HTML Summarized Document

Agenda:

HTML Introduction

HTML (HyperText Markup Language) is the standard language used to create web pages. It is a markup language that defines the structure and content of a web page. HTML consists of a series of elements and tags that are used to specify the structure and content of a web page.

Each HTML element is represented by a tag, which is enclosed in angle brackets (e.g.,). The tags define the type of content contained within the element, such as text, images, and links. The content of the element is placed between the opening and closing tags. For example, to create a paragraph element, you would use the tag, like this:

** This is a paragraph of text. **

HTML is the foundation of all websites, and is supported by all modern web browsers. It is a simple and flexible language that allows developers to create a wide range of web pages, from basic informational pages to complex web applications.

HTML5 is the latest version of HTML, and provides several new elements and attributes that make it easier to create web pages and web applications. HTML5 introduces new semantic elements, such as **<header>, <nav>, <main>, <article>, and <footer>,** which help to define the structure of a web page and make it easier for search engines and screen readers to understand the content. HTML5 also introduces new multimedia elements, such as **<video> and <audio>,** which make it easier to embed multimedia content in a web page without the need for additional plugins.

HTML File Structure

The HTML file structure refers to the basic structure of an HTML document, which consists of several key components:

Document Type Declaration (DOCTYPE): The DOCTYPE declaration defines the version of HTML being used in the document. It is the first line of the document and is used by web browsers to determine how to render the content.

HTML Element: The HTML element is the root element of the document and contains all other elements. It is defined using the <html> tag.

Head Element: The head element, defined using the <head> tag, contains information about the document that is not displayed to the user, such as the title of the page, metadata, and links to stylesheets.

Body Element: The body element, defined using the <body> tag, contains the content that is displayed to the user.

Here is an example of the basic HTML file structure:

```
<!DOCTYPE html>
```

```
<html>
        <head>
           <title>Title of the Page</title>
        </head>

        <body>

        </body>
</html>
```

In this example, the document is declared as an HTML5 document using the DOCTYPE declaration. The head element contains the title of the page, which is displayed in the browser tab, and the body element contains the content of the page. The content of the body element can include text, images, links, and other HTML elements.

HTML Tags

A tag is used to define an HTML element. A tag consists of the element name, enclosed in angle brackets **< >.** For example, the tag ** ** defines a paragraph element.
 is a tag, is a tag, <h1> is a tag, and so on.

HTML Elements

An HTML element is a component of an HTML document that has a specific purpose, such as defining headings, paragraphs, links, images, and more. The element is defined by the tag and its attributes, and the content of the element is defined by the text or other elements that are contained within the opening and closing tags.

For example, the following code defines a paragraph element with the text "This is a paragraph of text.":

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

HTML Head Tag

The HTML <head> tag is used to contain information about the web page that isn't displayed directly to the user. This information includes the title of the page, links to stylesheets, links to JavaScript files, and metadata such as the author of the page and keywords that describe the content of the page.

Here are some common elements that can be included in the <head> section of an HTML document:

<title> - This element is used to specify the title of the web page, which is displayed in the browser's title bar.

Example:

```
<head>
  <title>My Web Page</title>
</head>
```

<meta> - This element is used to include metadata about the page, such as keywords and descriptions that can be used by search engines.

Example:

```
<head>
  <meta name="description" content="This is a description of my web page">
  <meta name="keywords" content="HTML, CSS, JavaScript">
</head>
```

<link> - This element is used to include external resources such as stylesheets and icon files.
Example:

```
<head>
  <link rel="stylesheet" href="styles.css">
  <link rel="icon" href="favicon.ico">
</head>
```

<script> - This element is used to include JavaScript code in the page.

Example:

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

<base> - This element is used to specify the base URL for all relative URLs in the page.
Example:

```
<head>
  <base href="http://example.com/">
</head>
```

<style> - This element is used to include CSS styles directly in the HTML document.

Example:

```
<head>
  <style>
    body {
      background-color: #f2f2f2;
    }
  </style>
</head>
```

Overall, the <head> tag is an essential part of an HTML document that helps search engines and browsers to better understand and display the content of the page.

HTML Body Tag

The HTML <body> tag is used to define the body section of an HTML document. It contains all the visible contents of an HTML page, such as text, images, links, videos, and other multimedia elements. The <body> tag is a required element and must be included in all HTML documents.

The <body> tag can have various attributes such as bgcolor, background, text, link, and vlink. These attributes allow you to specify the background color or image of the body, text color, and hyperlink colors.

Example:

```
<!DOCTYPE html>
<html>
  <head>
    <title>My HTML Page</title>
  </head>
  <body bgcolor="#f2f2f2">
    <h1>Welcome to my website</h1>
    <p>This is a paragraph of text.</p>
    <img src="image.jpg" alt="My Image">
    <a href="https://www.example.com">Visit Example.com</a>
  </body>
</html>
```

In the above example, the <body> tag has a bgcolor attribute set to #f2f2f2, which sets the background color of the body to light gray. The body contains a heading, paragraph of text, an image, and a hyperlink to another website.

The <body> tag can also contain other HTML tags such as headings, paragraphs, lists, tables, forms, and more. You can use CSS to style the contents of the body, such as changing font sizes, colors, and layout.

HTML Heading Tag

The HTML heading tags, <h1>, <h2>, <h3>, <h4>, <h5>, and <h6>, are used to create headings and subheadings within an HTML document. These tags provide a way to structure the content of a web page and indicate the hierarchy of information.

The <h1> tag represents the most important heading, and each subsequent heading tag, <h2>, <h3>, etc., represents a lesser level of importance. For example, the following code creates a main heading and three subheadings:

```
<h1>Main Heading</h1>
<h2>First Subheading</h2>
<h3>Second Subheading</h3>
<h4>Third Subheading</h4>
```

It is recommended to use only one <h1> tag per page, as it represents the main topic of the page, and to use the other heading tags to create a hierarchy of subheadings.

The heading tags also affect the font size and style of the text, with the <h1> tag having the largest font size and the <h6> tag having the smallest font size.

HTML Paragraph Tag

The HTML <p> tag, also known as the HTML paragraph tag, is used to define a paragraph of text in an HTML document. The text within a <p> tag is automatically formatted with a new line before and after the paragraph. For example:

```
<p>This is a paragraph of text.</p>
<p>This is another paragraph of text.</p>
```

In this example, the two paragraphs of text are automatically separated by a new line. The <p> tag also automatically adds margins above and below the paragraph to separate it from other elements on the page.

It's important to use the <p> tag to properly format and structure text within an HTML document. This makes the document easier to read and understand, and helps with accessibility for users with disabilities.

HTML Anchor Tag

Utility 1:

The <a> tag, also known as the HTML anchor tag, is used to create a hyperlink to another web page or to a specific location within the same page. The <a> tag provides a convenient way for users to navigate from one page to another by clicking on a link.

The href attribute of the <a> tag is used to specify the destination URL for the hyperlink. For example, the following code creates a link to the OpenAI homepage:

```
<a href="https://www.mygreatlearning.com/entry-level-professionals">Great Learning</a>
```

In this example, the text "Great Learning" is displayed as the link text, and when the user clicks on the link, they will be redirected to the Great Learning homepage.

Utility 2:
To open the given link in the new tab, we will use the below html code.

```
<a href="https://www.mygreatlearning.com/entry-level-professionals" target="_blank"> Great Learning</a>
```

Utility 3:
The <a> tag can also be used to create links to specific locations within the same page by using the id (we will learn about id little later) attribute of the target element and the # symbol followed by the value of the id attribute in the href attribute of the <a> tag. For example:

```
<a href="#section1">Go to Section 1</a>
...
<h2 id="section1">Section 1</h2>
```

In this example, the link **"Go to Section 1"** will take the user to the heading with the id **"section1"**.

HTML Attributes

HTML attributes are additional properties that are used to provide more information about an HTML element. They are defined within the opening tag of the element and are used to specify the characteristics and behavior of the element.

For example, the **href attribute** of the <a> tag is used to specify the destination URL for the hyperlink. The style attribute is used to specify the styling information for an element, such as font size, color, and background.

Attributes are typically defined as name-value pairs, with the name of the attribute followed by an equal sign = and the value of the attribute enclosed in quotation marks "".

```
<a href="https://www.mygreatlearning.com/entry-level-professionals">Great Learning</a>
```

Here, href attribute has value as url. This will help us to go to the particular web page.

The **title** attribute can be added to an HTML link to provide additional information about the target resource. When the user hovers over the link, the title text will be displayed in most browsers as a tooltip.

Here's an example of a link with a title:

```
<a href="https://www.example.com" title="Visit Example.com">Example</a>
```

The title attribute is especially useful for providing additional context or information about the target resource, such as a brief description or the purpose of the link. This can help users understand where the link will take them and what they can expect to find on the target page.

Note that the title attribute is not essential for all links, and its use is optional. However, it can be a helpful tool for improving the accessibility and usability of your website.

HTML Image Tag

The HTML <img> tag is used to embed images into an HTML document. **The <img> tag does not have a closing tag**, and instead uses the src attribute to specify the URL or file path of the image to be displayed.

For example, the following code will display the image located at https://www.example.com/image.jpg:

```
<img src="https://www.example.com/image.jpg">
```

In addition to the src attribute, there are other attributes that can be used with the <img> tag to control the appearance and behavior of the image, such as alt (alternative text), width and height, and style.

For example, the following code sets the width and height of the image and provides alternative text to be displayed if the image cannot be loaded:

```
<img
src="https://bsmedia.business-standard.com/_media/bs/img/article/2021-08/04/full/1628096181-7187.jpg" alt="A beautiful landscape" width="500" height="300">
```

The alt attribute is especially important for accessibility purposes, as it provides a text description of the image for users with visual impairments who use screen readers.

Fetch images from local repositories.

To fetch an image from local storage in HTML, you can use the img tag with a src attribute that specifies the path to the image file.

Here's an example of how to display an image stored in the same directory as the HTML file:

Directory Structure

Demo
→ Learn-image-tag.html
→ my-image1.jpg
→ images-folder
→my-image2.jpg

In Learn-image-tag.html

```
<img src="my-image1.jpg" alt="My Image">
```

In the example above, the src attribute specifies the file name of the image, and the alt attribute provides a text description of the image for accessibility purposes.

If the image is stored in a subdirectory of the HTML file, you will need to specify the relative path to the image file in the src attribute. For example:

In Learn-image-tag.html

```
<img src="images-folder/my-image2.jpg" alt="My Image">
```

If the image is located on a different drive or storage device, you can still display it in HTML by using the complete file path in the src attribute of the img tag.

Here's an example of how to display an image stored on the D: drive:
Directory Structure

C:

Demo
→ Learn-image-tag.html
→ my-image1.jpg
→ images-folder
→my-image2.jpg

D:
images
→my-image3.jpg

In Learn-image-tag.html

```
<img src="D:/images/my-image3.jpg" alt="My Image">
```

Note that the file path must be correct and accessible by the browser in order for the image to display. If the image cannot be found, the alt text will be displayed instead.

It's important to note that images stored in local storage will only be accessible to users who have access to the local storage location on their own computer. To display images on a website, you will need to upload them to a web server or host them on a third-party image hosting service.

HTML Formatting Tags

HTML formatting tags are used to apply specific styles or effects to text in an HTML document. These tags are used to specify how text should be displayed, including font size and color, bold or italic text, and text alignment.

Here are some of the most commonly used HTML formatting tags:

Tag	Explanation	Example
****	This tag is used to bold text.	This text will be bold
****	This tag is used to semantically emphasize text. It is typically displayed as bold text, but this can be changed with CSS.	This text will be semantically emphasized
****	This tag is used to italicize text.	This text will be italic
****	This tag is used to semantically emphasize text. It is typically displayed as italic text, but this can be changed with CSS.	This text will be semantically emphasized
<u>	This tag is used to underline text.	<u>This text will be underlined</u>

<mark>	This tag is used to highlight text.	<mark>This text will be highlighted</mark>
<small>	This tag is used to make text smaller.	This text will be smaller
<sub>	This tag is used to display text as subscript.	_{This text will be subscript}.
<sup>	This tag is used to display text as superscript.	^{This text will be superscript}
<pre>	This tag is used to display preformatted text, preserving line breaks and whitespace.	<pre>This text will be preformatted</pre>

In addition to these basic formatting tags, HTML also provides tags for specifying text alignment, such as <p> (paragraph) and <h1> to <h6> (headings).

It's important to use these tags in an HTML document in a semantic and meaningful way, as it can impact the accessibility of the document for users with disabilities, as well as the search engine optimization of the site.

HTML Comments

HTML comments are used to add notes to the HTML code that will not be displayed in the final rendered web page. Comments are useful for documenting the purpose of sections of code, temporarily disabling code, and for collaboration among developers.

HTML comments start with <!-- and end with -->. Everything between these markers is considered a comment and will not be rendered in the final web page. For example:

```
<!-- This is an HTML comment -->
```

It's important to note that HTML comments can span multiple lines and can contain any characters except the --> closing tag.

HTML Style Tag

The style attribute in HTML allows you to specify inline styles for a specific HTML element. It provides a way to apply styles directly to an element, rather than having to create separate CSS rules in an external stylesheet.

The syntax of the style attribute is as follows:

```
<element_name style="property:value;">
```

Where element_name is the name of the HTML element you want to style, property is the CSS property you want to change, and value is the value you want to set for that property.

For example, the following code applies a red background color to a paragraph element:

```
<p style="background-color:red;">This is a paragraph with a red background.</p>
```

It's important to note that inline styles have a higher priority over external stylesheets, so if you have conflicting styles for an element, the inline style will override the external stylesheet.

HTML Style common properties

Here are some examples of common CSS properties and their values used to style HTML elements:

color:

```
<p style="color: blue;">This text is blue.</p>
```

background-color:

```
<div style="background-color: yellow;">This element has a yellow background.</div>
```

font-size:

```
<p style="font-size: 16px;">This text has a font size of 16 pixels.</p>
```

font-family:

```
<p style="font-family: Arial;">This text uses the Arial font family.</p>
```

text-align:

```
<p style="text-align: center;">This text is centered.</p>
```

padding:

```
<div style="padding: 10px;">This element has 10 pixels of padding on all sides.</div>
```

margin:

```
<p style="margin: 20px;">This element has a 20 pixel margin on all sides.</p>
```

width:

```
<div style="width: 200px;">This element has a width of 200 pixels.</div>
```

height:

```
<div style="height: 100px;">This element has a height of 100 pixels.</div>
```

border:

```
<div style="border: 1px solid black;">This element has a 1 pixel solid black border.</div>
```

These are just a few examples of the many CSS properties that can be used to style HTML elements. By combining different properties and values, you can create complex and custom styles for your web pages.

HTML Links

HTML links are elements that allow users to navigate between web pages or other resources on the web. The <a> tag is used to create a link in HTML, and the href attribute specifies the target of the link.

Here's an example of a basic HTML link:

```
<a href="https://www.example.com">Visit Example.com</a>
```

When this link is clicked, the user will be redirected to the specified URL: https://www.example.com.

Links can also be used to link to other resources on the web, such as email addresses, phone numbers, and files. Here are a few examples:

Email link:

```
<a href="mailto:example@email.com">Send an email to example@email.com</a>
```

Phone link:

```
<a href="tel:555-555-5555">Call 555-555-5555</a>
```

File link:

```
<a href="downloads/file.pdf">Download the file</a>
```

Links are a fundamental part of web design and are used to connect different pages and resources on the web.

Types of Links

In HTML, links can be either absolute or relative.

An absolute link specifies the full URL of the target resource, including the protocol (such as http:// or https://) and domain name. For example:

```
<a href="https://www.example.com">Visit Example.com</a>
```

A relative link, on the other hand, specifies the target resource in relation to the current page's URL. For example, if you have a file about.html in the same directory as the current page, you can create a relative link to it like this:

```
<a href="about.html">Learn more about us</a>
```

Using relative links is recommended when linking to resources within your own website, as it allows you to move pages or resources around without breaking the links. Absolute links are typically used when linking to external resources or when you need to specify the exact URL of the target resource.

HTML Lists

HTML lists are used to present information in an ordered or unordered fashion. There are two types of lists in HTML: ordered and unordered.

Unordered Lists:
An unordered list is created using the <ul> tag. The items in the list are represented by <li> (list item) tags. The browser renders each item in the list with a bullet point or some other symbol to denote the start of the item.

For example:

```
<ul>
  <li>Item 1</li>
```

```
 <li>Item 2</li>
 <li>Item 3</li>
</ul>
```

Output:

- Item 1
- Item 2
- Item 3

Ordered Lists:
An ordered list is created using the <ol> tag. The items in the list are represented by <li> (list item) tags. The browser renders each item in the list with a sequential number, starting with 1.

For example:

```
<ol>
 <li>Item 1</li>
 <li>Item 2</li>
 <li>Item 3</li>
</ol>
```

Output:

1. Item 1
2. Item 2
3. Item 3

Nested Lists:
You can also create nested lists in HTML. For example, you can create a list of items, and then have a sublist of items underneath one of the main items.

For example:

```
<ul>
 <li>Item 1</li>
 <li>Item 2
  <ul>
   <li>Sub-item 1</li>
   <li>Sub-item 2</li>
   <li>Sub-item 3</li>
  </ul>
```

```
  </li>
  <li>Item 3</li>
</ul>
```

Output:

- Item 1
- Item 2
    - Sub-item 1
    - Sub-item 2
    - Sub-item 3
- Item 3

Attributes:
You can also use various attributes to modify the appearance and behavior of your lists, such as the start attribute to set the starting number for an ordered list, the type attribute to set the type of bullet or number to be used, and the value attribute to set the value of an individual list item.

For example:

```
<ol start="10">
  <li>Item 0</li>
  <li value="11">Item 1</li>
  <li>Item 2</li>
  <li type="A">Item 3</li>
</ol>
```

Output:

10. Item 0
11. Item 1
12. Item 2
M. Item 3

(We have M here because 13th letter is M)

HTML Tables

HTML tables are a powerful tool for displaying data in a structured and organized manner. In this document, we will explore the various components of HTML tables, including how to create tables, format them, and manipulate their content.

Creating a Basic HTML Table

To create a basic HTML table, we use the <table> tag to define the table itself, and then the <tr> tag to define each row, and the <td> tag to define each cell. Here is an example of a simple table with two rows and two columns:

```
<table>
  <tr>
    <td>Row 1, Column 1</td>
    <td>Row 1, Column 2</td>
  </tr>
  <tr>
    <td>Row 2, Column 1</td>
    <td>Row 2, Column 2</td>
  </tr>
</table>
```

In this example, we have a table with two rows and two columns. Each row is defined by the <tr> tag, and each cell is defined by the <td> tag. The content of each cell is simply the text within the tags.

Table Headers

In some cases, we may want to add header rows to our table. We can do this using the <th> tag, which is similar to the <td> tag, but is used specifically for header cells. Here is an example of a table with a header row:

```
<table>
  <tr>
    <th>Header 1</th>
    <th>Header 2</th>
  </tr>
  <tr>
    <td>Row 1, Column 1</td>
    <td>Row 1, Column 2</td>
  </tr>
  <tr>
    <td>Row 2, Column 1</td>
    <td>Row 2, Column 2</td>
  </tr>
</table>
```

In this example, we have added a header row to our table by using the <th> tag. Note that this row is formatted differently than the other rows in the table, with a bold font and centered alignment.

Table Borders

By default, HTML tables have no borders, which can make them difficult to read. We can add borders to our tables using CSS. Here is an example of how to add a border to a table:

```
<style>
table {
  border: 1px solid black;
}
td, th {
  border: 1px solid black;
}
</style>
```

In this example, we have added a border to the table using the border property, and then added a border to each cell using the td and th selectors.

Table Widths

We can also specify the width of our table using CSS. Here is an example of how to set the width of a table to 50% of the page width:

```
<style>
table {
  width: 50%;
}
</style>
```

In this example, we have set the width property of the table to 50%. This will cause the table to take up 50% of the width of its parent element.

Table Spanning

In some cases, we may want a cell to span multiple columns or rows. We can do this using the colspan and rowspan attributes.

colspan and **rowspan** are attributes that can be added to table cells in HTML to span the cell across multiple columns or rows.

The colspan attribute specifies how many columns a cell should span, while the rowspan attribute specifies how many rows a cell should span. These attributes are used to merge cells horizontally or vertically in a table.

Here's an example of how to use colspan and rowspan in HTML:

```
<table>
  <tr>
    <td rowspan="2">Cell 1</td>
    <td>Cell 2</td>
    <td>Cell 3</td>
  </tr>
  <tr>
    <td colspan="2">Cell 4</td>
  </tr>
  <tr>
    <td>Cell 5</td>
    <td>Cell 6</td>
    <td>Cell 7</td>
  </tr>
</table>
```

In this example, the first cell spans 2 rows using rowspan="2", while the second cell spans 2 columns using colspan="2". The resulting table will have a merged cell in the first column that spans two rows and a merged cell in the second and third columns that spans two columns.

Keep in mind that when using colspan and rowspan, the total number of columns and rows in the table may change. It is important to maintain a consistent number of cells per row to avoid formatting issues.

HTML IFrames

An HTML iframe (short for "inline frame") is an HTML element that allows you to embed another HTML document inside your main HTML document. The content of the iframe can be anything that can be displayed in a web browser, such as HTML, CSS, JavaScript, images, videos, and so on. Here are some key features and attributes of iframes in HTML:

Syntax: To create an iframe, you use the <iframe> tag, with the src attribute specifying the URL of the document to be displayed. For example:

```
<iframe src="https://www.example.com"></iframe>
```

Width and Height: You can specify the width and height of the iframe using the width and height attributes. For example:

```
<iframe src="https://www.example.com" width="500" height="300"></iframe>
```

Borders and scrolling: You can control the display of borders and scrolling bars on the iframe using the frameborder and scrolling attributes. For example:

```
<iframe src="https://www.example.com" frameborder="0" scrolling="no"></iframe>
```

Target: You can specify a target for the iframe using the target attribute. For example:

```
<iframe src="https://www.example.com" target="_parent"></iframe>
```

Security considerations: When using iframes, it's important to consider security issues such as cross-site scripting (XSS) attacks and clickjacking. To mitigate these risks, you can use the sandbox attribute to restrict the behavior of the iframe, and use the allow attribute to specify which features are allowed to be used.
For example:

```
<iframe src="https://www.example.com" sandbox="allow-scripts"></iframe>
```

Accessibility: When using iframes, it's important to ensure that the content is accessible to users with disabilities. You can use the title attribute to provide a description of the content, and ensure that the content is keyboard accessible and screen reader friendly.

HTML Forms

HTML forms provide a way to collect user input and send it to a web server for processing. A form is an HTML element that contains various form elements such as input fields, text areas, checkboxes, radio buttons, and drop-down lists, which allow users to input data. When a user submits a form, the data is sent to a server for processing, and the server sends back a response to the user.

Here are some key features and attributes of HTML forms:

Syntax: To create a form, you use the <form> tag, with the action attribute specifying the URL of the server-side script that will process the form data. For example:

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

Form Elements: A form contains various form elements, such as input fields, text areas, checkboxes, radio buttons, and drop-down lists, which allow users to input data. Each form element is defined using an HTML element such as <input>, <textarea>, <select>, and so on.

Method: The method attribute of the <form> tag specifies the HTTP method to be used when submitting the form. The two most common methods are "get" and "post". The "get" method appends the form data to the URL, while the "post" method sends the form data in the body of the HTTP request.

Encoding: When a form contains file uploads, the encoding attribute of the <form> tag must be set to "multipart/form-data".

Submit Button: A form usually contains a submit button, which is defined using the <input> tag with type="submit". When the user clicks the submit button, the form data is sent to the server for processing.

Reset Button: A form may also contain a reset button, which is defined using the <input> tag with type="reset". When the user clicks the reset button, all form elements are reset to their default values.

Hidden Fields: A form can also contain hidden fields, which are used to store data that should not be visible to the user. Hidden fields are defined using the <input> tag with type="hidden".

HTML Form Elements

HTML form elements are the building blocks of a web form, which is a means of collecting user input. There are various types of form elements that allow users to input different types of data such as text, numbers, dates, options, and files. Here are some of the most common HTML form elements:

Input Element

These are various input types available in HTML forms that can be used to collect different types of input from users:

- **<input type="button">:**
  This creates a clickable button that can be used to perform an action when clicked.
- **<input type="checkbox">:**
  This creates a checkbox that can be checked or unchecked by the user.
- **<input type="color">:**
  This creates a color picker that allows the user to choose a color.
- **<input type="date">:**
  This creates a date picker that allows the user to choose a date.
- **<input type="datetime-local">:**
  This creates a date and time picker that allows the user to choose a specific date and time.
- **<input type="email">:**
  This creates a text box that is specifically designed to collect email addresses.
- **<input type="file">:**
  This creates a file upload control that allows the user to select a file from their local device.
- **<input type="hidden">:**
  This creates a hidden field that is not visible to the user, but can be used to store information that will be sent along with the form submission.

- **<input type="image">:**
  This creates an image button that can be clicked to perform an action.
- **<input type="month">:**
  This creates a month picker that allows the user to choose a specific month.
- **<input type="number">:**
  This creates a text box that is specifically designed to collect numeric input.
- **<input type="password">:**
  This creates a password field that hides the characters entered by the user.
- **<input type="radio">:**
  This creates a radio button that allows the user to choose one option from a list of options.
- **<input type="range">:**
  This creates a slider control that allows the user to select a value within a range.
- **<input type="reset">:**
  This creates a button that can be clicked to reset all form fields to their default values.
- **<input type="search">:**
  This creates a text box that is specifically designed to collect search queries.
- **<input type="submit">:**
  This creates a button that can be clicked to submit the form data to the server.
- **<input type="tel">:**
  This creates a text box that is specifically designed to collect phone numbers.
- **<input type="text">:**
  This creates a simple text box that can be used to collect any type of text input.
- **<input type="time">:**
  This creates a time picker that allows the user to choose a specific time.
- **<input type="url">:**
  This creates a text box that is specifically designed to collect URLs.
- **<input type="week">:**
  This creates a week picker that allows the user to choose a specific week.

Input Element Attributes

1. **type:** Specifies the type of input element.
   Example: <input type="text">.
2. **name:** Specifies the name of the input element.
   Example: <input type="text" name="username">.
3. **value:** Specifies the default value of the input element.
   Example: <input type="text" name="username" value="John">.
4. **required:** Specifies that the input field must be filled out before submitting the form.
   Example: <input type="text" name="username" required>.
5. **placeholder:** Specifies a short hint that describes the expected value of the input field.
   Example: <input type="text" name="username" placeholder="Enter your username">.
6. **disabled:** Specifies that the input field should be disabled.
   Example: <input type="text" name="username" disabled>.
7. **readonly:** Specifies that the input field is read-only.

Example: <input type="text" name="username" value="John" readonly>.

8. **size:** Specifies the width of the input field, in characters.
   Example: <input type="text" name="username" size="20">.
9. **maxlength:** Specifies the maximum number of characters allowed in the input field.
   Example: <input type="text" name="username" maxlength="10">.
10. **min and max:** Specifies the minimum and maximum values allowed in an input field of type number.
    Example: <input type="number" name="age" min="18" max="65">.
11. **multiple:** The multiple attribute is used with the select and input elements to allow users to select multiple options from a list.
    Example: <select multiple>.
12. **pattern:** The pattern attribute is used with the input element to specify a pattern that the user's input must match.
    Example: <input type="text" pattern="[A-Za-z]+" />
    (matches one or more alphabetical characters).
13. **step:** The step attribute is used with the input element of type number or range to specify the increment or decrement of the value.
    Example: <input type="number" step="0.5" />
    (increments or decrements the value by 0.5).
14. **autofocus:** The autofocus attribute is used with the input element to automatically focus on the input field when the page loads.
    Example: <input type="text" autofocus />.
15. **width and height:** The width and height attributes are used with the img element to specify the dimensions of the image in pixels.
    Example: .
16. **list:** The list attribute is used with the input element of type text or search to associate the input with a datalist element that provides a list of predefined options.
    Example: <input type="text" list="options" />
    <datalist id="options">
    <option value="Option 1">
    <option value="Option 2">
    </datalist>
17. **autocomplete:** The autocomplete attribute is used with the form, input, and textarea elements to enable or disable autocomplete for the input fields.
    Example: <input type="text" autocomplete="off" />.

Select Element

```
<select name="country">
  <option value="usa">USA</option>
  <option value="canada">Canada</option>
  <option value="mexico">Mexico</option>
</select>
```

Label Element

The HTML label element is used to associate a label with a form control, such as an input field, a select box, or a textarea. The label element provides a descriptive label for the form control and makes it easier for users to understand the purpose of the control.

The syntax for the label element is as follows:

```
<label for="control_id">Label Text</label>
```

The "for" attribute specifies the ID of the form control to which the label is associated. When the user clicks on the label, the associated form control is focused and the cursor is placed in the control, making it easier for the user to input data.

For example, the following code creates a label for an input field:

```
<label for="username">Username:</label>
<input type="text" id="username" name="username">
```

In this example, the "for" attribute of the label specifies the ID of the input field. When the user clicks on the label "Username:", the cursor is automatically placed in the input field with the ID "username".

Using the label element can improve the accessibility and usability of your web forms. Screen readers and other assistive technologies can use the association between the label and the form control to provide more meaningful feedback to users with disabilities. Additionally, the label element makes it easier for users to select the form control and input data, improving the overall user experience.

Textarea Element

The HTML <textarea> element is used to create a multi-line text input area for a form. This allows users to enter more than one line of text in a form field. The <textarea> element is an empty element, which means that it doesn't need a closing tag.

To use the <textarea> element, you need to specify the name attribute, which is used to identify the input element in the form, and the rows and cols attributes, which specify the size of the text area in terms of rows and columns.

Here is an example of using the <textarea> element:

```
<form>
  <label for="description">Enter your description:</label>
  <textarea id="description" name="description" rows="4" cols="50"></textarea>
</form>
```

In this example, the <textarea> element creates a text area with four rows and 50 columns, and the id and name attributes are both set to "description". The for attribute in the <label> element is used to associate the label with the input element, so that clicking on the label will give focus to the text area.

By default, the <textarea> element does not have any border or padding. However, you can use CSS to style the text area as needed. Here's an example of using CSS to add some styling to the text area:

```
<style>
  textarea {
    width: 100%;
    padding: 12px 20px;
    margin: 8px 0;
    box-sizing: border-box;
    border: 2px solid #ccc;
    border-radius: 4px;
    resize: none;
  }
</style>
```

```
<form>
  <label for="description">Enter your description:</label>
  <textarea id="description" name="description" rows="4" cols="50"></textarea>
</form>
```

In this example, we've added a few CSS rules to the textarea element to give it a border, padding, and some margin. We've also set the width to 100% so that the text area will take up the full width of its container. Finally, we've set resize: none to prevent users from resizing the text area.

Button Element

<button>: The <button> element is used to create a clickable button on a web page. It can be used to perform actions, submit forms, or trigger JavaScript functions. Here's an example:

```
<button type="button" onclick="alert('Hello, world!')">Click me!</button>
```

Fieldset Element

<fieldset>: The <fieldset> element is used to group related form elements together. It can be used to visually group form elements, as well as to provide a caption or legend for the group. Here's an example:

```
<fieldset>
  <legend>Contact Information</legend>
  <label for="name">Name:</label>
```

```
  <input type="text" id="name" name="name"><br>
  <label for="email">Email:</label>
  <input type="email" id="email" name="email"><br>
  <label for="phone">Phone:</label>
  <input type="tel" id="phone" name="phone"><br>
</fieldset>
```

<legend>: The <legend> element is used to provide a caption or title for a <fieldset> element. Here's an example:

```
<fieldset>
  <legend>Contact Information</legend>
  ...
</fieldset>
```

DataList Element

<datalist>: The <datalist> element is used to provide a list of options for an <input> element with the list attribute. It can be used to provide autocomplete suggestions, as well as to restrict the input to a predefined set of values. Here's an example:

```
<label for="fruit">Favorite fruit:</label>
<input type="text" id="fruit" name="fruit" list="fruits">
<datalist id="fruits">
  <option value="Apple">
  <option value="Banana">
  <option value="Cherry">
  <option value="Grape">
  <option value="Orange">
</datalist>
```

<output>: The <output> element is used to display the result of a calculation or form submission. It can be used to display the result of a JavaScript function, or to show the result of a form submission. Here's an example:

```
<form oninput="result.value=parseInt(a.value)+parseInt(b.value)">
  <label for="a">A:</label>
  <input type="number" id="a" name="a"><br>
  <label for="b">B:</label>
  <input type="number" id="b" name="b"><br>
  <label for="result">Result:</label>
  <output id="result" name="result"></output>
</form>
```

Option Element

<option>: The <option> element is used to define an option in a <select> element or a <datalist> element. It can be used to provide a set of choices for the user to select from. Here's an example:

```
<label for="color">Favorite color:</label>
<select id="color" name="color">
  <option value="red">Red</option>
  <option value="green">Green</option>
  <option value="blue">Blue</option>
</select>
```

Outgroup Element

The <optgroup> element is used to group a set of options within a <select> element. It is typically used to organize options into categories or subgroups.

The syntax for the <optgroup> element is:

```
<select>
  <optgroup label="group1">
    <option value="option1">Option 1</option>
    <option value="option2">Option 2</option>
  </optgroup>
  <optgroup label="group2">
    <option value="option3">Option 3</option>
    <option value="option4">Option 4</option>
  </optgroup>
</select>
```

In this example, two option groups are defined: group1 and group2. Each group contains two options. The label attribute is used to specify the label for the group.

The <optgroup> element can also contain a disabled attribute, which prevents the user from selecting any options within that group.

Here is an example that includes a disabled option group:

```
<select>
  <optgroup label="group1">
    <option value="option1">Option 1</option>
    <option value="option2">Option 2</option>
  </optgroup>
  <optgroup label="group2" disabled>
```

```
    <option value="option3">Option 3</option>
    <option value="option4">Option 4</option>
  </optgroup>
</select>
```

In this example, the group2 option group is disabled, so the user cannot select any options within that group.

HTML Media

HTML Media refers to the integration of multimedia elements such as images, videos, and audio into a web page using HTML. HTML provides several media-related elements that allow developers to embed multimedia content into their web pages, including the <img> tag for images, the <video> tag for videos, and the <audio> tag for audio files.

In addition to these tags, HTML also provides several attributes that allow developers to customize and control the media elements, such as the "src" attribute to specify the media file's location, the "autoplay" attribute to automatically play the media, and the "controls" attribute to display playback controls for the media.

HTML Video

HTML provides a video tag <video> to embed videos on a webpage. The video tag supports various attributes to control the playback and appearance of the video. Here are some key points about HTML video:

The **<video> tag** is used to embed videos in HTML documents.
The **source** of the video is defined using the src attribute.
The **controls attribute** displays the default video playback controls (play, pause, volume, etc.).
The **autoplay attribute** automatically plays the video when the page loads.
The **loop attribute** causes the video to loop continuously.
The **poster attribute** displays an image before the video is loaded or played.
The **preload attribute** specifies whether the browser should load the video when the page loads or only when the user clicks the play button.
The <track> tag is used to add subtitles, captions, or descriptions to the video.
The **width and height attributes** define the dimensions of the video display area.
Here's an example of how to embed a video in HTML:

```
<video src="example.mp4" controls width="640" height="360">
  <track kind="subtitles" src="subtitles.vtt" srclang="en" label="English">
</video>
```

In this example, the video source file is "example.mp4". The controls attribute displays the default playback controls, and the width and height attributes define the video display area. The <track> tag adds English subtitles to the video.

HTML Audio

HTML audio is a way to include audio files in web pages. The <audio> element is used to embed sound content in a document. It can be used to play different types of audio files such as MP3, WAV, and OGG. The <source> element is used to specify multiple media resources for a media element, providing the browser with different options for playing the media, depending on its compatibility.

Here are some of the attributes that can be used with the <audio> element:

src: specifies the URL of the audio file
controls: adds playback controls to the audio player
autoplay: makes the audio file automatically start playing when the page loads
loop: makes the audio file repeat continuously
preload: specifies whether the browser should preload the audio file (none, metadata, auto)
muted: specifies whether the audio should be muted by default
Example:

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

In this example, the <audio> element is used with the controls attribute to provide a basic set of controls for the user to play, pause, and stop the audio file. The <source> element is used to provide multiple options for the audio file, so the browser can choose the most suitable format to use. Finally, the "Your browser does not support the audio element." text is displayed in case the browser does not support the <audio> element.

Embed YouTube videos in HTML

To embed a YouTube video in an HTML document, you can use the YouTube embedded player.

Here's an example:

```
<!DOCTYPE html>
<html>
<head>
        <title>YouTube video</title>
</head>
```

```
<body>
        <iframe width="560" height="315"
src="https://www.youtube.com/embed/YOUR_VIDEO_ID" frameborder="0"
allowfullscreen></iframe>
</body>
</html>
```

Replace YOUR_VIDEO_ID with the ID of the YouTube video you want to embed. You can find the ID in the URL of the video, after the v= parameter.

You can also customize the player by adding additional parameters to the YouTube video URL. For example, you can add &autoplay=1 to automatically start the video when the page loads, or & controls=0 to hide the video controls. You can find a full list of parameters in the YouTube Embedded Players and Player Parameters documentation.

HTML ID

In HTML, an id is an attribute that is used to uniquely identify an element. It is typically used to apply styles or scripts to a specific element on a web page. Here's an example of how to use an id attribute in

HTML:

```
<div id="myDiv">This is my div.</div>
```

In the example above, the div element has been given an id of "myDiv". This can be used to target this specific element using CSS or JavaScript.

To target an element with a specific id using CSS, you would use the "#" symbol followed by the id name. Here's an example:

```
#myDiv {
  color: red;
}
```

In this example, the CSS code will target the element with an id of "myDiv" and set its color to red.

HTML Class

In HTML, the class attribute is used to apply a specific style or behavior to a group of elements. Multiple elements can share the same class, and each element can have multiple classes.

To define a class in HTML, we use the class attribute and assign it a unique name. We can then reference this name in our CSS stylesheet to style all elements with that class. Here is an example:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    .example {
      color: blue;
      font-size: 20px;
    }
  </style>
</head>
<body>
  <p class="example">This paragraph has the class "example".</p>
  <p class="example">This one does too.</p>
  <p>This paragraph does not have the class "example".</p>
</body>
</html>
```

In the example above, we have defined a class named "example" and applied it to two <p> elements. Both of these paragraphs will be styled with blue text and a font size of 20 pixels, because they share the "example" class.

Note that we can also assign multiple classes to an element, by separating them with a space:

```
<p class="example another-class">This paragraph has both the "example" and "another-class" classes.</p>
```