HTML

Introduction

HTML is the standard markup language for creating Web pages.

What is HTML?

- HTML stands for Hyper Text Markup Language
- HTML is the standard markup language for creating Web pages
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content
- HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

Features:

- HTML is the standard markup language for Web pages.
- With HTML you can create your own Website.
- HTML is easy to learn - You will enjoy it!

Note: HTML files are saved with .html or .htm extension

Basic Structure

```
<!DOCTYPE html>
<html>
    <head>
        <title>Page Title</title>
    </head>
    <body>
        <h1>My First Heading</h1>
        <p>My first paragraph.</p>
    </body>
</html>
```

- The <!DOCTYPE html> declaration defines that this document is an HTML5 document
- The <html> element is the root element of an HTML page
- The <head> element contains meta information about the HTML page
- The <title> element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)
- The <body> element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.
- The <h1> element defines a large heading
- The <p> element defines a paragraph

Elements

An HTML element is defined by a start tag, some content, and an end tag. Examples:

```
<h1>My First Heading</h1>
```

Nested HTML Elements

HTML elements can be nested (this means that elements can contain other elements). All HTML documents consist of nested HTML elements. Ex.

```
<body>
        <h1>My First Heading</h1>
    <p>My first paragraph.</p>
</body>
```

Empty HTML Elements

HTML elements with no content are called empty elements. Example:

```
<p>This is a <br> paragraph with a line break.</p>
<hr>
```

Note: HTML is Not Case Sensitive. The HTML standard does not require lowercase tags, but W3C recommends lowercase in HTML.

Attributes

- All HTML elements can have attributes.
- Attributes provide additional information about elements.
- Attributes are always specified in the start tag.
- Attributes usually come in name/value pairs like: name="value".

The <a> tag defines a hyperlink. The href attribute specifies the URL of the page the link goes to. Ex:

Visit Link

The <img> tag is used to embed an image in an HTML page. The src attribute specifies the path to the image to be displayed: Ex. <img src="img.jpg">

The <img> tag should also contain the width and height attributes, which specifies the width and height of the image (in pixels). Ex: <img src="img_girl.jpg" width="500" height="600">

The style attribute is used to add styles to an element, such as color, font, size, and more. Ex:

This is a red paragraph.

The title attribute defines some extra information about an element. The value of the title attribute will be displayed as a tooltip when you mouse over the element. Ex. This is a paragraph.

Note: The HTML standard does not require quotes around attribute values. However, W3C recommends quotes in HTML. Double quotes around attribute values are the most common in HTML, but single quotes can also be used.

Comments

HTML comments are not displayed in the browser, but they can help document your HTML source code. You can add comments to your HTML source by using the following syntax:

```
<!-- Write your comments here -->
```

Note: There is an exclamation point (!) in the start tag, but not in the end tag.

With comments you can place notifications and reminders in your HTML code. Example:

```
<!-- This is a comment -->
<p>This is a paragraph.</p>
<!-- Remember to add more information here -->
```

Comments are also great for debugging HTML, because you can comment out HTML lines of code, one at a time, to search for errors. Example:

```
<!-- Do not display this image at the moment
     <img border="0" src="pic_trulli.jpg" alt="Trulli">
-->
```

Head

It's a container for metadata. Metadata typically defines the document title, character set, styles, scripts, and other meta information.

<title> - title of document. It defines a title in the browser tab, provides a title for the page when it is added to favorites, displays a title for the page in search-engine results.

<style> - internal css styles

<base> - specifies the base URL/target for all relative URLs in a document.

<meta> - is used to specify page description, keywords, author of the document, last modified charset and other metadata. The metadata is used by browsers (how to display content or reload page), search engines (keywords), or other web services. HTML5 allows web designers to take control over the viewport (visible area of a web page), through <meta> tag. A <meta> viewport element gives the browser instructions on how to control the page's dimensions and scaling. The width=device-width part sets the width of the page to follow the screen-width of the device (which will vary depending on the device). The initial-scale=1.0 part sets the initial zoom level when the page is first loaded by the browser. <meta name="viewport" content="width=device-width, initial-scale=1.0">

<link> - defines a link between a document and an external resource. It's used to link to external style sheets.

<script> - It's used to define a client-side script. It either contains scripting statements, or it points to an external script file through the src attribute.

Head(contd.)

There are several ways an external script can be executed:

If async="async": The script is executed asynchronously with the rest of the page. (the script will be executed while the page continues the parsing). There is no guarantee that scripts will run in any specific order, only that they will not stop the rest of the page from displaying. It is best to use async when the scripts in the page run independently from each other & depend on no other script on the page. If your scripts should be run immediately and they don't have any dependencies, then use async.

If defer="defer": The script is run in the order they appear in the page and execute them as soon as the script and content are downloaded. If your scripts need to wait for parsing and depend on other scripts and/or the DOM being in place, load them using defer & put their corresponding <script> elements in the order you want the browser to execute them.

If neither async or defer is present: The script is fetched and executed immediately, before the browser continues parsing the page.

<noscript> - defines an alternate content for users that have disabled scripts in their browser or have a browser that doesn't support script. The <noscript> element can be used in both <head> and <body>. When used inside the <head> element: <noscript> must contain only <link>, <style>, and <meta> elements. The content inside the <noscript> element will be displayed if scripts are not supported, or are disabled in the user's browser.

Headings & Horizontal Rule

Headings

They are defined with the <h1> to <h6> tags. <h1> defines the most important heading. <h6> defines the least important heading. Search engines use the headings to index the structure and content of your web pages. It is important to use headings to show the document structure. Ex:

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

Horizontal Rules

It defines a thematic break in an HTML page, and is displayed as a horizontal rule. It's used to separate content or define a change in a page. Ex:

```
<hr>
```

Paragraph & Formatting

Paragraph

It defines a paragraph. It is written as: Hello .
 element defines a line break. Use
 if you want a line break (a new line) without starting a new paragraph. <pre> element defines preformatted text. The text inside a <pre> element is displayed in a fixed-width font (usually Courier), and it preserves both spaces and line breaks.

Formatting

- <b> - Bold text
- - Important text
- <i> - Italic text
- - Emphasized text
- <mark> - Marked text
- <small> - Small text
- <del> - Deleted text
- <ins> - Inserted text
- <sub> - Subscript text
- <sup> - Superscript text

Anchor

Hyperlinks are defined with the HTML <a> tag. The target attribute specifies where to open the linked document and can have one of the following values:

- _blank - Opens the linked document in a new window or tab.
- _self - Opens the linked document in the same window/tab as it was clicked (default).
- _parent - Opens the linked document in the parent frame.
- _top - Opens the linked document in the full body of the window.
- framename - Opens the linked document in a named frame.

HTML bookmarks are used to allow readers to jump to specific parts of a Web page and can be created using <a> tag

- Use the id attribute (id="value") to define bookmarks in a page.
- Use the href attribute (href="#value") to link to the bookmark.

The title attribute specifies extra information about an element. The information is most often shown as a tooltip text when the mouse moves over the element.

Image

They are defined with the <img> tag. The <img> tag is empty, it contains attributes only, and does not have a closing tag. The src attribute specifies the URL (web address) of the image. The alt attribute provides an alternate text for an image, if the user for some reason cannot view it (because of slow connection, an error in the src attribute, or if the user uses a screen reader). It has width and height attributes which defines the width and height of the image in pixels. Ex:

```
<img src="img_girl.jpg" alt="Girl in a jacket" width="500" height="600">
```

The <picture> element contains a number of <source> elements, each referring to different image sources. This way the browser can choose the image that best fits the current view and/or device. Each <source> element has attributes describing when their image is the most suitable. The browser will use the first <source> element with matching attribute values, and ignore any following <source> elements. Always specify an <img> element as the last child element of the <picture> element. The <img> element is used by browsers that do not support the <picture> element, or if none of the <source> tags matched. Ex:

```
<picture>
  <source media="(min-width: 650px)" srcset="img_food.jpg">
  <source media="(min-width: 465px)" srcset="img_car.jpg">
  <img src="img_girl.jpg">
</picture>
```

Table

An HTML table is defined with the <table> tag. Each table row is defined with the <tr> tag. A table header cell is defined with the <th> tag which is bold and centered. A table cell is defined with the <td> tag. Ex:

```
<table style="width:100%">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
  </tr>
</table>
```

Table(contd.)

<thead>, <tbody> & <tfoot> groups the header, body & footer content respectively. Ex:

```
<table>
 <thead>
  <tr>
   <th>Month</th>    <th>Savings</th>
  </tr>
 </thead>
 <tbody>
  <tr>
   <td>January</td>    <td>$100</td>
  </tr>
 </tbody>
 <tfoot>
  <tr>
   <td>Sum</td>    <td>$180</td>
  </tr>
 </tfoot>
</table>
```

Table(contd.)

To add a caption to a table, use the <caption> tag. Ex:

```
<table>
  <caption>Monthly savings</caption>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
</table>
```

Table(contd.)

The rowspan attribute specifies the number of rows a cell should span. Ex:

```
<table style="width:100%">
  <tr>
    <th>Name:</th>
    <td>Bill Gates</td>
  </tr>
  <tr>
    <th rowspan="2">Telephone:</th>
    <td>55577854</td>
  </tr>
  <tr>
    <td>55577855</td>
  </tr>
</table>
```

Table(contd.)

The colspan attribute specifies the number of columns a cell should span. Ex:

```
<table style="width:100%">
  <tr>
    <th>Name</th>
    <th colspan="2">Telephone</th>
  </tr>
  <tr>
    <td>Bill Gates</td>
    <td>55577854</td>
    <td>55577855</td>
  </tr>
</table>
```

Lists

There are 2 types of list: unordered & ordered.

An **unordered list** starts with the <ul> tag. Each list item starts with the <li> tag. The list items will be marked with bullets (small black circles) by default. list-style-type property is used to define the style of the list item marker. Example:

```
<ul>
  <li>Coffee</li>
  <li type="square">Tea</li>
  <li>Milk</li>
</ul>
```

Types:

Disc - Sets the list item marker to a bullet (default)

Circle - Sets the list item marker to a circle

Square - Sets the list item marker to a square

None - The list items will not be marked

Lists(contd.)

An **ordered list** starts with the <ol> tag. Each list item starts with the <li> tag. The list items will be marked with numbers by default. type attribute of the <ol> tag, defines the type of the list item marker. Example:

```
<ol>
  <li>Coffee</li>
  <li type="a">Tea</li>
  <li>Milk</li>
</ol>
```

Types:

- type="1", The list items will be numbered with numbers (default)
- type="A", The list items will be numbered with uppercase letters
- type="a", The list items will be numbered with lowercase letters
- type="I", The list items will be numbered with uppercase roman numbers
- type="i", The list items will be numbered with lowercase roman numbers

Lists(contd.)

By default, an ordered list will start counting from 1. If you want to start counting from a specified number, you can use the start attribute. Example:

```
<ol start="50">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

A **description list** is a list of terms, with a description of each term. The <dl> tag defines the description list, the <dt> tag defines the term (name), and the <dd> tag describes each term. Example:

```
<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>
```

Block & Inline Elements

A **block-level element** always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can). Eg. div, table.

<address>	<article>	<aside>	<blockquote>	<canvas>	<dd>	
<dl>	<dt>	<fieldset>	<figcaption>	<figure>	<footer>	<form>
<h1>-<h6>	<header>	<hr>		<main>	<nav>	<noscript>
	<p>	<pre>	<section>	<table>	<tfoot>	
<video>						

An **inline element** does not start on a new line and only takes up as much width as necessary. Eg. span, img, etc.

<a>	<abbr>	<acronym>	<b>	<bdo>	<big>	
<button>	<cite>	<code>	<dfn>		<i>	<img>
<input>	<kbd>	<label>	<map>	<object>	<output>	<q>
<samp>	<script>	<select>	<small>			<sub>
<sup>	<textarea>	<time>	<tt>	<var>		

Note: A block level element has a top and a bottom margin, whereas an inline element does not.

Iframe

<iframe> used to display a web page within a web page. src attribute specifies the URL (web address) of the inline frame page. height and width attributes to specify the size of the iframe. Note: By default, an iframe has a border around it. An iframe can be used as the target frame for a link. The target attribute of the link must refer to the name attribute of the iframe. Eg.

```
<iframe src="demo_iframe.htm" name="iframe_a"></iframe>
<p>
        <a href="https://www.w3schools.com" target="iframe_a">W3Schools.com</a>
</p>
```

<iframe> is unsafe. Hence we should use the sandbox attribute that enables an extra set of restrictions for the content in the iframe. When the sandbox attribute is present, and it will:

- Treat the content as being from a unique origin.
- Block form submission.
- Block script execution.
- Disable APIs.
- Prevent links from targeting other browsing contexts.
- Prevent content from using plugins (through <embed>, <object>, <applet>, or other).
- Prevent the content to navigate its top-level browsing context.
- Block automatically triggered features (such as automatically playing a video or automatically focusing a form control).

Special text & special characters

Special text:

- The HTML <kbd> element represents user input, like kbd input or voice commands.
- The HTML <samp> element represents output from a program or computing system.
- The HTML <code> element defines a fragment of computer code.
- The HTML <var> element defines a variable. The variable could be a variable in a mathematical expression or a variable in programming context.

Text surrounded by these tags are typically displayed in a monospace font except <var>.

Entities(special characters)

Reserved characters in HTML must be replaced with character entities. Characters that are not present on your keyboard can also be replaced by entities. If you use the less than (<) or greater than (>) signs in your text, the browser might mix them with tags. Character entities are used to display reserved characters in HTML. A character entity looks like this: &entity_name; OR #entity_number; Eg. For <: < or #60; For >: > For ©: ©

A common character entity used in HTML is the non-breaking space: It will not break into a new line. Eg. If you write 10 spaces in your text, the browser will remove 9 of them. To add real spaces to your text, you can use the   character entity.

Form

It defines a form that is used to collect user input. Example:

```
<form action="/action_page.php" target="_blank" method="get" novalidate>
  <fieldset>
    <legend>Personal information:</legend>
            First name:<br>
    <input type="text" name="firstname" value="Mickey"><br>
            Last name:<br>
    <input type="text" name="lastname" value="Mouse"><br><br>
    <input type="submit" value="Submit">
  </fieldset>
</form>
```

Each input field must have a name attribute to be submitted. If the name attribute is omitted, the data of that input field will not be sent at all.

Form(contd.)

Form Elements

- The <input> element can be displayed in several ways, depending on the type attribute. If the type attribute is omitted, the input field gets the default type: "text". The default width of a text field is 20 characters.
- The <select> element defines a drop-down list. The <option> elements defines an option that can be selected. By default, the first item in the drop-down list is selected. To define a pre-selected option, add the selected attribute to the option. size attribute specifies the number of visible values. multiple attribute allows the user to select more than one value. Eg.

```
<select name="cars" size="4" multiple>
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat" selected>Fiat</option>
  <option value="audi">Audi</option>
</select>
```

- The <textarea> element defines a multi-line input field. rows attribute specifies the visible number of lines & cols attribute specifies the visible width of a text area. Eg.

```
<textarea name="message" rows="10" cols="30">
```

Form(contd.)

- The <button> element defines a clickable button.
- The <datalist> HTML5 element specifies a list of pre-defined options for an <input> element. Users will see a drop-down list of the pre-defined options as they input data. The list attribute of the <input> element, must refer to the id attribute of the <datalist> element. Eg.

```
<form action="/action_page.php">
  <input list="browsers">
  <datalist id="browsers">
    <option value="Internet Explorer">
    <option value="Firefox">
    <option value="Chrome">
    <option value="Opera">
    <option value="Safari">
  </datalist>
</form>
```

Form(contd.)

- The <output> element represents the result of a calculation (like one performed by a script). Ex:

```
<form oninput="x.value=parseInt(a.value)+parseInt(b.value)">
  <input type="number" id="a" name="a"> +
  <input type="number" id="b" name="b"> =
  <output name="x" for="a b"></output>
</form>
```

Input Types

- <input type="button"> button.
- <input type="checkbox"> checkbox field.
- <input type="color"> color field.
- <input type="date"> date field. Eg. <input type="date" name="bday" min="1979-12-31" max="2000-01-02">
- <input type="datetime-local"> date and time input field, with no time zone.
- <input type="email"> e-mail address field.
- <input type="file"> file-select field and a "Browse" button for file uploads.

Form(contd.)

- <input type="hidden"> defines a hidden input field. A hidden field let web developers include data that cannot be seen or modified by users when a form is submitted. A hidden field often stores what database record that needs to be updated when the form is submitted.
- <input type="image"> defines an image as a submit button.
- <input type="month"> allows the user to select a month and year.
- <input type="number"> numeric input field.

  ```
  <input type="number" name="quantity" min="1" max="5">
  ```
- <input type="password"> password field.
- <input type="radio"> radio button. Eg.

  ```
  <form>
    <input type="radio" name="gender" value="male" checked> Male<br>
    <input type="radio" name="gender" value="female"> Female<br>
    <input type="radio" name="gender" value="other"> Other
  </form>
  ```

Form(contd.)

- <input type="range"> a control for entering a number whose exact value is not important (like a slider control). Default range is 0 to 100. However, you can set restrictions on what numbers are accepted with the min, max, and step attributes.
- <input type="reset"> a reset button that will reset all form values to their default values.
- <input type="search"> search field.
- <input type="submit"> a button for submitting form data to a form-handler.
- <input type="tel"> input fields that should contain a telephone number.
- <input type="text"> one-line text input field.
- <input type="time"> allows the user to select a time (no time zone).
- <input type="url"> input fields that should contain a URL address.
- <input type="week"> allows the user to select a week and year.

Input Attributes

These attributes can be used with only one **one type**.

- value specifies the initial value for an input field.
- readonly specifies that the input field is read only (cannot be changed).

Form(contd.)

- <input type="range"> a control for entering a number whose exact value is not important (like a slider control). Default range is 0 to 100. However, you can set restrictions on what numbers are accepted with the min, max, and step attributes.
- <input type="reset"> a reset button that will reset all form values to their default values.
- <input type="search"> search field.
- <input type="submit"> a button for submitting form data to a form-handler.
- <input type="tel"> input fields that should contain a telephone number.
- <input type="text"> one-line text input field.
- <input type="time"> allows the user to select a time (no time zone).
- <input type="url"> input fields that should contain a URL address.
- <input type="week"> allows the user to select a week and year.

Input Attributes

These attributes can be used with only one **one type**.

- value specifies the initial value for an input field.
- readonly specifies that the input field is read only (cannot be changed).

Form(contd.)

- disabled specifies that the input field is disabled. This field is unusable and unclickable, and its value will not be sent when submitting the form.
- size specifies the size (in characters) for the input field.
- maxlength specifies the maximum allowed length for the input field.
- autofocus specifies that the input field should automatically get focus on page load.
- form specifies one or more forms an <input> element belongs to.
- height and width specify the height and width of an <input type="image"> element.
- list refers to a <datalist> that contains predefined options for <input> element.
- autocomplete specifies whether a form or input field should have autocomplete on or off. When it's on, the browser automatically completes the input values based on values that the user has entered before. Tip: It is possible to have autocomplete "on" for the form, and "off" for specific input fields, or vice versa. It works with <form> and the following <input> types: text, search, url, tel, email, password, datepickers, range, and color.
- novalidate is a <form> attribute. When present, novalidate specifies that the form data should not be validated when submitted.

Form(contd.)

These attributes can be used with **multiple types**.

- min and max specify the minimum and maximum values for an <input> element. They work with: number, range, date, datetime-local, month, time and week.
- multiple specifies that the user is allowed to enter more than one value in the <input> element. It works with: email, and file.
- pattern specifies a regular expression that the <input> element's value is checked against. It works with: text, search, url, tel, email & password.
- placeholder specifies a hint that describes the expected value of an input field. The hint is displayed in the input field before the user enters a value. It works with: text, search, url, tel, email, and password.
- required specifies that an input field must be filled out before submitting the form. It works with: text, search, url, tel, email, password, date pickers, number, checkbox, radio, and file.
- step specifies the legal number intervals for an <input> element. It works with: number, range, date, datetime-local, month, time and week.

Form(contd.)

The following attributes are used with type="submit" and type="image" and they override the related attribute of the <form> element.

- formaction specifies the URL of a file that will process the input control when the form is submitted. Overrides action attribute.
- formenctype specifies how the form data should be encoded when submitted (only for forms with method="post"). Overrides enctype attribute.
- formmethod defines the HTTP method for sending form-data to the action URL. Overrides method attribute.
- formnovalidate specifies that the form data should not be validated when submitted. Overrides novalidate attribute. Only used with type="submit".
- formtarget specifies a name or a keyword that indicates where to display the response that is received after submitting the form. Overrides target attribute.

Canvas & SVG

Canvas used to draw graphics on a web page, on the fly, via JS. The <canvas> element is only a container for graphics and JS is used to actually draw the graphics. Canvas has several methods for drawing paths, boxes, circles, text, and adding images. By default, a canvas has no border and no content. Eg.

```
<canvas id="myCanvas" width="200" height="100"></canvas>
<script>
        var c = document.getElementById("myCanvas");
        var ctx = c.getContext("2d");
        ctx.moveTo(0, 0);
        ctx.lineTo(200, 100);
        ctx.stroke();
</script>
```

Scalable Vector Graphics is used to define graphics for the Web. The HTML <svg> element is a container for SVG graphics. SVG has several methods for drawing paths, boxes, circles, text, and graphic images. Eg.

```
<svg width="100" height="100">
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />
</svg>
```

Canvas vs SVG

Differences Between SVG and Canvas:

- SVG is a language for describing 2D graphics in XML. Canvas draws 2D graphics, on the fly (with JavaScript).
- SVG is XML based, which means that every element is available within the SVG DOM. You can attach JavaScript event handlers for an element.
- In SVG, each drawn shape is remembered as an object. If attributes of an SVG object are changed, the browser can automatically re-render the shape.
- Canvas is rendered pixel by pixel. In canvas, once the graphic is drawn, it is forgotten by the browser. If its position should be changed, the entire scene needs to be redrawn, including any objects that might have been covered by the graphic.

Canvas	**SVG**
Resolution dependent	Resolution independent
Poor text rendering capabilities	You can save the resulting image as .png or .jpg
Well suited for graphic-intensive games	Not suited for game applications
Support for event handlers	No support for event handlers
Slow rendering if complex	Best suited for applications with large rendering areas
(anything that uses the DOM a lot will be slow)	(Google Maps)

Video

The HTML5 <video> element specifies a standard way to embed a video in a web page. HTML5 supports only MP4, WebM, and Ogg video. Eg.

```
<video width="320" height="240" controls autoplay>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  <track src="sub.vtt" kind="subtitles" srclang="en" label="English">
  Your browser does not support the video tag.
</video>
```

- It is a good idea to always include width and height attributes because if they are not set, the page might flicker while the video loads.
- The controls attribute adds video controls, like play, pause, and volume.
- The autoplay attribute starts the video automatically.
- The <source> element allows you to specify alternative video files which the browser may choose from. The browser will use the first recognized format.
- The <track> element is used to specify subtitles, caption files or other files containing text that should be visible when the media is playing.
- The text between the <video> and </video> tags will only be displayed in browsers that do not support the <video> element.

Audio

The HTML5 <audio> element specifies a standard way to embed audio in a web page.

HTML5 supports only MP3, WAV, and Ogg audio. Eg.

```
<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
  Your browser does not support the audio element.
</audio>
```

- The controls attribute adds audio controls, like play, pause, and volume.
- The <source> element allows you to specify alternative audio files which the browser may choose from. The browser will use the first recognized format.
- The text between the <audio> and </audio> tags will only be displayed in browsers that do not support the <audio> element.

Thank You