HTML TUTORIAL

1. HTML introduction done
2. HTML DOCUMENT STRUCTURE done
3. HTML headings done
4. HTML paragraph done
5. HTML Hyperlinks done
6. HTML File Links done
7. HTML Styles done
8. HTML span done
9. HTML div done
10. HTML section done
11. HTML button done
12. HTML class done
13. HTML id done
14. HTML images
15. HTML icons
16. HTML tables
17. HTML media

HTML INTRODUCTION

Hypertext Markup Language, commonly referred to as HTML, plays a crucial role in creating and organizing information on the internet. All web browsers rely on this fundamental technology to effectively render web pages. An HTML document consists of numerous tags and elements that dictate the organization and visual presentation of various web content, including text, images, hyperlinks, and multimedia

Key aspects of HTML:

1. **Markup Language:** HTML, known as a markup language, is not a programming language. It uses tags and attributes to structure web pages and convey content meaning.
2. **Text and Multimedia:** HTML enables text, images, audio, and videos on web pages. It lets you format text, make headers, lists, and paragraphs, and add various multimedia content.
3. **Hyperlinks:** HTML is crucial for creating hyperlinks, which are pathways that help users move between web pages and websites. This is done with the <a> element.

HTML DOCUMENT STRUCTURE

1. The structure of an HTML document is essential for organizing and defining the elements and content of a web page. An HTML file follows a structured format to ensure compatibility with web browsers and accurate display. Below is an overview of the fundamental components of an HTML document:
2. **Document Type Declaration (DOCTYPE):**This is the first line of an HTML document that indicates the HTML version in use. For instance, in HTML5, it appears like this:
3. **HTML Element:**The HTML file starts with the <html> element. This element acts as the foundation for the whole HTML document, containing all other elements on the page.
4. **Head Section:**The ‘<head>’ section includes ‘<meta>’ data, ‘<title>’,‘<links>’, and other metadata or resource links as needed.
5. **Body Section:**The main content of the web page, including text, images, hyperlinks, forms, and various HTML elements, is found in the '<body>' section. Users can see this content when they visit your webpage.

**Main compiler**

HTML Headings

HTML offers six types of headings, each represented by the tags: <h1>, <h2>, <h3>, <h4>, <h5>, and <h6>. Each tag signifies a different level of section heading.

Here's a simple explanation for each:

<h1>: The biggest and most important heading, usually reserved for the main page title. There should ideally be only one <h1> tag per page.

<h2>: The second most important heading, often used for major section titles within the document.

<h3>: This tag is typically used for subsections within a <h2> section.

<h4>: Used for another level of subsections within a <h3> section or for additional information related to a higher-level heading.

<h5> and <h6>: The smallest headings, are often used for highlighting minor points under larger sections or introducing less significant information.

**Main compiler**

HTML PARAGRAPH

The HTML paragraph tag, <p>, is used to indicate paragraph text on a website. In HTML, a paragraph begins where you put the opening <p> tag and ends where you put the closing </p> tag.

Here's how it works:

Mini compiler

Key points to remember about HTML paragraphs:

Paragraph Spacing: Browsers automatically add some space (margin) before and after each <p> element by default. This makes the text more readable.

Nested Tags: You can include links, emphasized text, or even images within a paragraph. For example, to make text bold within a paragraph, you can use the <strong> tag:

Mini compiler

HTML HYPERLINKS

HTML hyperlinks are like signposts that help you move between web pages or different parts of the same page. They are created using the <a> element (anchor element) and the href attribute (hypertext reference) to specify where they go.

Here's an example of how to create a basic hyperlink:

MINI COMPILER

In this example, when you click "Click here," it takes you to "https://www.example.com."

Hyperlinks can also be used to link to sections within the same page by using the id attribute to identify the target element. Here's an example:

MINI COMPILER

Clicking "Go to Section 1" scrolls the page to the element with id="section1."

You can make a link open in a new tab or window by using the target attribute:

MINI COMPILER

When clicked, "Open in New Tab" opens the link in a new browser tab or window.

HTML FILE LINKS

To connect files like CSS and JavaScript to your HTML, you can use the <link> and <script> elements, respectively. Here's how you can do it:

Linking a CSS file:

Typically, put the CSS link in the <head> section of your HTML document.

Use this code, replacing 'your\_css\_file.css' with the actual path to your CSS file:

Linking a JavaScript file:

Usually, place the JavaScript link just before the closing </body> tag. This ensures that the JavaScript code is loaded after the HTML content is parsed.

Use this code, replacing 'your\_js\_file.js' with the actual path to your JavaScript file:

Main compiler

HTML STYLES

In HTML, you can use CSS (Cascading Style Sheets) to style your webpage elements. CSS helps you manage how elements look, including their color, size, font, layout, and more. There are several methods to apply styles in HTML:

Inline styles: You can set styles directly in HTML elements by using the 'style' attribute.

Main compiler with 3 slides

Internal styles: You can create styles inside the <style> tag located in the <head> section of your HTML document.

External styles: You can make a separate CSS file with the .css extension and connect it to your HTML document using the <link> tag.

HTML SPAN

In HTML, the <span> element is an inline-level container used to group and apply styles or manipulate specific parts of text within a larger content block. It doesn't carry any specific meaning by itself but enables targeting and styling of text for design or scripting purposes.

Here's an example of how you can use the <span> element:

Mini compiler

In the previous example, the word 'blue' is enclosed within a <span> element with an inline style applied, changing the text color to blue. This enables you to style only that particular portion of text differently from the rest of the paragraph.

You can also assign an id or class attribute to a <span> element, allowing you to style it with CSS or manipulate it with JavaScript.

HTML DIV

In HTML, the <div> element, short for 'division,' serves as a block-level container for organizing and grouping content on a web page. Unlike the <span> element, which is an inline-level container, the <div> element is commonly used to create larger content sections or divisions.

Here's how you can utilize the <div> element:

Mini compiler

In the previous example, the <div> element is employed to group the heading <h1> and paragraph <p> elements. It forms a block-level container that can be styled or accessed with CSS and JavaScript.

The <div> element can also be assigned an id or class attribute, just like other HTML elements, enabling you to apply particular styles or select them with CSS and JavaScript.

HTML SECTION

In HTML, the <section> element is a semantic tag used to define a distinct section of content within a web page. It's commonly utilized to group related content together, providing it with a specific meaning or purpose. The <section> element aids in enhancing the organization and structure of the HTML document.

Here's an example of how you can employ the <section> element:

Main compiler

In the example on the right side, the <section> element is utilized to enclose content that pertains to providing information about a company. Inside it, you'll find a heading <h2> element and a paragraph <p> element.

Additionally, you can nest <section> elements within other <section> elements or other semantic elements to create a hierarchical structure.

HTML BUTTON

In HTML, the <button> element is used to create interactive buttons on a web page. These buttons can trigger actions or perform specific functions when clicked. The <button> element is versatile and can be customized with text, icons, or other HTML elements.

Here's an example of how to use the <button> element:

Mini compiler

In the example on the right, a simple <button> element is created with the text "Click Me" displayed on it. The type="button" attribute is added to explicitly specify that it's a button element.

You can also use the <button> element to wrap other HTML content, such as text, images, or icons:

In this example, an <img> element is placed inside the <button> element along with the text "Submit." This allows you to show an icon or an image on the button.

Mini compiler 2

The <button> element can also be used to execute JavaScript functions or navigate to a different page when clicked. This can be accomplished using JavaScript event handlers, like onclick:

HTML CLASS

In HTML, the class attribute helps you add one or more class names to an element. This makes it easy to style it with CSS or find it using JavaScript or CSS selectors. It's like giving elements labels to group them based on their similarities.

Here's an example:

Mini compiler

In the example above, we have a <p> element with a class attribute set to "highlight." This lets you style or add functionality to this specific paragraph using CSS or JavaScript.

You can also assign multiple classes to an element by separating them with spaces:

For instance, the <div> element has two classes: "box" and "large." This way, you can style elements using multiple class names, which makes your styling more flexible and reusable.

In CSS, you target elements with a specific class using the dot notation:

In this CSS example, the .highlight selector is used to style any element with the class "highlight" by making the text yellow.

HTML ID

In HTML, the id attribute gives a special name to an element, making it unique. This unique name helps you identify and control the element with JavaScript or CSS.

Here's an example:

Mini compiler

In the example above, the <h1> element has an id attribute set to "main-heading." This unique name lets you target this heading specifically for styling or other actions.

To target an element with a particular id in CSS, use the hash (#) notation:

Mini compiler

In this CSS example, the #main-heading selector is used to style the element with the id "main-heading" by making its text blue.

HTML image  
  
In HTML, you use the <img> element to put pictures on your webpage. It's a simple tag that doesn't need a closing tag.

Here's how to use it:

Mini compiler

In this example, the src attribute tells the browser where to find the image. The alt attribute provides text to show if the image can't load or for accessibility.

You can also set the image's width and height with the width and height attributes:

Mini compiler

For instance, width="300" and height="200" make it 300 pixels wide and 200 pixels tall. Be careful with this, as it can distort the image.

You can style the <img> element with CSS classes or inline styles:

Mini compiler

In this example, the <img> has a class attribute set to "rounded" for styling.

Depending on the image's location, you can use relative or absolute paths for the src attribute:

Mini compiler

In this example, src uses an absolute path from the website's root.

The <img> element is crucial for adding images to your webpage. It offers options to control the image source, size, alternative text, and appearance.