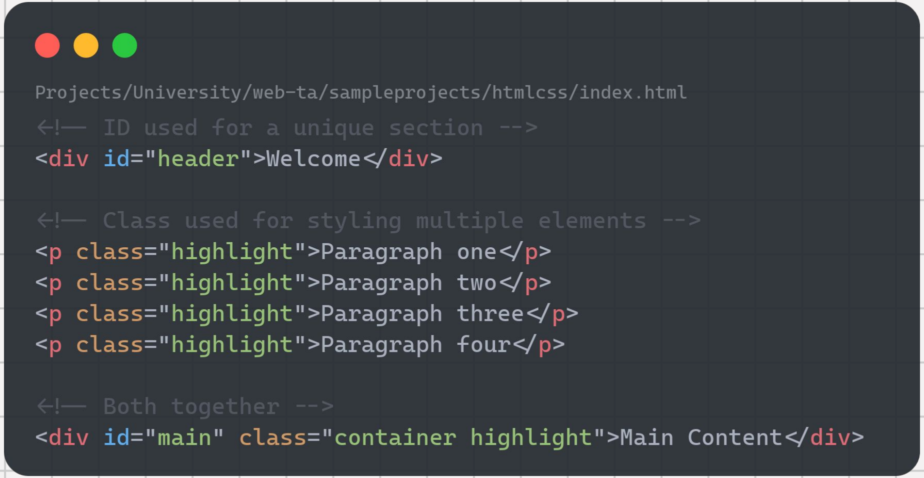
- The first rule of debugging: Don't Panic.

Debugging (cont.)

- Always try to discover the **root cause** of a problem, not just this particular appearance of it.
- A **debugger** that allows you to visualize your data and all of the interrelationships that exist.
- A very simple but particularly useful technique for finding the cause of a problem is simply to explain it to someone else.
- Source: *The Pragmatic Programmer* book

Distinguishing Elements

- `id` is a unique identifier used for a single element.
- `class` is used to group multiple elements with shared styling or behavior.
- An element can have only one `id`, but multiple `class` values.
- `id` is often used for anchors, scripting, or targeting specific elements.
- `class` is used mostly for styling and reusable components.

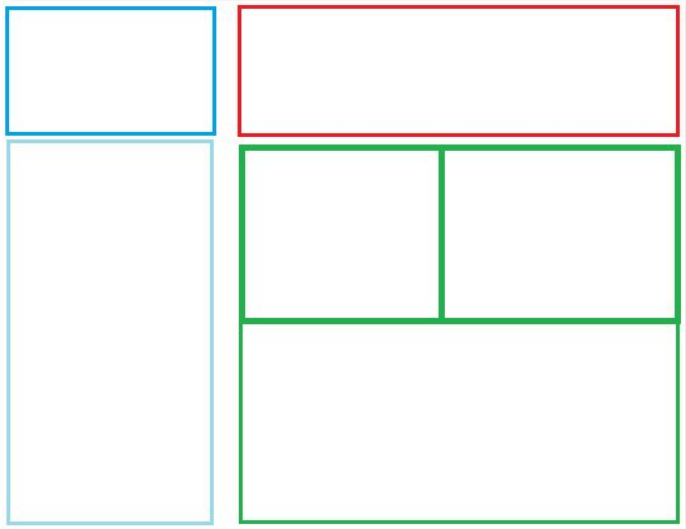


Projects/University/web-ta/sampleprojects/htmlcss/index.html

```html
<!-- ID used for a unique section -->
<div id="header">Welcome</div>

<!-- Class used for styling multiple elements -->
<p class="highlight">Paragraph one</p>
<p class="highlight">Paragraph two</p>
<p class="highlight">Paragraph three</p>
<p class="highlight">Paragraph four</p>

<!-- Both together -->
<div id="main" class="container highlight">Main Content</div>
```

Sections In a Page

- When designing a web page, you can **divide** the content into sections.
- This makes your content more organized and accessible.
- This also improves **accessibility** and **SEO** by clearly structuring your sections.



source: stackoverflow

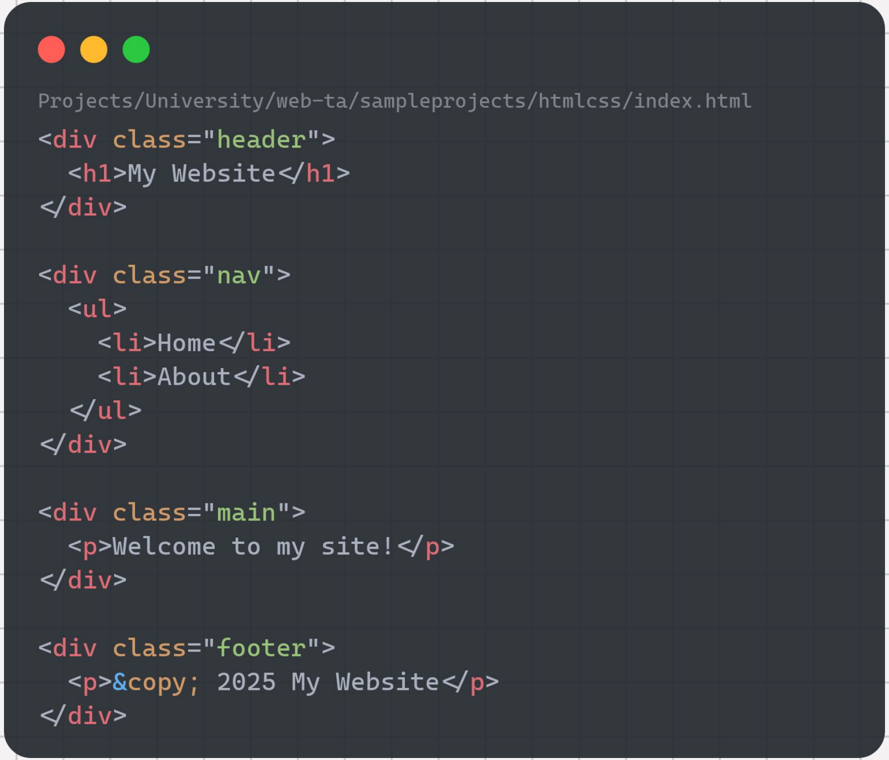
Method I: Use `<div>`

- `<div>` is a generic container used to group HTML elements.
- Commonly used with classes or IDs to apply styles or behavior.
- Does not carry any **semantic meaning** on its own.
- Useful for layout structures like grids, wrappers, and content blocks.

Method II: Use Semantic Tags

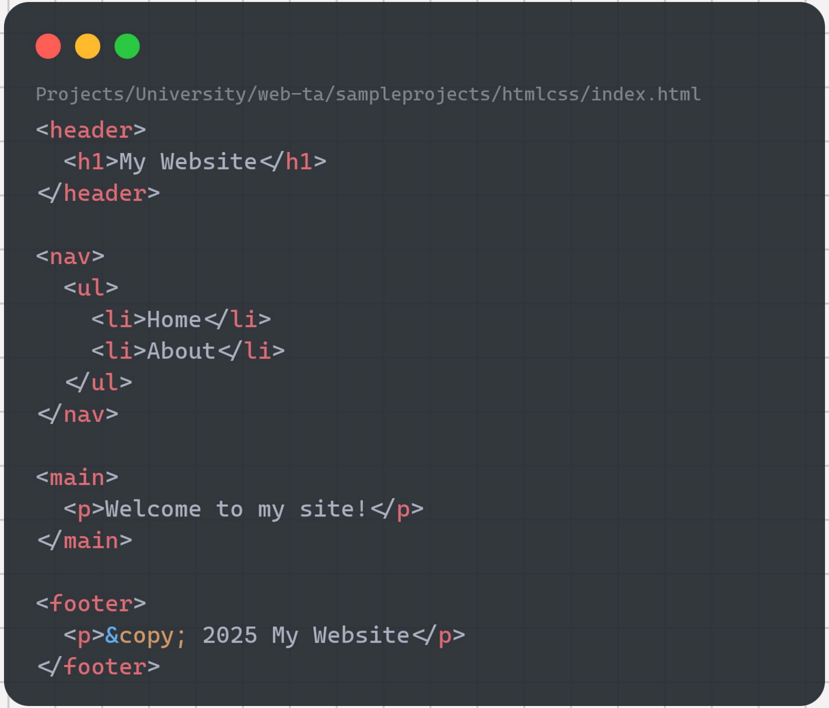
- Semantic tags like `<header>`, `<nav>`, `<main>`, `<section>`, and `<footer>` describe the purpose of content.
- Improve readability of the code for developers.
- Enhance accessibility for screen readers and assistive technologies.
- Help search engines better understand and index your content.

Method I VS. Method II



Projects/University/web-ta/sampleprojects/htmlcss/index.html

```html
<div class="header">
  <h1>My Website</h1>
</div>

<div class="nav">
  <ul>
    <li>Home</li>
    <li>About</li>
  </ul>
</div>

<div class="main">
  <p>Welcome to my site!</p>
</div>

<div class="footer">
  <p>&copy; 2025 My Website</p>
</div>
```

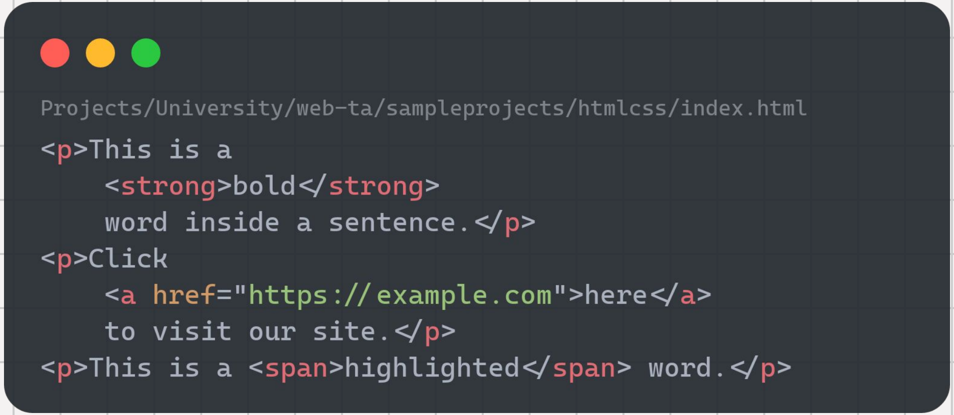


Projects/University/web-ta/sampleprojects/htmlcss/index.html

```html
<header>
  <h1>My Website</h1>
</header>

<nav>
  <ul>
    <li>Home</li>
    <li>About</li>
  </ul>
</nav>

<main>
  <p>Welcome to my site!</p>
</main>

<footer>
  <p>&copy; 2025 My Website</p>
</footer>
```

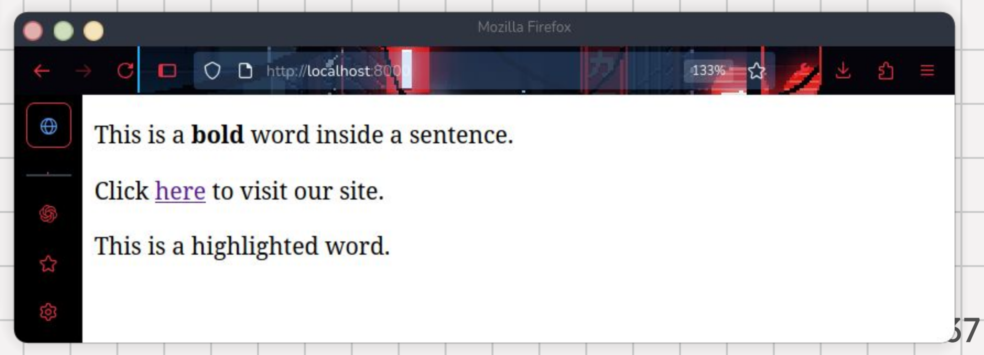
What Does Inline Mean?

- **Inline** elements do not start on a new line and only take up as much width as needed.
- They are used within **block** elements to style or mark up small parts of content.
- Unlike block elements, inline elements cannot contain block-level elements.
- `<span>` is a generic inline container.



Projects/University/web-ta/sampleprojects/htmlcss/index.html

```
<p>This is a
    <strong>bold</strong>
    word inside a sentence.</p>
<p>Click
    <a href="https://example.com">here</a>
    to visit our site.</p>
<p>This is a <span>highlighted</span> word.</p>
```



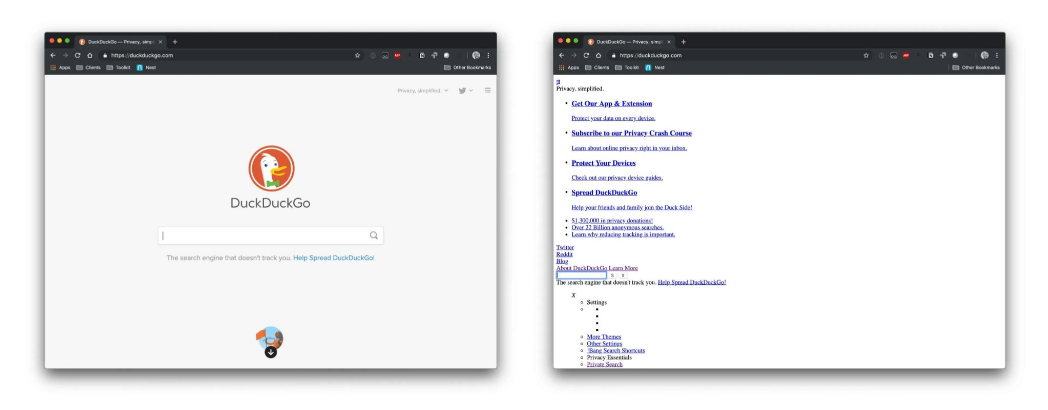
This is a **bold** word inside a sentence.

Click here to visit our site.

This is a highlighted word.

The Problems With Pure HTML

- It’s non-interactive: HTML alone can’t respond to clicks, input validation, or animations.
- It’s ugly: You can’t change colors, fonts, or layout.
- Pure HTML pages are static. Hence, they don’t adapt to users or fetch live data.



The search engine that doesn't track you. Help Spread DuckDuckGo!

Privacy, simplified.

- **Get Our App & Extension**

  Protect your data on every device.
- **Subscribe to our Privacy Crash Course**

  Learn about online privacy right in your inbox.
- **Protect Your Devices**

  Check out our privacy device guides.
- **Spread DuckDuckGo**

  Help your friends and family join the Duck Side!
- $1,300,000 in privacy donations!
- Over 22 Billion anonymous searches.
- Learn why reducing tracking is important.

Twitter
Reddit
Blog
About DuckDuckGo Learn More
S X
The search engine that doesn't track you. Help Spread DuckDuckGo!

X
- Settings
- More Themes
- Other Settings
- !Bang Search Shortcuts
- Privacy Essentials
- Private Search

source: CSSTricks

Make It Interactive and Beautiful!

- Use CSS to style elements with colors, fonts, layouts, and animations.
- Add JavaScript to handle clicks, forms, and dynamic updates.
- Combine HTML, CSS, and JS to build responsive, user-friendly interfaces.
- Use libraries or frameworks (like Bootstrap or React) to speed up development.
- Always keep accessibility and performance in mind for a better user experience.