# Unit 2: Hyper Text Markup Language

Presented By:
Bipin Maharjan

# Learning Objectives

- Understand HTML's role in web development and its key components.
- Comprehend the basic structure of HTML documents, including head and body sections.
- Identify and use tags, elements, and attributes within HTML.
- Apply proper HTML doctype declaration for consistent rendering.
- Utilize meta tags to specify character encoding, viewport, and other metadata.
- Format content using heading, paragraph, strong, em, underline, and strikethrough tags.
- Implement line breaks, and horizontal rules, and create hyperlinks with anchor tags.
- Establish navigation menus, and lists, and organize content effectively.
- Insert images and multimedia elements, setting attributes like src, alt, width, and height.
- Construct tables, including headers, data cells, merging cells, and adding captions.
- Create interactive forms with input elements, validate user input, and handle form submission.

# Table of Contents

- Introduction to HTML
- Document Structure
- Text Formatting
- Links and Navigation
- Hyperlink
- Images and Multimedia
- Lists, Tables, Forms and Input
- Semantic HTML

# Introduction to HTML

# Introduction to HTML

- HTML stands for HyperText Markup Language.
- It is the standard language used to create and structure web pages on the World Wide Web.
    - HyperText: means text that contains links (called hyperlinks) to other pages.
    - Markup Language: means a language that uses tags to describe the structure and presentation of text.
- HTML is not a programming language, its a Markup Language
- A markup language is a set of markup tags.
- HTML uses markup tags to describe web pages.
- HTML is not case sensitive language.
- HTML documents contain HTML tags and plain text.

# Role of HTML in Web Development

HTML provides the skeleton or structure of a webpage.

**Other technologies like:**

- CSS → adds design and styling (colors, fonts, layouts).
- JavaScript → adds interactivity and behavior.

**Example analogy:**

- HTML = bones (structure)
- CSS = skin & clothes (appearance)
- JavaScript = muscles (movement/interactivity)

# Basic Components of HTML

HTML is made up of tags, elements, and attributes.

1. **Tags**

Tags are the building blocks of HTML.

They are keywords enclosed in angle brackets < >.

**Example:**

```
<p>This is a paragraph.</p>
```

Here:

- <p> → opening tag
- </p> → closing tag

Everything between them is the content of the element.

# 2. Elements

An element includes:

- Opening tag
- Content
- Closing tag

**Example:** <h1>Welcome to HTML</h1>

This is a heading element.

Some elements are empty or self-closing, meaning they do not have closing tags.

```
<br>   <!-- line break →

<hr>   <!-- horizontal line →

<img src="logo.png" alt="Logo">
```

# 3. Attributes

- Attributes give extra information about an element.
- They are written inside the opening tag as name-value pairs.

Attributes give extra information about an element.

They are written inside the opening tag as name-value pairs.

**Syntax:** <tagname attribute="value">Content</tagname>

**Example:** <img src="photo.jpg" alt="A beautiful scenery" width="400" height="300">

**Here:**

- src → source of the image
- alt → alternate text
- width / height → image dimensions

# Structure of an HTML Document

A basic HTML file looks like this:

```
<!DOCTYPE html>
<html>
<head>
   <title>My First Webpage</title>
</head>
<body>
   <h1>Welcome to HTML!</h1>
   <p>This is my first webpage.</p>
</body>
</html>
```

# Structure of an HTML Document

- <!DOCTYPE html> → Declares document type (HTML5).
- <html> → Root element; everything goes inside it.
- <head> → Contains information about the webpage (metadata, title, links to CSS, etc.).
- <body> → Contains the visible content shown in the browser.

# Document Structure

# <!DOCTYPE html> - Doctype Declaration

- Appears at the top of every HTML document.
- Tells the browser which version of HTML is being used.
- In HTML5, it is written as:

```
<!DOCTYPE html>
```

This ensures the browser renders the page in standard mode (not in older “quirks mode”).

# <html> - Root Element

- Encloses all the content of the page.
- It has two main parts:
  - <head> — contains meta information.
  - <body> — contains visible content.
- Example:

```
<html>
   …
</html>
```

Note: You can add a lang attribute to specify the page language.

```
<html lang="en">
```

# <head> - Head Section

- This section contains information about the page, not the content that appears on the web page itself.
- It includes:
  - Title of the page
  - Metadata (author, description, keywords)
  - Links to stylesheets and scripts
  - Character encoding

```
<head>
   <title>My Website</title>
   <meta charset="UTF-8">
   <meta name="description" content="Learn HTML basics">
   <meta name="author" content="John Doe">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
```

# <body> - Body Section

- This section contains all the visible content that appears on the webpage, such as:
  - Headings
  - Paragraphs
  - Images
  - Links
  - Lists, Tables, Forms, etc.

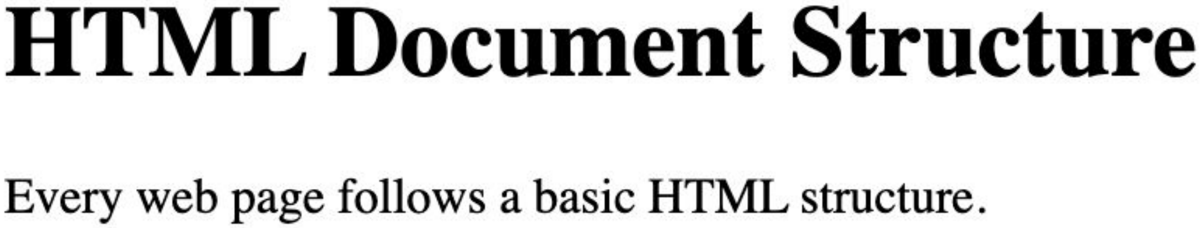
**Example:**

```
<body>
  <h1>Welcome to My Page</h1>
  <p>This is an example of body content.</p>
  <img src="photo.jpg" alt="Sample image">
</body>
```

Everything between <body> and </body> is displayed in the browser window.

# Putting It All Together

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document Structure Example</title>
</head>
<body>
    <h1>HTML Document Structure</h1>
    <p>Every web page follows a basic HTML structure.</p>
</body>
</html>
```



# HTML Document Structure

Every web page follows a basic HTML structure.

# Text Formatting in HTML

# Introduction

- HTML provides several tags that allow us to format text and make it more readable and meaningful.
- These formatting tags help structure content by defining headings, paragraphs, emphasis, highlights, and other text styles.

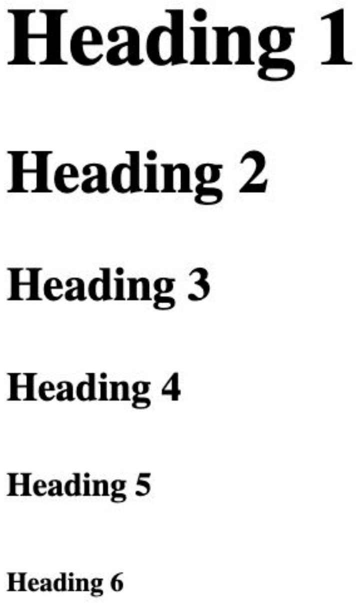
# Headings

HTML provides six levels of headings, from <h1> (most important) to <h6> (least important).

**Example:**

```
<h1>Heading 1</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
<h4>Heading 4</h4>
<h5>Heading 5</h5>
<h6>Heading 6</h6>
```

Use headings hierarchically. For example, <h1> should be the main title, and <h2> or <h3> for sub-sections.



Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

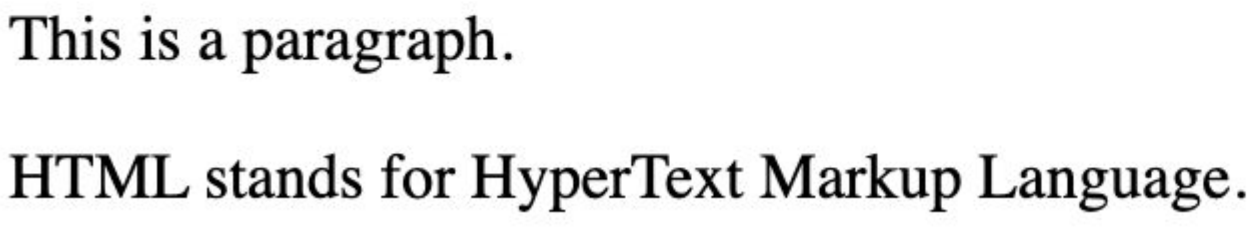
Heading 6

# Paragraphs

The <p> tag is used to write blocks of text.

**Example:**

```
<p>This is a paragraph.</p>

<p>HTML stands for HyperText Markup Language.</p>
```



This is a paragraph.

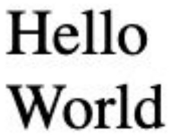
HTML stands for HyperText Markup Language.

Browsers automatically add line breaks before and after paragraphs.

# Line Break and Horizontal Rule

- Line Break (<br>) → Moves text to the next line.

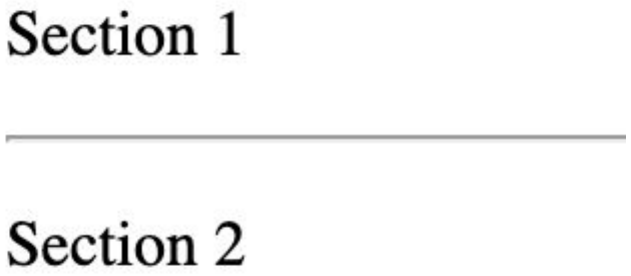
```
Hello<br>World
```



Hello
World

- Horizontal Rule (<hr>) → Creates a horizontal line to separate content.

```
<p>Section 1</p>

<hr>

<p>Section 2</p>
```



Section 1

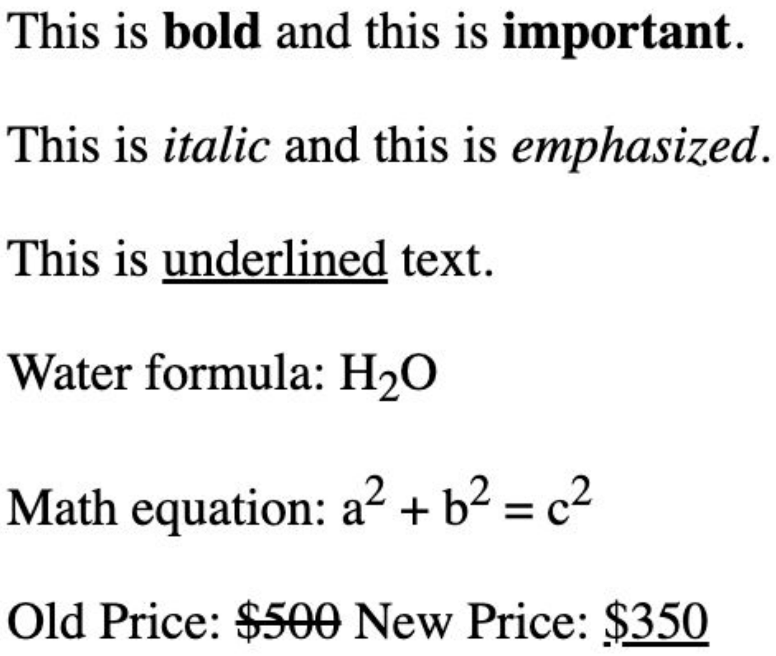
Section 2

# Text Styling Tags

| Tag | Purpose | Example |
|---|---|---|
| `<b>` | Bold text | `<b>Important</b>` |
| `<strong>` | Important text (semantic) | `<strong>Warning!</strong>` |
| `<i>` | Italic text | `<i>Italic style</i>` |
| `<em>` | Emphasized text (semantic) | `<em>Pay attention!</em>` |
| `<u>` | Underlined text | `<u>Underlined</u>` |
| `<mark>` | Highlight text | `<mark>Highlighted</mark>` |
| `<small>` | Small text | `<small>Note</small>` |
| `<sub>` | Subscript | `H<sub>2</sub>O` |
| `<sup>` | Superscript | `2<sup>nd</sup>` |
| `<del>` | Deleted text | `<del>Old Price</del>` |
| `<ins>` | Inserted text | `<ins>New Price</ins>` |

# Text Styling Tags Example

```
<p>This is <b>bold</b> and this is
<strong>important</strong>.</p>

<p>This is <i>italic</i> and this is
<em>emphasized</em>.</p>

<p>This is <u>underlined</u> text.</p>

<p>Water formula: H<sub>2</sub>O</p>

<p>Math equation: a<sup>2</sup> +
b<sup>2</sup> = c<sup>2</sup></p>

<p>Old Price: <del>$500</del> New Price:
<ins>$350</ins></p>
```



This is **bold** and this is **important**.

This is *italic* and this is *emphasized*.

This is underlined text.

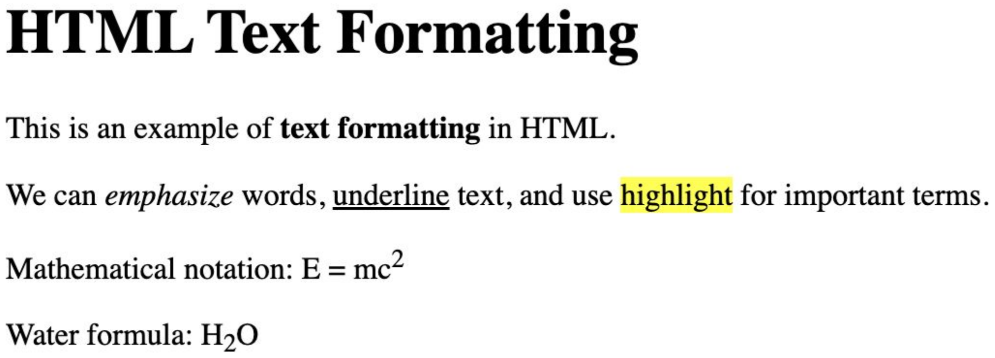
Water formula: $H_2O$

Math equation: $a^2 + b^2 = c^2$

Old Price: ~~$500~~ New Price: $350

# Text Formatting Example

```
<!DOCTYPE html>
<html>
<head>
  <title>Text Formatting Example</title>
</head>
<body>
  <h1>HTML Text Formatting</h1>
  <p>This is an example of <strong>text formatting</strong> in HTML.</p>
  <p>We can <em>emphasize</em> words, <u>underline</u> text, and use <mark>highlight</mark> for important terms.</p>
  <p>Mathematical notation: E = mc<sup>2</sup></p>
  <p>Water formula: H<sub>2</sub>O</p>
</body>
</html>
```



## HTML Text Formatting

This is an example of **text formatting** in HTML.

We can *emphasize* words, underline text, and use highlight for important terms.

Mathematical notation: $E = mc^2$

Water formula: $H_2O$

# Links and Navigation

# Links

- One of the most powerful features of the web is the ability to link from one page to another.
- This is done using hyperlinks, which allow users to navigate between:
  - Different pages of the same website, or
  - Completely different websites.
- In HTML, links are created using the anchor tag (<a>).

# The Anchor Tag

**Syntax:**

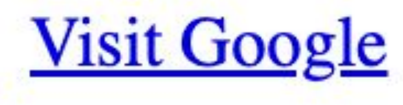
```
<a href="URL">Link Text</a>
```

- <a> → stands for anchor.
- href → stands for hyperlink reference; it defines the destination of the link.
- Link Text → The clickable text shown to users.

**Example:**

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

When you click Visit Google, it opens Google’s website.



Visit Google

# Navigation Menu

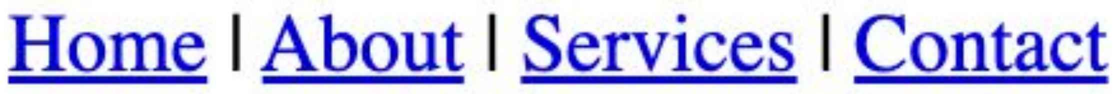
Navigation menus help users move around a website.

They are often created using multiple <a> tags within a <nav> container.

**Example:**

```
<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>
```

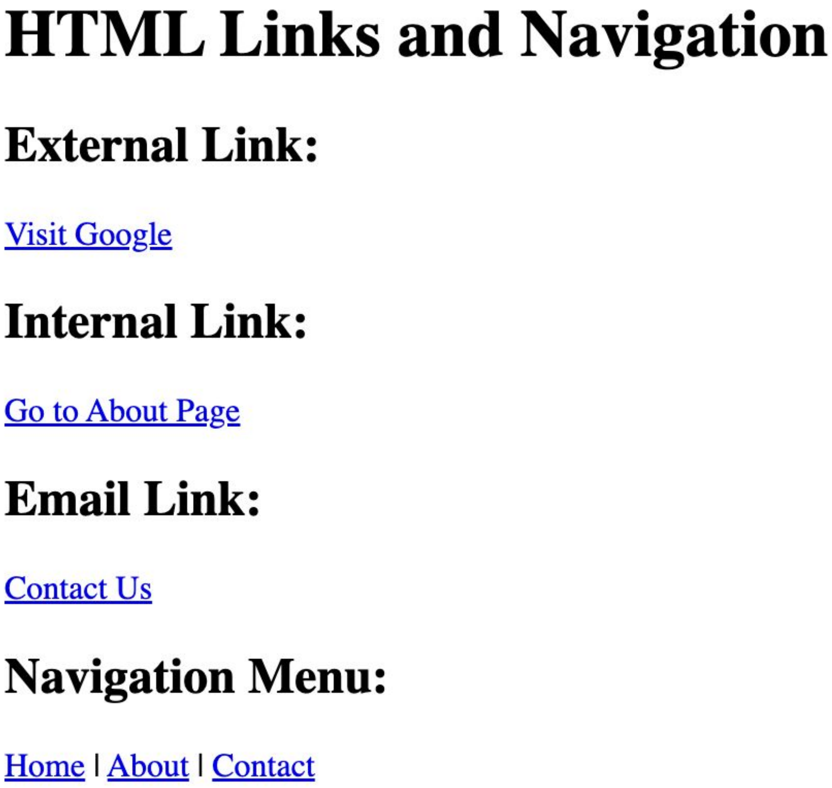
**Output:**



Home | About | Services | Contact

# Links and Navigation Example

```
<!DOCTYPE html>
<html>
<head>
   <title>Links and Navigation Example</title>
</head>
<body>
   <h1>HTML Links and Navigation</h1>

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

   <h2>Internal Link:</h2>
   <a href="about.html">Go to About Page</a>

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

   <h2>Navigation Menu:</h2>
   <nav>
      <a href="index.html">Home</a> |
      <a href="about.html">About</a> |
      <a href="contact.html">Contact</a>
   </nav>
</body>
</html>
```



# HTML Links and Navigation

## External Link:

Visit Google

## Internal Link:

Go to About Page

## Email Link:

Contact Us

## Navigation Menu:

Home | About | Contact

# Hyperlink

# What is a Hyperlink?

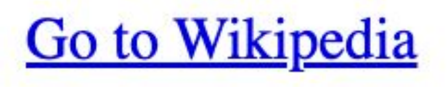
- A hyperlink (or simply link) is a text, image, or any clickable element on a webpage that connects to another webpage, section, file, or resource.
- Hyperlinks are the foundation of the World Wide Web (WWW) — they connect billions of web pages together, creating the web’s “network” structure.
- HTML links are hyperlinks.
- You can click on a link and jump to another document.
- When you move the mouse over a link, the mouse arrow will turn into a little hand.
- A link does not have to be text. A link can be an image or any other HTML element.
- **Syntax:** <a href="URL">Link Text</a>

# Types of Links

1. **Absolute Links**

These link to an external website using the full URL.

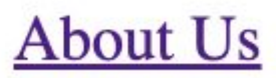
```
<a href="https://www.wikipedia.org">Go to Wikipedia</a>
```



2. **Relative Links**

These link to pages within the same website or project folder.

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



The browser looks for about.html in the same directory as the current page.

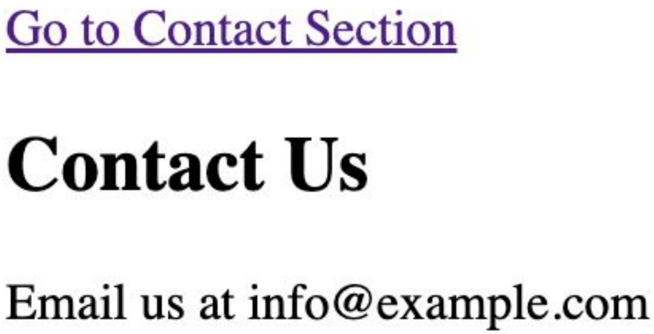
# 3. Anchor Links (Internal Page Links)

- Used to jump to a specific section within the same page.

**Example:**

```
<a href="#contact">Go to Contact Section</a>

<h2 id="contact">Contact Us</h2>

<p>Email us at info@example.com</p>
```

- Here, the link scrolls directly to the element with the id="contact".



Go to Contact Section

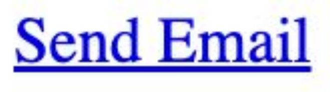
## Contact Us

Email us at info@example.com

## 4. Email Links

- To open the user’s default email application and start composing an email:

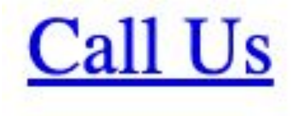
```
<a href="mailto:info@example.com">Send Email</a>
```



## 5. Telephone Links (Mobile Friendly)

- On mobile devices, this allows users to call directly.

```
<a href="tel:+9779812345678">Call Us</a>
```



# Anchor Tag Attributes

| Attribute | Description | Example |
|---|---|---|
| `href` | URL of the page/file | `<a href="about.html">About</a>` |
| `target` | Defines where to open the link | `<a href="page.html" target="_blank">Open in new tab</a>` |
| `title` | Tooltip text (appears on hover) | `<a href="home.html" title="Go Home">Home</a>` |
| `download` | Prompts file download | `<a href="notes.pdf" download>Download Notes</a>` |

# Target Attribute Values

| Value | Description |
|---|---|
| `_self` | Opens in the same tab (default) |
| `_blank` | Opens in a new tab |
| `_parent` | Opens in the parent frame |
| `_top` | Opens in the full browser window |

# Hyperlink on an Image

You can also make an image clickable by placing it inside an <a> tag.

**Example:**

```
<a href="https://www.google.com">
    <img src="logo.png" alt="Google Logo" width="400">
</a>
```

When the image is clicked, it opens the specified URL.

# Images and Multimedia

# Introduction

- Web pages aren’t just about text — they also include images, audio, and video to make content more engaging and interactive.
- HTML provides several tags to embed and control multimedia content, making web pages visually appealing and user-friendly.

# Inserting Images in HTML

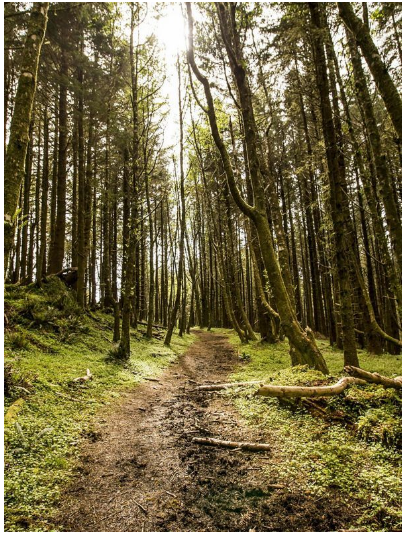
- Images are inserted in HTML using the <img> tag.
- Syntax:

```
<img src="image_url" alt="description">
```

**Explanation:**

- <img> → Used to embed an image.
- src → (Source) Specifies the image file path or URL.
- alt → (Alternate text) Describes the image; displayed if the image fails to load or for screen readers (important for accessibility).

**Example:** <img src="nature.jpg" alt="Beautiful Nature">



# Common Attributes of <img> Tag

| Attribute | Description | Example |
|---|---|---|
| `src` | Specifies image path or URL | `<img src="flower.png">` |
| `alt` | Alternative text if image can’t load | `<img alt="Rose flower">` |
| `width` | Defines image width | `<img width="200">` |
| `height` | Defines image height | `<img height="150">` |
| `title` | Tooltip shown when hovered | `<img title="Click to view">` |
| `loading` | Improves performance by lazy-loading images | `<img loading="lazy">` |

<img src="student.jpg" alt="Student working on computer" width="300" height="200" title="HTML Student">

# Image File Formats Commonly Used

| Format | Description | Example Use |
|---|---|---|
| **.jpg / .jpeg** | Best for photographs | `<img src="photo.jpg">` |
| **.png** | Supports transparency | `<img src="logo.png">` |
| **.gif** | Supports animation | `<img src="animation.gif">` |
| **.svg** | Scalable vector graphics, used for logos/icons | `<img src="icon.svg">` |
| **.webp** | Modern format with good compression | `<img src="image.webp">` |

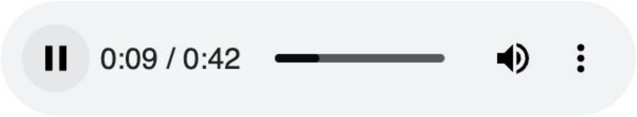
# Embedding Audio

- Modern HTML (HTML5) allows direct embedding of audio without plugins.
- The <audio> tag is used to play sound files like MP3 or WAV.

**Syntax:**

```
<audio controls>
  <source src="sound.mp3" type="audio/mpeg">
</audio>
```

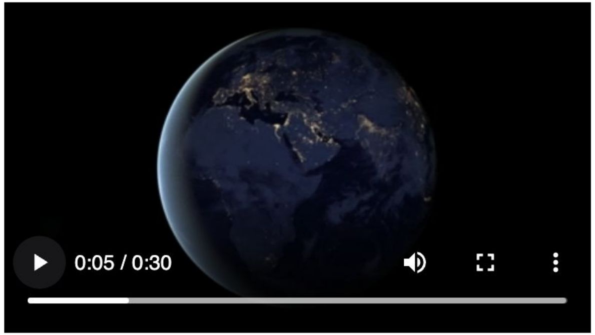
**Explanation:**



- controls → Displays play/pause buttons.
- <source> → Specifies the audio file and its type.
- Fallback text appears if the browser doesn’t support audio.

# Embedding Video

- Modern HTML (HTML5) allows direct embedding of video without plugins.
- The <video> tag is used for embedding videos directly into a webpage.



**Syntax:**

```
<video width="400" controls>
  <source src="movie.mp4" type="video/mp4">
</video>
```

**Explanation:**

- controls → Adds playback controls (play, pause, volume).
- width and height → Define video size.
- <source> → Specifies the video file and format.
- Fallback text ensures accessibility.

# Common Attributes for Audio/Video

| **Attribute** | **Description** |
|---|---|
| `controls` | Adds play/pause buttons |
| `autoplay` | Starts playing automatically *(use carefully)* |
| `loop` | Repeats media automatically |
| `muted` | Starts media with no sound |
| `poster` (video) | Displays an image before playback |
| `preload` | Defines how media loads (auto / metadata / none) |

# Embedding External Multimedia (YouTube Example)

- You can also embed videos from platforms like YouTube using the <iframe> tag.

```
<iframe width="560" height="315"

src="https://www.youtube.com/embed/tgbNymZ7vqY"

title="YouTube video" frameborder="0"
allowfullscreen></iframe>
```

# Lists, Tables, Forms, and Input

# Lists

- HTML lists allow web developers to group a set of related items in lists.
- Lists help display related items in a structured way.
- There are three main types of lists in HTML.
  - Ordered List
  - Unordered List
  - Description List

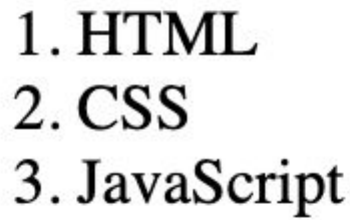
# Ordered List (<ol>)

- An ordered list is a collection of related items that have special order or sequence.
- An ordered list created using the <ol> element, and each list item starts with the <li> element.
- Each item in an ordered list is typically marked with numbers or letters.

**Syntax:**

```
<ol>
  <li>HTML</li>
  <li>CSS</li>
  <li>JavaScript</li>
</ol>
```

**Output:**



1. HTML
2. CSS
3. JavaScript

# Ordered List Attributes

| Attribute | Description | Example |
|---|---|---|
| `type` | Changes number style (`1`, `A`, `a`, `I`, `i`) | `<ol type="A">` |
| `start` | Specifies starting number | `<ol start="5">` |

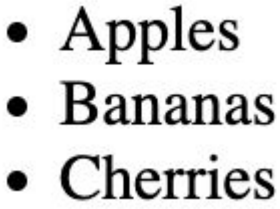
# Unordered List (<ul>)

- An unordered list is a collection of related items that have no special order or sequence.
- An unordered list created using the <ul> element, and each list starts with the <li> element.

**Syntax:**

```
<ul>

  <li>Apples</li>

  <li>Bananas</li>

  <li>Cherries</li>

</ul>
```

**Output:**



```
• Apples
• Bananas
• Cherries
```

# Unordered List Attributes

| Attribute | Description | Example |
|---|---|---|
| `type` | Bullet style (`disc`, `circle`, `square`) | `<ul type="square">` |

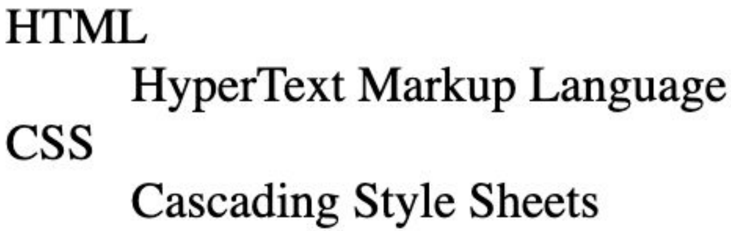
# Description List (<dl>)

- A description list is a list of items with a description or definition of each item.
- The description list is created using <dl> element.
- The <dl> element is used in conjunction with the <dt> element which specify a term, and the <dd> element which specify the term’s definition.

**Syntax:**

```
<dl>
  <dt>HTML</dt>
  <dd>HyperText Markup Language</dd>
  <dt>CSS</dt>
  <dd>Cascading Style Sheets</dd>
</dl>
```

**Output:**



```
HTML
        HyperText Markup Language
CSS
        Cascading Style Sheets
```

# Tables

- Table is combination of rows and columns, which intersect each other and create cell where user store the value.
- Tables can contain:
  - Text
  - Additional markup
  - Other tables
  - Images
  - And just about anything else
- Tables are block level elements

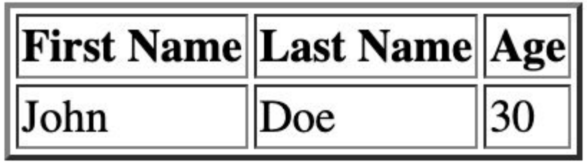
# Basic Table Tags

- The <table> tag is the outermost tag and marks the table
  - The border attribute defines the width of the border surrounding the table
- The <tr> tag appears inside the <table> tag and marks a table row
- The <td> tag appears inside the <tr> tag and marks the table data (cell)
- Example: Table with 2-pixel border

# Table Example

```
<table border="2">
    <tr>
        <th>First Name</th>
        <th>Last Name</th>
        <th>Age</th>
    </tr>
    <tr>
        <td>John</td>
        <td>Doe</td>
        <td>30</td>
    </tr>
 </table>
```

**Output:**



| First Name | Last Name | Age |
|---|---|---|
| John | Doe | 30 |

<table> – Defines the table

<tr> – Table row

<th> – Table header (bold and centered by default)

<td> – Table data cell

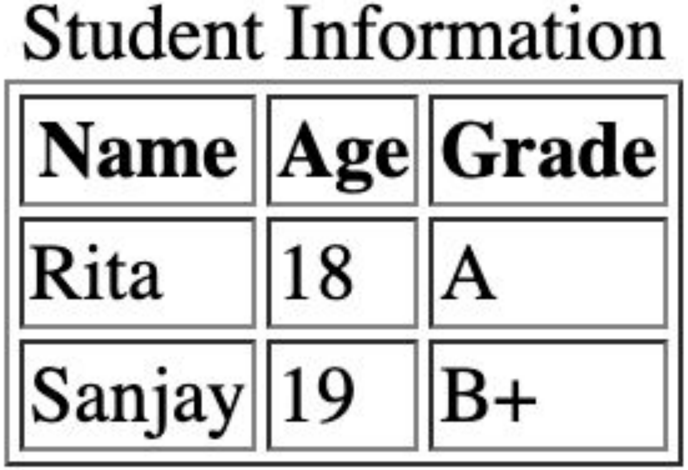
# Additional Table Elements

| Tag | Description |
|---|---|
| `<caption>` | Adds a title to the table |
| `<thead>` | Groups the header content |
| `<tbody>` | Groups body content |
| `<tfoot>` | Groups footer content |
| `<colspan>` | Merges cells horizontally |
| `<rowspan>` | Merges cells vertically |

# Tables with Captions

The <caption> element appears as a child of a <table> and contains the table’s title. The caption appears just above the table itself.

```
<table border="1">
    <caption>Student Information</caption>
    <tr>
        <th>Name</th>
        <th>Age</th>
        <th>Grade</th>
    </tr>
    <tr>
        <td>Rita</td>
        <td>18</td>
        <td>A</td>
    </tr>
    <tr>
        <td>Sanjay</td>
        <td>19</td>
        <td>B+</td>
    </tr>
</table>
```



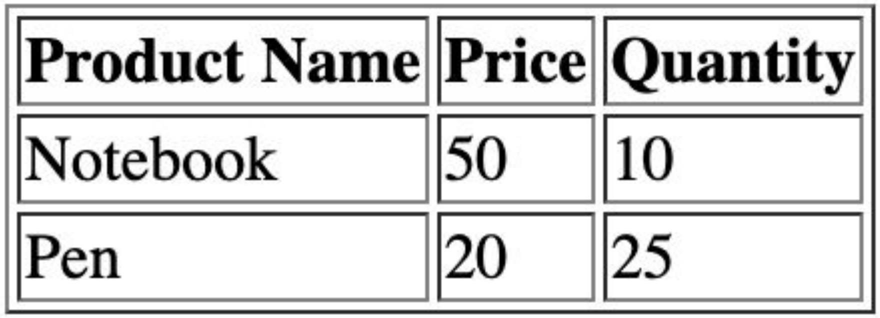
Student Information

| Name | Age | Grade |
|---|---|---|
| Rita | 18 | A |
| Sanjay | 19 | B+ |

# Table with Headers

The <th> element is similar to the <tr> element but contains the table’s header. The header is typically emphasized.

```
<table border="1">
    <caption>Product List</caption>

    <tr>
        <th>Product Name</th>
        <th>Price</th>
        <th>Quantity</th>
    </tr>

    <tr>
        <td>Notebook</td>
        <td>50</td>
        <td>10</td>
    </tr>

    <tr>
        <td>Pen</td>
        <td>20</td>
        <td>25</td>
    </tr>
</table>
```

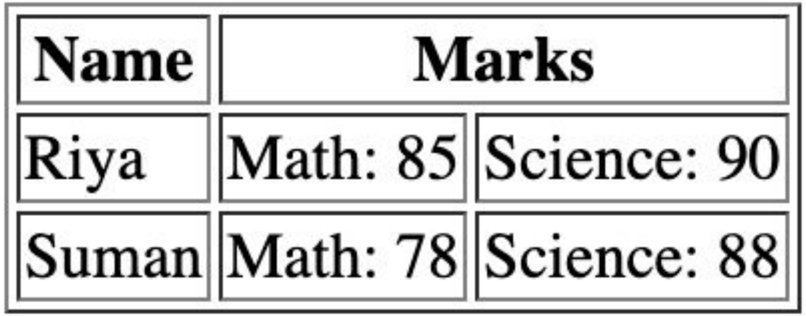


| Product Name | Price | Quantity |
|---|---|---|
| Notebook | 50 | 10 |
| Pen | 20 | 25 |

# Table with colspan

colspan="2" merges two columns into one.

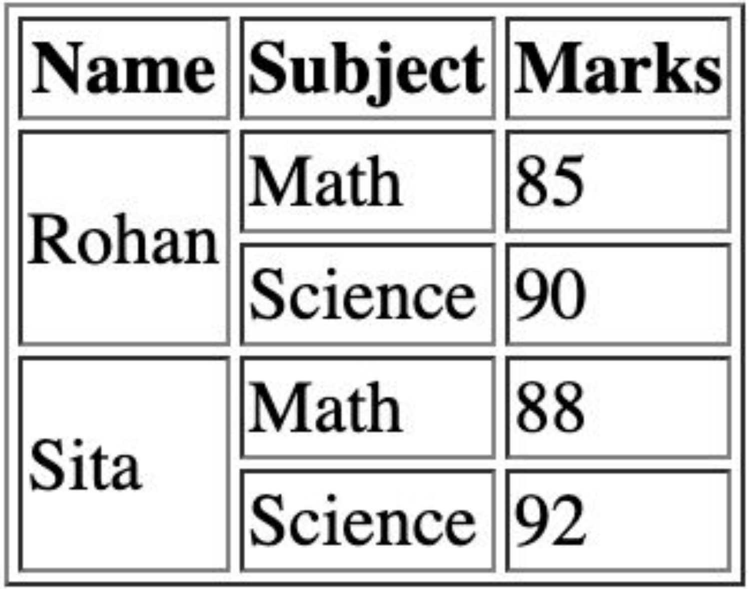
```
<table border="1">
    <caption>Student Marks</caption>

    <tr>
        <th>Name</th>
        <th colspan="2">Marks</th>
    </tr>

    <tr>
        <td>Riya</td>
        <td>Math: 85</td>
        <td>Science: 90</td>
    </tr>

    <tr>
        <td>Suman</td>
        <td>Math: 78</td>
        <td>Science: 88</td>
    </tr>
</table>
```



| Name | Marks | |
|---|---|---|
| Riya | Math: 85 | Science: 90 |
| Suman | Math: 78 | Science: 88 |

# Table with rowspan

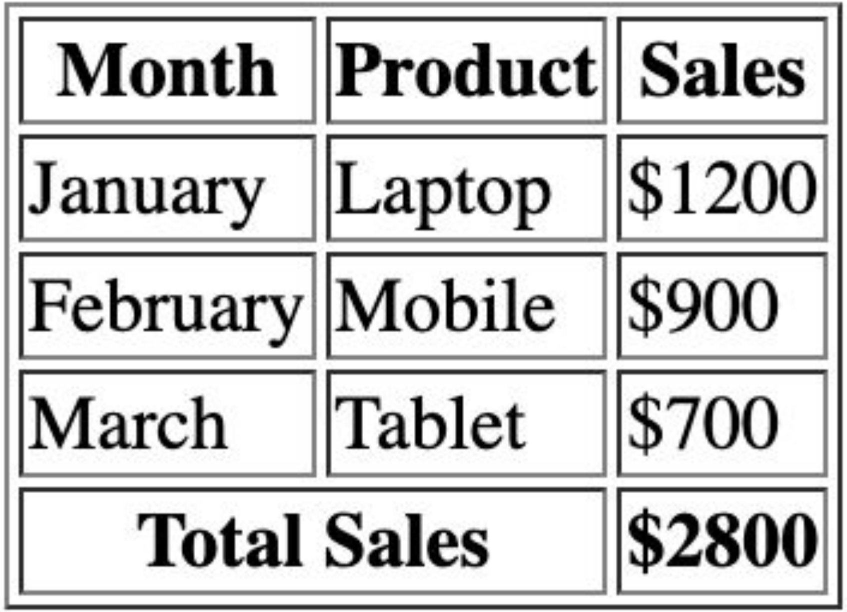
```
<table border="1">
  <tr>
    <th>Name</th>
    <th>Subject</th>
    <th>Marks</th>
  </tr>

  <tr>
    <td rowspan="2">Rohan</td>
    <td>Math</td>
    <td>85</td>
  </tr>

  <tr>
    <td>Science</td>
    <td>90</td>
  </tr>

  <tr>
    <td rowspan="2">Sita</td>
    <td>Math</td>
    <td>88</td>
  </tr>

  <tr>
    <td>Science</td>
    <td>92</td>
  </tr>
</table>
```



| Name | Subject | Marks |
|---|---|---|
| Rohan | Math | 85 |
| | Science | 90 |
| Sita | Math | 88 |
| | Science | 92 |

# Table with groups

```
<table border="1">
    <thead>
    <tr>
        <th>Month</th>
        <th>Product</th>
        <th>Sales</th>
    </tr>
    </thead>

    <tbody>
    <tr>
        <td>January</td>
        <td>Laptop</td>
        <td>$1200</td>
    </tr>
    <tr>
        <td>February</td>
        <td>Mobile</td>
        <td>$900</td>
    </tr>
    <tr>
        <td>March</td>
        <td>Tablet</td>
        <td>$700</td>
    </tr>
    </tbody>

    <tfoot>
    <tr>
        <th colspan="2">Total Sales</th>
        <th>$2800</th>
    </tr>
    </tfoot>
</table>
```



| Month | Product | Sales |
|---|---|---|
| January | Laptop | $1200 |
| February | Mobile | $900 |
| March | Tablet | $700 |
| **Total Sales** | | **$2800** |

# Forms

- Forms are used to collect user input (like name, email, password, etc.) and send it to a server.
- Syntax:

```
<form action="submit_form.php" method="post">
  <!-- form elements here -->
</form>
```

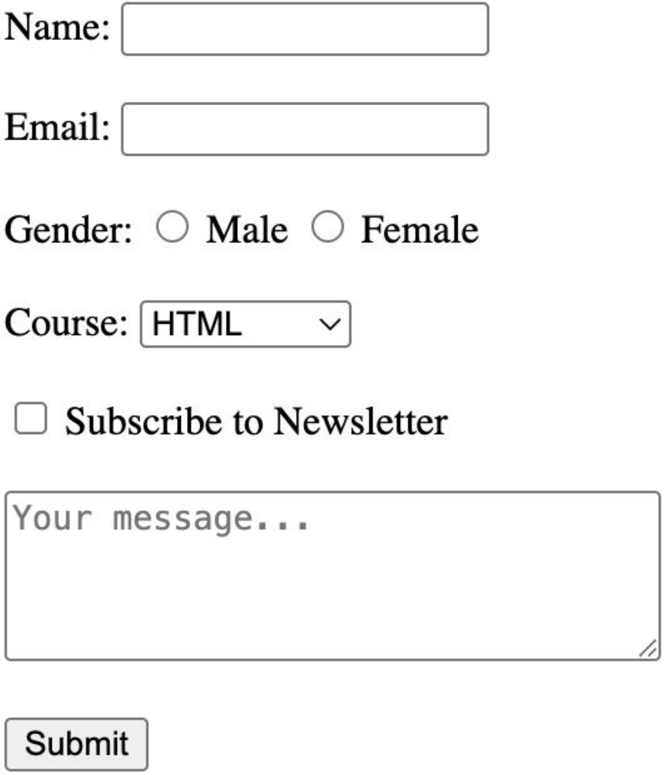
- <form> → Defines the form area.
- action → URL where form data will be sent.
- method → “GET” (visible in URL) or “POST” (hidden, more secure).

# Form Elements

| Tag | Description | Example |
|---|---|---|
| `<input>` | Takes user input | `<input type="text">` |
| `<label>` | Describes an input field | `<label for="name">Name:</label>` |
| `<textarea>` | Multiline input area | `<textarea></textarea>` |
| `<select>` | Dropdown menu | `<select><option></option></select>` |
| `<button>` | Clickable button | `<button>Submit</button>` |

# Form Example

```
<form action="" method="post">
  Name:<input type="text" id="name" name="username"
required><br><br>

  Email: <input type="email" id="email" name="email"
required><br><br>

  Gender: <input type="radio" name="gender" value="Male"> Male
  <input type="radio" name="gender" value="Female">
Female<br><br>

  Course:
  <select id="course" name="course">
    <option>HTML</option>
    <option>CSS</option>
    <option>JavaScript</option>
  </select><br><br>

  <input type="checkbox" name="subscribe" checked> Subscribe to
Newsletter<br><br>

  <textarea name="message" rows="4" cols="30" placeholder="Your
message..."></textarea><br><br>

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



Name:

Email:

Gender: ○ Male ○ Female

Course: HTML

□ Subscribe to Newsletter

Your message...

Submit

# <input> Types

| Type | Description | Example |
|---|---|---|
| text | Single-line text field | <input type="text"> |
| password | Hidden characters | <input type="password"> |
| email | Validates email format | <input type="email"> |
| number | Numeric input only | <input type="number"> |
| radio | Select one option | <input type="radio"> |
| checkbox | Select multiple options | <input type="checkbox"> |
| date | Choose a date | <input type="date"> |
| file | Upload files | <input type="file"> |
| submit | Submit the form | <input type="submit"> |
| reset | Reset all fields | <input type="reset"> |

# <input> Attributes

**1. type**

Specifies the kind of input control.

Examples:

```
<input type="text">

<input type="password">

<input type="email">

<input type="number">

<input type="date">
```

**2. name**

Used to identify form data when submitted to a server.

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

**3. value**

Sets a default value.

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

# <input> Attributes

**4. placeholder**

Shows light text inside the field as a hint.

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

**5. required**

Makes the field mandatory.

```
<input type="email" required>
```

**6. readonly**

Makes the input uneditable but still selectable.

```
<input type="text" value="Nepal" readonly>
```

**7. disabled**

Completely disables the input.

```
<input type="text" disabled>
```

**8. maxlength & minlength**

Controls the allowed number of characters.

```
<input type="text" maxlength="10">
```

```
<input type="text" minlength="4">
```

**9. size**

Sets input field width (in characters).

```
<input type="text" size="30">
```

# <input> Attributes

**10. pattern**

Uses regular expressions to validate input.

```
<input type="text" pattern="[A-Za-z]{3,}" placeholder="At least
3 letters">
```

**11. autocomplete**

Allows browser to auto-fill.

```
<input type="text" autocomplete="on">
```

```
<input type="email" autocomplete="off">
```

**12. autofocus**

Automatically focuses on this field when the page loads.

```
<input type="text" autofocus>
```

**13. min & max**

Sets the numeric/date limits.

```
<input type="number" min="1" max="10">
```

```
<input type="date" min="2024-01-01"
max="2024-12-31">
```

**14. step**

Defines step intervals.

```
<input type="number" min="0" max="100"
step="5">
```

# <input> Attributes

**15. multiple**

Allows selecting multiple files or emails.

```
<input type="file" multiple>
```

```
<input type="email" multiple>
```

**16. accept**

Specifies allowed file types.

```
<input type="file" accept="image/*">
```

```
<input type="file" accept=".pdf">
```

**17. checked**

Used with radio/checkbox to mark as selected by default.

```
<input type="radio" name="gender" checked> Male
```

# Semantic HTML

# Semantic Elements

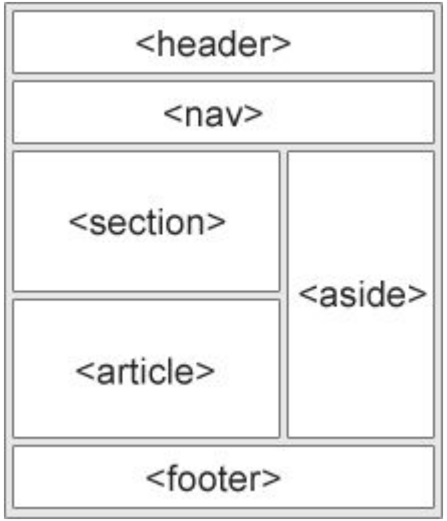
- Semantic elements = elements with a meaning.
- A semantic element clearly describes its meaning to both the browser and the developer.
- Examples of non-semantic elements: <div> and <span> - Tells nothing about its content.
- Examples of semantic elements: <img>, <table>, and <article> - Clearly defines its content.

# Semantic Element in HTML

Many web sites contain HTML code like: <div id="nav"> <div class="header"> <div id="footer"> to indicate navigation, header, and footer.

In HTML there are several semantic elements that can be used to define different parts of a web page:

- <article>
- <aside>
- <details>
- <figcaption>
- <figure>
- <footer>
- <header>
- <main>
- <mark>
- <nav>
- <section>
- <summary>
- <time>



# Why Semantic HTML Matters

- **Accessibility:** Screen readers and assistive technologies rely on semantic structure to navigate and present content to users with disabilities.
- **SEO (Search engines):** Search engines use semantic tags to understand the structure and importance of content.
- **Maintainability:** Code is easier to read and maintain when tags describe purpose (e.g., <header> vs <div class="header">).
- **Progressive enhancement / interoperability:** Semantic markup degrades more gracefully across devices, user agents, and future tech.
- **Default browser behavior:** Browsers apply reasonable default styles and behavior to many semantic elements.

# HTML <section> Element

The <section> element defines a section in a document.

According to W3C's HTML documentation: "A section is a thematic grouping of content, typically with a heading."

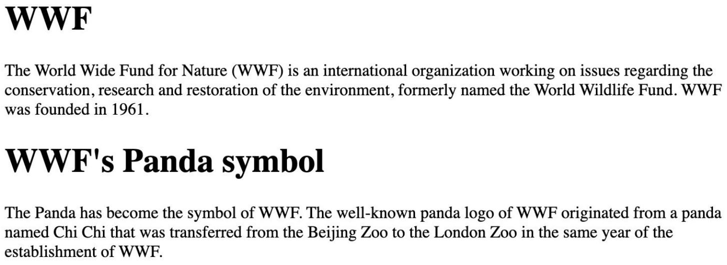
Examples of where a <section> element can be used:

- Chapters
- Introduction
- News items
- Contact information

A web page could normally be split into sections for introduction, content, and contact information.

# HTML <section> Element Example

```
<section>
<h1>WWF</h1>
<p>The World Wide Fund for Nature (WWF) is an
international organization working on issues regarding
the conservation, research and restoration of the
environment, formerly named the World Wildlife Fund.
WWF was founded in 1961.</p>
</section>
<section>
<h1>WWF's Panda symbol</h1>
<p>The Panda has become the symbol of WWF. The
well-known panda logo of WWF originated from a panda
named Chi Chi that was transferred from the Beijing Zoo
to the London Zoo in the same year of the establishment
of WWF.</p>
</section>
```



# WWF

The World Wide Fund for Nature (WWF) is an international organization working on issues regarding the conservation, research and restoration of the environment, formerly named the World Wildlife Fund. WWF was founded in 1961.

# WWF's Panda symbol

The Panda has become the symbol of WWF. The well-known panda logo of WWF originated from a panda named Chi Chi that was transferred from the Beijing Zoo to the London Zoo in the same year of the establishment of WWF.

# HTML <article> Element

The <article> element specifies independent, self-contained content.

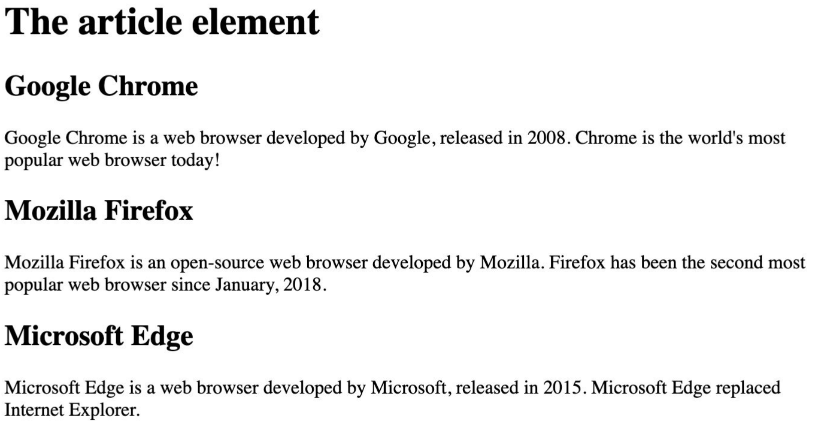
An article should make sense on its own, and it should be possible to distribute it independently from the rest of the web site.

Examples of where the <article> element can be used:

- Forum posts
- Blog posts
- User comments
- Product cards
- Newspaper articles

# HTML <article> Element Example

```
<article>
<h2>Google Chrome</h2>
<p>Google Chrome is a web browser developed by Google, released
in 2008. Chrome is the world's most popular web browser today!</p>
</article>

<article>
<h2>Mozilla Firefox</h2>
<p>Mozilla Firefox is an open-source web browser developed by
Mozilla. Firefox has been the second most popular web browser
since January, 2018.</p>
</article>

<article>
<h2>Microsoft Edge</h2>
<p>Microsoft Edge is a web browser developed by Microsoft,
released in 2015. Microsoft Edge replaced Internet Explorer.</p>
</article>
```



**The article element**

**Google Chrome**

Google Chrome is a web browser developed by Google, released in 2008. Chrome is the world's most popular web browser today!

**Mozilla Firefox**

Mozilla Firefox is an open-source web browser developed by Mozilla. Firefox has been the second most popular web browser since January, 2018.

**Microsoft Edge**

Microsoft Edge is a web browser developed by Microsoft, released in 2015. Microsoft Edge replaced Internet Explorer.

# Common Semantic Elements

| Element | Purpose / Meaning |
|---:|---|
| `<header>` | Introductory content or navigation for a page or section |
| `<nav>` | Primary navigation links |
| `<main>` | The main content unique to the page (one per page) |
| `<article>` | Self-contained composition (blog post, news article) |
| `<section>` | Thematic grouping of content (with heading) |
| `<aside>` | Side content related to main content (sidebar, pull-quote) |
| `<footer>` | Footer for page/section (copyright, links) |

# Common Semantic Elements

| Element | Purpose / Meaning |
|---:|---|
| <figure> | Self-contained content like images or diagrams |
| <figcaption> | Caption/legend for a <figure> |
| <mark> | Highlighted text of special relevance |
| <time> | Dates/times (machine-readable) |
| <address> | Contact information for author/owner |
| <details> / <summary> | Collapsible details widget (disclosure) |
| <main> | Main unique page content (only one allowed) |
| <strong> / <em> | Strong importance, emphasis (semantic emphasis) |

Any Questions?