

HTML Tutorial



HTML Introduction

HTML is easy to learn, even if you don't have any programming experience. With a few simple tags, you can create a basic web page. And as you learn more about HTML, you can create more complex and dynamic pages.

So what are you waiting for? Start learning HTML today! It's the first step to creating your own amazing web pages.

What is HTML?

- HTML is an abbreviation for HyperText Markup Language, which functions as a markup language for crafting web pages.
- Its purpose is to define the layout and components found on a webpage, including headings, paragraphs, lists, and images.

Example

Code:- <https://pastebin.com/70JRsMLv>

Output



This is my first HTML page!

I'm so excited to learn HTML!

```
<html>
```

```
  <head>
```

```
    <title>Page title</title>
```

```
  </head>
```

```
<body>
```

```
    <h1>This is a heading</h1>
```

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

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

```
</body>
```

```
</html>
```

- HTML editors are software applications that are used to create and edit HTML code. They provide a variety of features that make it easier to write and debug HTML code, such as syntax highlighting, auto-completion, and error checking.

- There are many different HTML editors available, both free and commercial. Some popular HTML editors include Notepad++, Sublime Text, and Atom.
- The best HTML editor for you will depend on your individual needs and preferences. If you are just starting out with HTML, you may want to use a simple text editor like Notepad++. If you are more experienced, you may want to use a more powerful editor like Sublime Text or Atom.
- No matter which HTML editor you choose, it is a valuable tool for anyone who wants to create and edit HTML code.

- They can help you write more accurate and efficient code.
- They can make it easier to debug your code.
- They can provide you with a variety of features that can help you to improve your productivity.

HTML editors are software applications that are used to create and edit HTML code. They provide a variety of features that make it easier to write and debug HTML code, such as syntax highlighting, auto-completion, and error checking.

There are many different HTML editors available, both free and commercial. Some popular HTML editors include:

- Notepad++: A free and open-source text editor with syntax highlighting for HTML, CSS, and JavaScript.



- Sublime Text: A proprietary text editor with a large number of features, including syntax highlighting, auto-completion, and snippets.



- Brackets: A free and open-source text editor with a focus on web development.



- Visual Studio Code: A free and open-source code editor from Microsoft with a wide range of features, including syntax highlighting, auto-completion, and debugging.



Visual Studio Code

The best HTML editor for you will depend on your individual needs and preferences. If you are just starting out with HTML, you may want to use a simple text editor like Notepad++. If you are more experienced, you may want to use a more powerful editor like Sublime Text or Atom.

HTML Structure: An HTML document has a specific structure. It starts with a `<!DOCTYPE html>` declaration to indicate that it's an HTML5 document. HTML elements are the building blocks of a web page. They define the structure and content of the page. Each HTML element has a specific purpose and function.

The entire content of the page is wrapped in the `<html>` element, which contains two main sections: `<head>` and `<body>`.

- `<html>`: The root element of an HTML document. All other elements are nested within it.
- `<head>`: Contains meta-information about the web page, such as the page title, links to external resources (CSS, JavaScript), and character encoding declarations.
- `<title>`: Sets the title of the web page, which appears in the browser's title bar or tab.
- `<body>`: Contains the visible content of the web page, such as text, images, links, and multimedia.

Example

Code:- <https://pastebin.com/AenpdcNY>

HTML Basic

HTML is a text-based language, which means that it is made up of text and tags. Tags are used to mark up the text in a way that tells the browser how to display it. For example, the `<h1>` tag is used to mark up a heading, and the `<p>` tag is used to mark up a paragraph.

HTML Elements

- `<h1>`, `<h2>`, ..., `<h6>`: Heading elements used for defining different levels of headings on the page, with `<h1>` being the most important and `<h6>` the least important.
- `<p>`: Represents a paragraph of text.

- <a>: Creates a hyperlink to another web page or resource. The href attribute specifies the URL.
- : This tag is used to insert an image.
-

And we have many more elements.

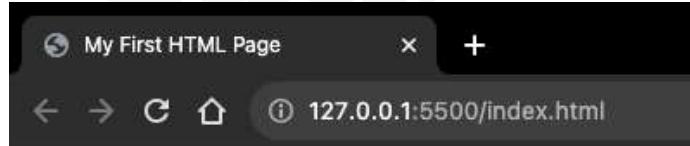
Code:- <https://pastebin.com/auwcqHb3>



HTML elements

HTML elements serve as the foundational components of a web page. These elements are denoted by tags and play a crucial role in determining the page's arrangement and content. Every HTML element serves a distinct purpose and carries out a specific function.

- The HTML page title is the text that appears in the title bar of the browser window and in search engine results pages (SERPs). The page title is also used by search engines to index your web page.
- The page title is defined with the <title> tag. The <title> tag is placed in the <head> section of the HTML document. The content of the <title> tag is the text that will appear in the title bar of the browser window and in SERPs.



HTML Page title

Here we are learning about HTML Page title

The page title serves multiple purposes:

- Browser Identification: It helps users distinguish between different open tabs, making it easier to switch between them.
- Bookmarking: When users bookmark a web page, the title is used as the default name for the bookmark.
- Search Engine Results: The page title is displayed as the link's title in search engine results, significantly impacting the click-through rate.
- Accessibility: Screen readers use the page title to announce the page to users with visual impairments.
- Social Media Sharing: When users share a link on social media platforms, the page title is often used as the default link text.

HTML Headings

HTML headings are used to define the hierarchical structure of a web page's content. They indicate the relative importance and organisation of different sections or headings within the document. HTML offers six levels of headings, ranging from `<h1>` (the highest importance) to `<h6>` (the lowest importance).

Each heading level conveys a specific meaning and should be used appropriately to structure your content. Here's a brief overview of when to use each heading level:

- `<h1>`: This is the most important heading and should be used for the main title of the page. There should typically be only one `<h1>` on a page.
- `<h2>`: Used for major section headings that are a level below the main title.
- `<h3>`, `<h4>`, `<h5>`, `<h6>`: These are used for subsections or subheadings within the major sections. The importance decreases as the heading level number increases.

Example

Output



Main Title

This is some introductory text for our web page.

Section 1

This is the content of section 1.

Section 2

This is the content of section 2.

Subsection 2.1

This is a subsection within section 2.

Subsection 2.2

This is another subsection within section 2.

Sub-subsection 2.2.1

This is a sub-subsection within subsection 2.2.

Sub-subsection 2.2.2

This is a sub-subsection within subsection 2.2.

Sub-subsection 2.2.3

This is a sub-subsection within subsection 2.2.

HTML Paragraphs

- HTML paragraphs are used to group blocks of text together in a structured manner.
- They are represented by the `<p>` element and create a new paragraph each time they are used.
- Paragraphs are essential for organising content and making it easier to read and understand.

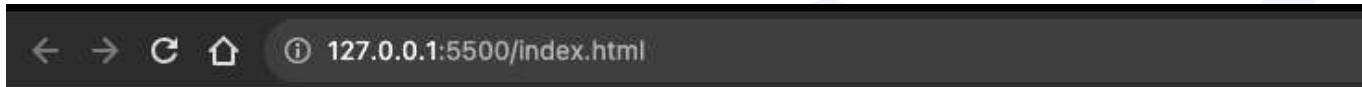
You can use HTML paragraphs to create a clear and organized structure for your web pages. They can also be used to group related content together.

Here are some additional tips for using HTML paragraphs:

- Use paragraphs to separate different sections of content.
- Use paragraphs to group related content together.
- Use consistent paragraph formatting throughout your web page.
- Use paragraphs to make your web pages more readable.

Code:- <https://pastebin.com/hacnQR81>

Output



This is a paragraph with multiple lines of text. This is the second line of text. This is the third line of text.

You can use the `<p>` tag to create paragraphs of text on your web pages. Paragraphs are a great way to organise your content and make it easier to read.

HTML Styles

HTML styles are used to control the appearance of HTML elements. They can be used to change the font, color, size, and other properties of text. They can also be used to add borders, backgrounds, and other effects to elements.

There are two ways to add styles to HTML elements:

- **Inline styles:** Inline styles are added directly to the HTML element. For example, the following code would change the font size of the `<h1>` element to 24px:

Example

Output



This is a paragraph with multiple lines of text. This is the second line of text.

This is a heading

- **Internal Styles**

Internal styles are placed within the `<style>` element inside the `<head>` section of an HTML document. They affect all elements on that page.

Example

Code:- <https://pastebin.com/YedfUxV7>

Output



This is a paragraph with multiple lines of text. This is the second line of text.

This is a heading

- **External styles:** External styles are stored in a separate CSS file. The CSS file is then linked to the HTML document. For example, the following code would link the `styles.css` file to the HTML document:

Example

Code:- <https://pastebin.com/WtN92CMj>

Styles.css

Code:- <https://pastebin.com/gdmnTt7S>

Output



This is a paragraph with multiple lines of text. This is the second line of text.

This is a heading

- **HTML Formatting**

HTML formatting is the process of changing the appearance of text on a web page using HTML tags. There are many different HTML tags that can be used to format text, including tags for bold, italic, underline, and more.

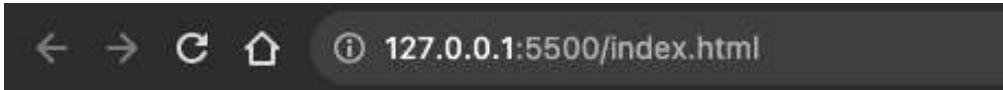
Here are some of the most common HTML formatting tags:

- `` tag: This tag is used to make text bold.
- `<i>` tag: This tag is used to make text italic.
- `<u>` tag: This tag is used to underline text.
- `` tag: This tag is used to make text strong.
- `` tag: This tag is used to make text emphasized.
- `<mark>` tag: This tag is used to highlight text.
- `<sub>` tag: This tag is used to display text as subscript.
- `<sup>` tag: This tag is used to display text as superscript.

Example

Code:- <https://pastebin.com/uGHVQHqP>

Output



It shows the use of Paragraph tag

This text is bold.

This text is important!

This text is italic.

This text is emphasised.

This is some smaller text.

Do not forget to buy **milk** today.

My favourite color is ~~blue~~ red.

My favourite color is ~~blue~~ red.

This is _{subscripted} text.

This is ^{superscripted} text.

These are some of the essential HTML formatting elements. There are many more tags and attributes available to enhance the structure and presentation of web pages.

HTML Quotations

HTML quotations are used to display text that is quoted from another source. There are two main HTML elements used for quotations:

- <q> element: This element is used to define a short quotation. Browsers normally insert quotation marks around the quotation.
- <blockquote> element: This element is used to define a longer quotation. Browsers usually indent <blockquote> elements and add quotation marks around the quotation.

Here is an example of how to use the <q> element to display a short quotation:

Inline Quote: For shorter quotations that can be included within a paragraph, you can use the <q> element. Browsers may apply quotation marks around the inline quote.

Example

Code:-<https://pastebin.com/gqjUTZW6>

Output



Html Inline Quotations

He said, “This is an inline quote.”

Blockquote: The <blockquote> element is used to represent a longer quotation, typically set apart from the surrounding content. Browsers usually indent the blockquote and apply styling to distinguish it from regular text.

Example

Code:-<https://pastebin.com/2fAFds44>

Output



Html blockquote Quotations

This is a longer quotation that should be set apart from the surrounding text.

Author's Name

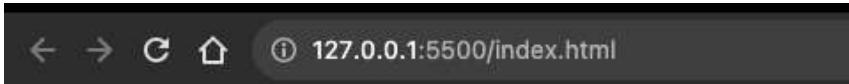
HTML Comments

- HTML comments are used to add notes to HTML code that are not displayed in the browser. Comments can be used to explain the code, provide instructions, or even just to leave a reminder for yourself.
- HTML comments are enclosed in special tags that start with <! -- and end with -->. Everything between the tags is considered a comment and will be ignored by the browser.

Example

Code:-<https://pastebin.com/u42jUgaT>

Output



Welcome to our website!

This is some important content.

© 2023 Our Website. All rights reserved.

In the above example, comments are used to explain each section of the HTML code, and some temporary code is commented out and will not be displayed when the page is viewed in a browser. This can be useful for debugging or making notes about the code structure.

Here are some tips for writing effective HTML comments:

- Keep comments short and to the point.
- Use clear and concise language.
- Explain the purpose of the code, not how it works.
- Use comments to document your code, not to replace it.

HTML Colors

In HTML, colors can be represented using various formats. The most common ways to specify colors are using named colors, hexadecimal values, RGB (Red, Green, Blue) values, and HSL (Hue, Saturation, Lightness) values. Here's an overview of each method:

- **Named Colors:** HTML provides a set of predefined color names that you can use directly.

Example

Code:-<https://pastebin.com/mPMEk9n8>

Output



This text is red.

This text is blue.

This text is green.

Hexadecimal Colors: Hexadecimal values represent colors using a combination of six characters, where each pair of characters represents the intensity of the Red, Green, and Blue components. The format is #RRGGBB, where RR is the red component, GG is the green component, and BB is the blue component. Each component can have values from 00 to FF (in hexadecimal), where 00 is the lowest intensity (0) and FF is the highest intensity (255).

Example

Code:- <https://pastebin.com/c3JtPSkz>

Output



This text is red using hexadecimal color.

This text is green using hexadecimal color.

This text is blue using hexadecimal color.

RGB Colors: RGB values represent colors using three integers, each representing the intensity of the Red, Green, and Blue components. The format is rgb(red, green, blue), where red, green, and blue are integers in the range of 0 to 255.

Example

Code:- <https://pastebin.com/rquaiVVR>

Output



This text is red using RGB color.

This text is green using RGB color.

This text is blue using RGB color.

HSL Colors: HSL values represent colors using three components: Hue (the type of color), Saturation (the vividness of the color), and Lightness (the brightness of the color). The format is hsl(hue, saturation, lightness).

Example

Code:- <https://pastebin.com/t4PxYCHt>

Output



This text is red using HSL color.

This text is green using HSL color.

This text is blue using HSL color.

Remember that colors can be applied to various HTML elements like text (color property), background (background-color property), borders, and more

HTML Links

HTML links are used to create hyperlinks that allow users to move from one web page or resource to another. When a user clicks on a link, they are directed to the URL specified in the link's href attribute. Links are essential for web navigation and help users explore and interact with websites.

Here is a more detailed explanation of each part of the sentence:

- **HTML links:** HTML links are elements in HTML code that create hyperlinks. Hyperlinks are text or images that, when clicked, take the user to another web page or resource.
- **Hyperlinks:** Hyperlinks are a way to navigate between different web pages or resources. When a user clicks on a hyperlink, they are redirected to the URL specified in the hyperlink's href attribute.
- **href attribute:** The href attribute is an attribute of the HTML link element that specifies the URL of the linked resource.
- **URL:** A URL, or Uniform Resource Locator, is a unique address that identifies a web page or resource on the internet.

HTML Attributes

HTML attributes are additional pieces of information that can be added to HTML elements. They provide extra details about an element's behavior, appearance, or functionality. Attributes are specified within the opening tag of an element and are written as name-value pairs, separated by an equals sign (=). The value is usually enclosed in double or single quotes.

Here's an example of an HTML element with attributes:

In this example, the `<a>` element is an anchor (link), and it has two attributes:

- **href:** This attribute defines the URL to which the link points. When users click on the link, they will be redirected to the specified URL ("https://www.google.com" in this case).
- **target:** This attribute specifies how the linked page should open. In this case, the value `_blank` indicates that the link should open in a new browser tab or window.

Here's how to create HTML links:

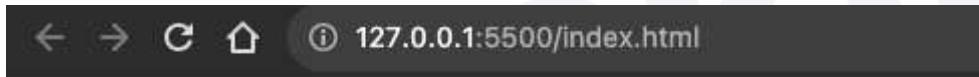
In this syntax:

- **href:** This attribute specifies the URL or destination to which the link should point. It can be an absolute URL (e.g., "<https://www.example.com>") or a relative URL (e.g., "page.html" or "../other_folder/page.html").
- **Link Text:** This is the visible text that appears on the webpage and serves as the clickable part of the link. When users click on this text, they are taken to the URL specified in the href attribute.

Example

Code:- <https://pastebin.com/h0EXXCwg>

Output



Use of links in html

[Visit Google Website](#)

HTML Images

HTML uses the `` tag to add images to webpages. Images can be used to enhance the visual appeal of a website and are commonly used to display photos, illustrations, logos, and other graphics.

Here are the steps on how to use the `` tag to add images to your HTML page:

Example

Code:- <https://pastebin.com/YhZBty89>

Output



My Webpage

Here is an image:



Make sure to specify the correct path to the image file and use descriptive alt text for better accessibility. Additionally, consider optimizing your images for the web to improve page loading times. You can use image editing tools or online services to resize and compress images without compromising their quality.

HTML file paths

HTML file paths are used to specify the location of external resources, such as images, CSS files, JavaScript files, and other assets, on a web page. The browser uses the file path to find the resource and load it into the web page.

There are three main types of file paths used in HTML:

- **Absolute File Paths:** An absolute file path provides the full URL or file system path from the root of the web server or local file system. Absolute paths start with the scheme (e.g., "http://" for web URLs or "file://" for local file paths).

Example

```

```

- **Relative File Paths:** A relative file path provides the location of the resource relative to the current HTML file or the current working directory. Relative paths are commonly used when referencing resources within the same website or project.

Example

```
<link rel="stylesheet" href="/css/styles.css">
```

Let's take an example to create a webpage to show the use of file path:

Code:- <https://pastebin.com/GXDmPxLP>

Output



HTML Favicon

A favicon (short for "favourite icon") is a small icon that is displayed in the browser tab and in the bookmarks bar. Favicons are a great way to brand your website and make it easier for users to identify your site. To add a favicon to your HTML document, you need to include the following code in the `<head>` section of your document:

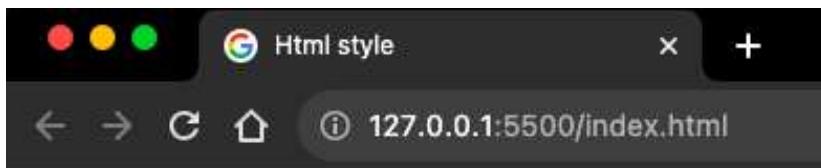
The `href` attribute of the `<link>` tag specifies the location of the favicon file. The favicon file should be a small image file, such as a .ico file.

You can also use the `rel="icon"` attribute instead of the `rel="shortcut icon"` attribute. The `rel="icon"` attribute is more recent and is supported by most browsers.

Example

Code:- <https://pastebin.com/2TZPghWA>

Output



favicon

Here we are using favicons

Having a **favicon** is a small yet important detail that contributes to the overall branding and user experience of your website. It adds a touch of professionalism and recognition, making your site more memorable to users.

HTML tables

- HTML tables are used to display data in a tabular format. They are made up of rows and columns, and each cell in the table can contain text, images, or other elements.
- HTML tables are defined with the `<table>` tag. The `<table>` tag is placed in the `<body>` section of the HTML document. The `<table>` tag has a number of attributes that can be used to control the appearance of the table, such as the width, height, and border.
- The `<table>` tag can contain a number of other tags, including the `<tr>` tag, the `<th>` tag, and the `<td>` tag.
- The `<tr>` tag defines a row in the table. The `<th>` tag defines a header cell in the table. The `<td>` tag defines a data cell in the table.

Example

Code:- <https://pastebin.com/8AHvBn8r>

Output



html tables

Name	Age	Occupation
John Doe	30	Software Engineer
Jane Doe	25	Teacher

- `<table>`: This is the container element for the entire table.
- `<tr>`: This represents a table row, which contains table cells (either header cells or data cells). Each row of the table starts with this element.
- `<th>`: This is used to create header cells. It provides bold and centered text by default.
- `<td>`: This represents data cells. It contains the actual data to be displayed in the table.

Tables can have multiple rows (`<tr>`) and columns (`<th>` and `<td>`). You can create complex tables by merging cells, adding headers, and applying different styling through CSS.

HTML Lists

HTML lists are used to display a series of items in a logical order.

There are three main types of HTML lists:

- Ordered lists (`` tags) are used to display a list of items in a numbered order.
- Unordered lists (`` tags) are used to display a list of items in a bulleted list.
- Definition lists (`<dl>` tags) are used to display a list of terms and their definitions.

Each list item is enclosed in a `` tag. The `` tag can contain any type of HTML content, including text, images, and other elements.

Let's understand the Types of lists:

- **Unordered List (``):** An unordered list is a list of items where the order doesn't matter, and the list items are typically represented with bullet points.
- **Ordered List (``):** An ordered list is a list of items that are sequentially numbered. The order matters, and the list items are automatically numbered in ascending order.
- **Description List (`<dl>`):** A description list consists of a list of terms and their corresponding descriptions. It uses the `<dt>` tag for the term and the `<dd>` tag for the description.



Html Lists

Use of Unordered List

- Item 1
- Item 2
- Item 3

Use of Ordered List

1. First Item
2. Second Item
3. Third Item

Use of Description List

Term 1	Description of Term 1
Term 2	Description of Term 2

HTML lists are a versatile way to display information on a web page. They can be used to present a series of items in a logical order, or to provide definitions of terms.

HTML Iframes

In HTML, iframes are used to embed another HTML document within the current document. This allows you to display content from other websites or resources directly on your webpage. The content displayed in an iframe behaves like an independent document, and users can interact with it as if it were loaded in a separate browser window.

Here's the basic syntax for using iframes:

<iframe src="URL"></iframe>

Here's an example of how to use an iframe to embed a Google Map in a webpage:

Example

Code:- <https://pastebin.com/iATDPq7F>

Output



Embedded Google Map



In the example above, we used an iframe to embed a Google Map by providing the Google Maps embed URL in the src attribute. When the page is loaded, the Google Map will be displayed within the iframe.

HTML Block & Inline

In HTML, elements can be categorized as either "block-level" or "inline" elements based on their default behaviour and the way they interact with other elements.

- **Block-level Elements:** Block-level elements are HTML elements that create a block-level box on the web page. This means they typically start on a new line and take up the full available width of their parent container. Block-level elements are often used to create major structural components of a webpage, such as headings, paragraphs, divs, lists, and sections.

Example of block-level elements include:

Code:- <https://pastebin.com/MjmNAPqK>

Output



This is a block-level element.

This is a block-level heading.

This is a block-level paragraph.

- Item 1
- Item 2

This is a block-level section.

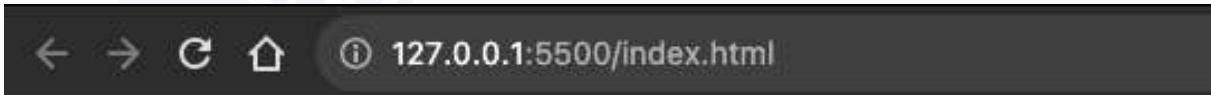
By default, block-level elements have a width of 100% of their parent container. They will stretch horizontally to fill the entire available width, pushing other elements to the next line.

- **Inline Elements:** Inline elements are HTML elements that do not create a new block-level box. They typically flow within the content and do not force a new line. Inline elements are used to style or wrap smaller parts of the content within block-level elements.

Example of inline elements include:

Code:- <https://pastebin.com/gtJiVg3D>

Output



This is an inline element. [This is an inline link.](#)

This is inline text with strong emphasis. *This is inline text with emphasis.*

Inline elements only take up the space they need, so multiple inline elements can appear on the same line. They are commonly used for styling text or adding inline links to the content.

HTML Forms

An HTML form is a way to collect user input from a web page. Forms are used for a variety of purposes, such as collecting contact information, registering for a website, or submitting an order.

The `<form>` element is used to create an HTML form. The `<form>` element has a few attributes that you can use to configure the form, such as the `action` attribute, which specifies where the form data will be sent, and the `method` attribute, which specifies how the form data will be sent.

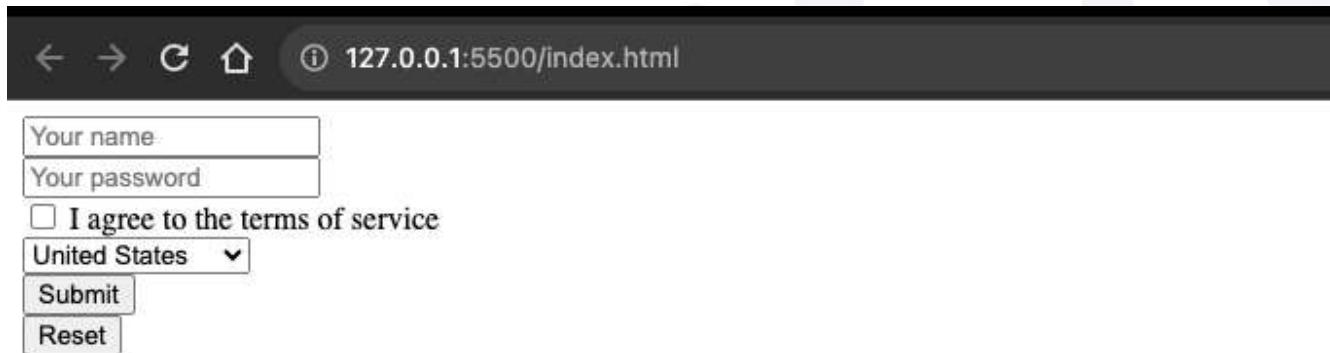
There are a variety of input elements that you can use to collect user input in a form. Some of the most common input elements include:

- **Text input:** This is used to collect text input from the user.
- **Password input:** This is used to collect password input from the user.
- **Checkbox:** This is used to collect a binary value from the user.
- **Radio button:** This is used to collect a single value from the user from a set of options.
- **Select:** This is used to collect a value from the user from a list of options.
- **TextArea:** This element is used to create a text area for the user to enter multiple lines of text.
- **File Input:** Enables users to upload files from their local system.
- **Submit button:** This is used to submit the form data to the server.
- **Reset button:** This is used to reset the form data to its original state.

Example

Code:- <https://pastebin.com/QzDRcV5C>

Output



The screenshot shows a web browser window with the URL `127.0.0.1:5500/index.html`. The page displays a form with the following fields:

- A text input field labeled "Your name".
- A text input field labeled "Your password".
- A checkbox labeled "I agree to the terms of service".
- A dropdown menu labeled "United States".
- A "Submit" button.
- A "Reset" button.

Here are some common HTML text input types and their descriptions:

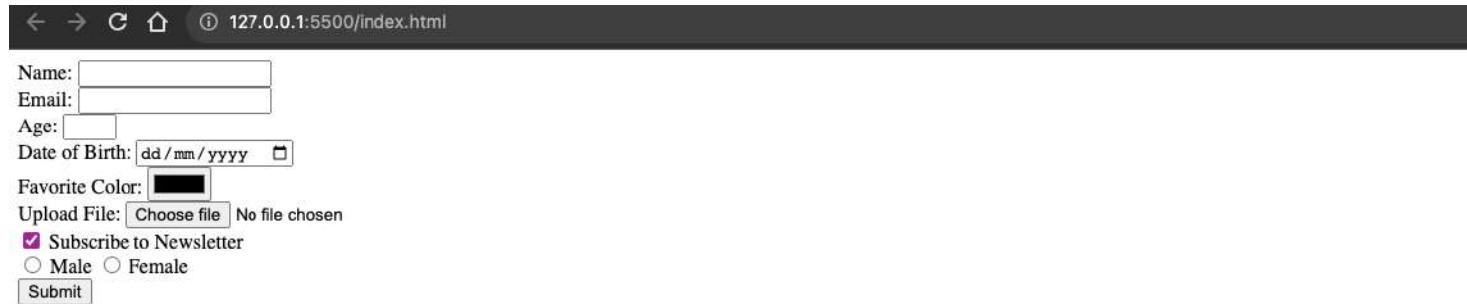
- **Text Input (`type="text"`):** Allows users to enter single-line text, such as names, email addresses, or messages.
- **Password Input (`type="password"`):** Similar to text input but hides the entered characters with asterisks or dots for password security.
- **Email Input (`type="email"`):** Validates that the input is in the form of an email address, including an "@" symbol.
- **Number Input (`type="number"`):** Restricts input to numeric values and can include optional attributes like min, max, and step.

- **Range Input (`type="range"`):** Allows users to select a value from a specified range, typically used with a `<input>` element of type range.
- **Date Input (`type="date"`):** Provides a date picker interface for users to select a date.
- **Time Input (`type="time"`):** Allows users to select a time using a time picker interface.

Example

Code:- <https://pastebin.com/PYkLy5vM>

Output



The screenshot shows a web browser window with the URL 127.0.0.1:5500/index.html. The page displays an HTML form with the following fields:

- Name:
- Email:
- Age:
- Date of Birth: dd / mm / yyyy
- Favorite Color:
- Upload File: Choose file (No file chosen)
- Subscribe to Newsletter
- Male Female
-

HTML Form Attributes

HTML forms have various attributes that allow you to customize their behavior and appearance.

Here are some common HTML form attributes and their purposes:

- **action:** This attribute specifies the URL that the form data will be sent to.
- **method:** This attribute specifies how the form data will be sent. The two most common methods are GET and POST.
- **name:** This attribute specifies the name of the form. This name is used to identify the form when the data is submitted to the server.
- **enctype:** This attribute specifies the MIME type of the form data. The most common MIME type is application/x-www-form-urlencoded.
- **accept:** This attribute specifies the types of files that the form can accept.
- **autocomplete:** This attribute specifies whether or not the form should automatically fill in the user's input.
- **required:** This attribute specifies that the input field must be filled out before the form can be submitted.
- **placeholder:** This attribute specifies a placeholder text that will be displayed in the input field when the field is empty.
- **disabled:** Disables the input element, making it uneditable and non-functional.
- **accept:** Specifies the types of files that can be uploaded using file input. For example, `accept="image/*"` for image files.
- **list:** Associates an `<input>` element with a `<datalist>` element, providing pre-defined options for auto-completion.
- **pattern:** Specifies a regular expression for text input validation.
- **multiple:** Allows users to select multiple files for file input. Used as `multiple="multiple"` or simply `multiple`.

Example

Code:- <https://pastebin.com/AFpzi0si>

Output

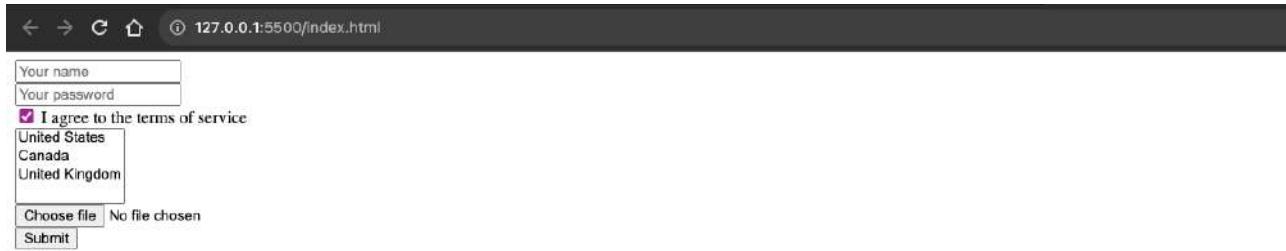


A screenshot of a web browser window showing a form at the URL 127.0.0.1:5500/index.html. The form includes fields for 'Enter your username' (containing '25'), 'Your password' (containing '25'), a checked checkbox for 'Subscribe to Newsletter', a file input field 'Choose files' (showing 'No file chosen'), and a 'Submit' button.

Example

Code:- <https://pastebin.com/RVSNFmFm>

Output



A screenshot of a web browser window showing a form at the URL 127.0.0.1:5500/index.html. The form includes fields for 'Your name' and 'Your password', a checked checkbox for 'I agree to the terms of service', a dropdown menu for 'United States', 'Canada', or 'United Kingdom', a file input field 'Choose file' (showing 'No file chosen'), and a 'Submit' button.

HTML Entities

HTML entities are special codes that are used to represent characters that cannot be typed directly into HTML code. These characters include special symbols, accented letters, and non-breaking spaces.

There are two ways to represent HTML entities:

- Entity names:** These are the names of the characters, such as "&" for the ampersand symbol.
- Entity numbers:** These are the Unicode code points of the characters, such as "&" for the ampersand symbol.

For example, to display the ampersand symbol in HTML, you could use either of the following codes:

&
amp;
&

HTML entities are often used to prevent browsers from interpreting certain characters as HTML tags.

For example, the less-than sign (<) is a reserved character in HTML, so if you want to display it in your text, you need to use the entity code "<". Otherwise, the browser might think that you are trying to start a new tag.

Here are some common HTML entities:

- < : Represents the less than sign (<).
- > : Represents the greater than sign (>).
- & : Represents the ampersand (&) itself.
- ": Represents a double quotation mark (").
- ': Represents a single quotation mark (').

For special characters with diacritical marks or accent symbols, you can use entities like:

- é : Represents the letter "é".

For special characters with diacritical marks or accent symbols, you can use entities like:

- é : Represents the letter "é".
- ñ : Represents the letter "ñ".

HTML Symbols

HTML symbols are special characters that are used in HTML documents. These symbols are used to represent a variety of things, such as mathematical symbols, currency symbols, and punctuation marks.

There are two types of HTML symbols:

- **Character entities:** Character entities are a way of representing special characters in HTML documents. They are written as &#code; followed by the hexadecimal code for the character. For example, the copyright symbol is represented by the character entity ©;

Example

Code:- <https://pastebin.com/xCAaYtcK>

Output

```
← → C ⌄ 127.0.0.1:5500/index.html
Greater than symbol: >
Less than symbol: <
Copyright symbol: ©
Euro currency symbol: €
Degree symbol: °
En dash: –
Em dash: This is an em dash — like this.
```

- **Named character references:** Named character references are a more convenient way of representing special characters in HTML documents. They are written as &name;, where name is the name of the character. For example, the copyright symbol is represented by the named character reference ©.

Example

Code:- <https://pastebin.com/w3s0VGjQ>

Output

```
← → C ⌄ 127.0.0.1:5500/index.html
```

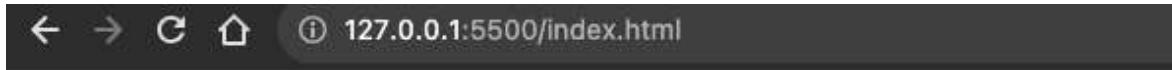
This is a smiling face emoji: 😊

- **Using Named Character Entities:** Some emojis have named character entities that can be used directly in HTML. For example, the smiley face emoji (😊) can be represented using the named character entity 😊 or its short form 😊.

Example

Code:- <https://pastebin.com/JfP8gDw9>

Output



Here's a smiley face emoji: 😊 or 😃

Both representations will display the same smiley face emoji 😊.

HTML CSS

HTML and **CSS** are two of the most important languages for creating web pages. HTML is used to structure the content of a web page, while CSS is used to style the content.

HTML stands for HyperText Markup Language. It is a markup language that is used to create web pages. HTML tags are used to define the structure of a web page, such as headings, paragraphs, lists, and images.

CSS stands for **Cascading Style Sheets**. It is a style sheet language that is used to style the content of a web page. CSS rules are used to define the appearance of a web page, such as the font, color, and size of text, the layout of elements, and the background color of a page.

HTML and CSS work together to create web pages that are both functional and visually appealing. HTML provides the structure, while CSS provides the style.

Here are some of the benefits of using HTML and CSS:

- Easier to create and maintain web pages: HTML and CSS are both relatively easy to learn and use. This makes it easy to create and maintain web pages, even if you are not a professional web developer.
- Flexible and scalable: HTML and CSS are both flexible and scalable. This means that they can be used to create a wide variety of web pages, from simple static pages to complex dynamic pages.
- Widely supported: HTML and CSS are widely supported by web browsers. This means that your web pages will look the same regardless of the browser that is used to view them.

CSS can be added to HTML documents in 3 ways:

- **Inline** - by using the style attribute inside HTML elements
- **Internal** - by using a <style> element in the <head> section
- **External** - by using a <link> element to link to an external CSS file

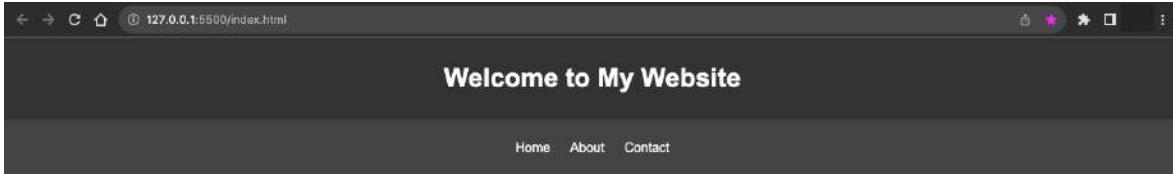
Example

Code:- <https://pastebin.com/hDhJtGQp>

Styles.css

Code:- <https://pastebin.com/5rfPKyCK>

Output



The screenshot shows a browser window with the URL 127.0.0.1:5500/index.html. The page title is "Welcome to My Website". Below the title is a navigation bar with links to "Home", "About", and "Contact".

About Us

We are a team of developers...

Contact Information

Email: contact@example.com

Phone: (123) 456-7890

© 2023 My Website. All rights reserved.

HTML Classes

In HTML, classes are used to group elements together and apply CSS styles or JavaScript functionality to those elements. Classes allow you to target multiple elements on the page and style or manipulate them collectively. Each HTML element can have one or more class names assigned to it using the class attribute.

The syntax for adding a class to an element is as follows:

```
<tagname class="class-name1 class-name2 ...">Content</tagname>
```

In this syntax:

- <tagname> is the HTML element you want to add a class to, such as <div>, <p>, <h1>, <a>, etc.
- class is the attribute, and "class-name1 class-name2 ..." is the value of the attribute. Multiple class names can be separated by spaces.

Example

Code:-<https://pastebin.com/e8r7FyVK>

Styles.css

Code:-<https://pastebin.com/3In5cXHT>

Output



The screenshot shows a browser window with the URL 127.0.0.1:5500/index.html#. The page displays the heading "HTML Classes" and a paragraph explaining the purpose of HTML classes. A blue button labeled "Click Me" is visible at the bottom.

HTML Classes

In HTML, classes are used to group elements together and apply CSS styles or JavaScript functionality to those elements. Classes allow you to target multiple elements on the page and style or manipulate them collectively.

[Click Me](#)

In this CSS, the different class names are used to style the elements. The .container class sets a maximum width and centers the content horizontally, the .heading class applies specific styles to the heading, the .text class styles the text, and the .button class styles the link to look like a button.

Using classes allows you to create consistent and organized styles across your web page and provides a convenient way to apply CSS styles to specific groups of elements.

HTML Id

The id attribute in HTML is used to uniquely identify an element on a web page. Unlike the class attribute, which can be shared by multiple elements, the id attribute must be unique within the HTML document. The id attribute is primarily used to target and style individual elements using CSS or to interact with them through JavaScript.

The syntax for adding an ID to an element is as follows:

```
<tagname id="unique-id">Content</tagname>
```



Welcome to Our Website

- [Home](#)
- [About](#)
- [Contact](#)

Home Section

This is the home section of the website.

About Section

This is the about section of the website.

Contact Section

This is the contact section of the website.

Remember to use IDs responsibly and only for elements that truly require a unique identifier. IDs should not be used to apply shared styles; for that purpose, classes are more suitable.

HTML JavaScript

In html, JavaScript is a widely used programming language that can be used to add dynamic and interactive features to web pages. It allows you to change the content of the page, respond to user input, handle events, and communicate with the server without reloading the page. JavaScript is a client-side scripting language, which means that it runs in the user's web browser and interacts with the Document Object Model (DOM) of the web page.

To include JavaScript in an HTML page, you can use the `<script>` tag. There are two main ways to add JavaScript code to your HTML file:

- **Inline JavaScript:** You can directly include JavaScript code within the `<script>` tag inside the HTML document. This approach is suitable for small code snippets, but it is generally not recommended for larger scripts, as it can make the HTML file less maintainable.

Example

Code:- <https://pastebin.com/zjTdeDQE>

Output



The screenshot shows a web browser window with the URL 127.0.0.1:5500/index.html#. The page title is "My Website". Below the title, there is a navigation bar with links to Home, About, and Contact. The main content area contains two sections: "This is my main section" and "This is my second section", each with some descriptive text. At the bottom, there is a copyright notice and a footer section labeled "use of grid" and "use of flexbox", each containing three items labeled Item 1, Item 2, and Item 3.

HTML Responsive

Responsive HTML is a way of creating web pages that can adapt to different screen sizes and resolutions. This means that your web pages will look good on any device, whether it is a desktop computer, laptop, tablet, or smartphone.

There are a few things you can do to make your HTML responsive:

- **Use fluid grids:** Fluid grids are made up of columns that can resize to fit the width of the screen. This will help to ensure that your content is always readable, regardless of the device it is being viewed on.
- **Use media queries:** Media queries allow you to specify how your web pages should be displayed on different screen sizes. For example, you could use a media query to hide a sidebar on small screens.
- **Use CSS flexbox:** CSS flexbox is a layout property that allows you to easily arrange elements on a page in a responsive way.
- **Use Media Queries:** Apply CSS media queries to adjust the layout, styles, and content based on the screen size. Media queries allow you to target specific screen widths or device features:

Example

```
<style>
/* Example media query for screens smaller than 600px */
@media (max-width: 600px) {
  .container {
    padding: 10px;
  }
}
</style>
```

- **Use Viewport Meta Tag:** Include the viewport meta tag in the head section of your HTML document to control how the page is displayed on different devices. The viewport meta tag sets the width of the viewport and enables proper scaling:

Example

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

- Use Flexible Images: Use CSS techniques like max-width: 100%; to ensure that images scale appropriately with the container:

Example

```
<style>
img {
  max-width: 100%;
  height: auto;
}
</style>
```

- **Use Responsive Frameworks:** Utilize CSS frameworks like Bootstrap, Foundation, or Bulma, which offer pre-built responsive components and layout systems.
- **Use Relative Font Sizes:** Use relative font sizes (e.g., em or rem) to ensure that text scales with the container and user settings.
- **Use Hidden Content:** Use CSS to hide or show content based on screen sizes using classes and media queries.

There are many different ways to make your HTML responsive. The best approach for you will depend on the specific needs of your web pages.

Example

Code:- <https://pastebin.com/Z1yYj5RL>

Output



HTML Computercode

- HTML computer code is used to define the structure and content of a web page. It is a markup language that uses tags to define different elements of a web page, such as headings, paragraphs, lists, and images.
- HTML computer code can be used to create a wide variety of web pages, from simple static pages to complex dynamic pages. It is a powerful tool that can be used to create visually appealing and functional web pages.

Here's an example of using the `<code>` element:

Code:- <https://pastebin.com/PSJ6Cf2X>

Output



HTML Code Example

This is a simple HTML code snippet:

```
<p>Hello, World!</p>
```

HTML Semantics

- HTML Semantic is the practice of using HTML elements to convey the meaning of the content on a web page. This can help make pages more accessible to users with disabilities, as well as more understandable to search engines.
- By using the correct HTML elements, developers can also make their pages more maintainable over time.

Here are some key semantic HTML elements and their purposes:

- **<header>**: Represents the header section of a page or a section within a page. It typically contains a logo, navigation menu, and other introductory content.
- **<nav>**: Represents a navigation menu, usually containing links that allow users to navigate between different sections or pages of the website.
- **<main>**: Represents the main content of the web page. It should contain the primary content that is unique to that page.
- **<section>**: Represents a thematic grouping of content within a page. It helps divide content into meaningful chunks.
- **<footer>**: Represents the footer section of a page or a section within a page. It usually contains copyright information, contact details, or links to related pages.
- **<h1>, <h2>, <h3>, <h4>, <h5>, <h6>**: Heading elements represent the hierarchical structure of the content. `<h1>` is used for the main page title, followed by subheadings in decreasing order of importance.
- **<address>**: Represents contact information for the nearest `<article>` or `<body>` element ancestor. It can include information like email, phone number, physical address, etc.
- **<time>**: Represents a specific time or a range of time. It is useful for displaying dates, times, or durations.
- **<figure> and <figcaption>**: Used together to represent figures (images, diagrams, charts) along with their captions.

Here are some of the benefits of using semantic HTML:

- **Accessibility:** Semantic HTML can make web pages more accessible to users with disabilities. This is because screen readers and other assistive technologies can use the meaning of the HTML elements to understand the content of the page.
- **SEO:** Semantic HTML can help improve the search engine optimization (SEO) of web pages. This is because search engines use the meaning of the HTML elements to understand the content of the page and rank it accordingly.
- **Maintainability:** Semantic HTML can make web pages more maintainable over time. This is because the meaning of the content is conveyed by the HTML elements, rather than by the CSS or JavaScript. This makes it easier to change the look and feel of the page without changing the underlying content.

HTML Style Guide

A HTML guide style is a set of rules that help programmers write clean, consistent, and easy-to-read code. Following a style guide can make code more readable for other developers, improve collaboration between teams, and increase the overall quality of web projects.

Below are some key aspects often covered in an HTML style guide:

- **Indentation:** Use consistent indentation, usually with 2 or 4 spaces, to improve code readability and structure. Avoid using tabs for indentation.
- **Lowercase Element and Attribute Names:** Write HTML element and attribute names in lowercase. HTML is case-insensitive, but using lowercase makes the code easier to read and maintain.
- **Closing Tags:** Always include closing tags for all elements, except for self-closing elements like `
` or ``. Additionally, avoid using the closing slash for regular elements, like `<div></div>` instead of `<div/>`.
- **Quotation Marks:** Use double quotation marks ("") for attribute values. For example, `<input type="text" />`.
- **HTML5 doctype:** Use the HTML5 doctype declaration at the beginning of your HTML file: `<!DOCTYPE html>`.
- **Semantic HTML:** Use semantic HTML elements (e.g., `<header>`, `<nav>`, `<main>`, `<footer>`, `<section>`, etc.) to add meaning to the structure of your web page, improving accessibility and SEO.
- **Comments:** Include descriptive and concise comments in your code to explain complex sections, provide context, or help other developers understand your intent.
- **Whitespace:** Use appropriate whitespace between elements and attributes to improve code readability.
- **Avoid Inline Styles:** Minimize the use of inline CSS styles (style attributes) in favor of using external CSS files for better separation of concerns.
- **CSS and JavaScript Separation:** Avoid using inline CSS or JavaScript, and instead, use external files for better maintainability.
- **Attribute Order:** Maintain a consistent order when writing attributes (e.g., class, id, src, alt, etc.) to improve readability.
- **HTML Entity Usage:** Use HTML entities for characters that have special meanings in HTML, as discussed in the previous response.
- **Use ARIA Attributes:** For accessibility purposes, use ARIA attributes where appropriate to enhance the accessibility of interactive elements.
- **Form Accessibility:** Ensure that all form elements have appropriate labels and provide feedback for form validation.
- **Validation and Linting:** Regularly validate your HTML code using tools like the W3C Markup Validation Service, and use linting tools to catch common code style issues.

Example

Code:- <https://pastebin.com/Q7pMQkvN>

Output



My Website

- [Home](#)
- [About](#)
- [Contact](#)

Welcome to our website!

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Contact Information

Email: contact@example.com

Phone: (123) 456-7890

Subscribe to our Newsletter

Email:

© 2023 My Website. All rights reserved.

In this example:

- The code starts with the HTML5 doctype declaration (`<!DOCTYPE html>`).
- Elements and attributes are written in lowercase.
- Proper indentation is used for readability.
- Closing tags are provided for all elements except for self-closing elements.
- Double quotation marks are used for attribute values.
- Semantic HTML elements like `<header>`, `<nav>`, `<main>`, `<section>`, and `<footer>` are used.
- Comments are not shown in this example but would be included for explanatory purposes where needed.
- Whitespace is used judiciously to improve readability.
- External CSS is linked using the `<link>` tag to separate CSS from HTML.
- The form includes appropriate labels and uses the required attribute for email input field validation.
- There are no inline styles or inline JavaScript.

HTML Charset

- An HTML charset is a character encoding scheme that is used to represent text in an HTML document. The charset specifies the set of characters that can be used in the document, as well as the way in which those characters are represented in binary form.
- The most common charset used in HTML is UTF-8. UTF-8 is a variable-width encoding scheme that can represent all of the characters in the Unicode character set.
- Other common charsets include ISO-8859-1, which is a fixed-width encoding scheme that can represent the characters used in Western European languages, and Shift JIS, which is a fixed-width encoding scheme that can represent the characters used in Japanese.
- The charset for an HTML document is specified in the `<head>` element of the document. The `<meta>` tag is used to specify the charset. The `charset` attribute of the `<meta>` tag specifies the name of the charset.

For example, the following code specifies that the charset for the document is UTF-8:

Example

Code:- <https://pastebin.com/vT2Siq2m>

Output

HTML URL Encode

- HTML URL encoding is the process of converting special characters into their corresponding hexadecimal codes so that they can be safely used in a URL. This is necessary because some characters, such as spaces and ampersands, have special meanings in URLs and can cause problems if they are not encoded.
- There are several different ways to encode characters in a URL. One way is to use the `encodeURIComponent()` function in JavaScript. This function will convert all special characters into their hexadecimal codes. For example, the string "hello world" would be encoded as "hello%20world".
- Another way to encode characters in a URL is to use the % character followed by the hexadecimal code for the character. For example, the string "hello world" would be encoded as "hello%20world".
- It is important to note that not all characters need to be encoded in a URL. Only characters that have special meanings in URLs need to be encoded. For example, spaces do not need to be encoded in a URL.

Here is a table of the characters that need to be encoded in a URL:

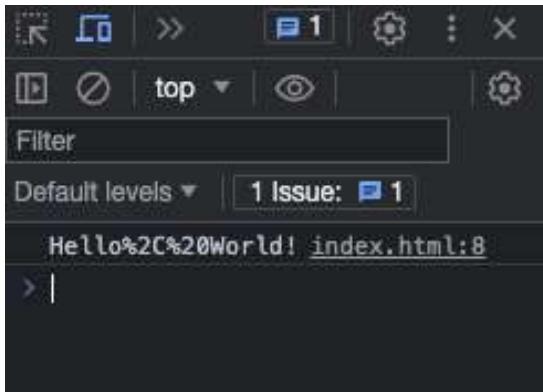
Character	Hexadecimal code
Space	%20
Ampersand	%26
Question mark	%3F
Slash	%2F
Colon	%3A
At symbol	%40
Equal sign	%3D
Left angle bracket	%3C
Right angle bracket	%3E

If you are not sure whether or not a character needs to be encoded in a URL, it is always best to encode it. This will help to ensure that your URL is correctly parsed by browsers.

Example

Code:- <https://pastebin.com/L5cjUdq7>

Output



HTML vs. XHTML

HTML and XHTML are both markup languages used to create web pages. However, there are some key differences between the two languages.

- HTML is a subset of XML. XHTML is a strict subset of XML, which means that XHTML documents must be well-formed and valid. HTML documents, on the other hand, can be less strict.
- HTML tags are case-insensitive. XHTML tags are case-sensitive, which means that the opening and closing tags must be in the same case. HTML tags, on the other hand, can be in either upper or lower case.
- HTML documents must have a DOCTYPE declaration. XHTML documents must have a DOCTYPE declaration that specifies the version of XHTML that is being used. HTML documents, on the other hand, do not need a DOCTYPE declaration.

In general, XHTML is considered to be a more robust and future-proof language than HTML. However, HTML is still widely used and supported by most browsers.

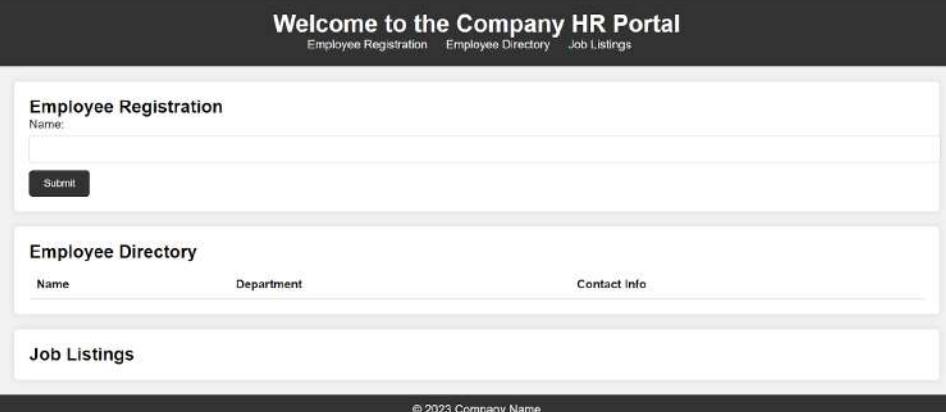
Here is a table that summarizes the key differences between HTML and XHTML:

Feature	HTML	XHTML
Subset of XML	No	Yes
Case-insensitive tags	Yes	No
DOCTYPE declaration	Optional	Required
Robustness	Less robust	More robust
Future-proofing	Less future-proof	More future-proof
Support by browsers	Widely supported	Widely supported

- Ultimately, the best language to use depends on your specific needs. If you need a language that is robust and future-proof, then XHTML is a good choice.
- However, if you need a language that is widely supported by browsers, then HTML is a good choice.

Project Idea :-

HR Portal



The screenshot shows the homepage of a Company HR Portal. At the top, a dark header bar displays the text "Welcome to the Company HR Portal" and links for "Employee Registration", "Employee Directory", and "Job Listings". Below the header, there are three main sections: "Employee Registration" (with a form for Name and a "Submit" button), "Employee Directory" (with fields for Name, Department, and Contact Info), and "Job Listings". At the bottom of the page is a footer bar with the copyright notice "© 2023 Company Name".

Project Introduction:

To build a Company HR Portal, step by step, while learning HTML along the way.

Why Are We Doing This?

The main goal of this project is twofold. First, we're creating a practical web application – an HR Portal. This portal is a valuable tool for businesses to manage employees, job listings, and HR processes efficiently.

The Learning Opportunity:

We're using this project as an opportunity to dive deep into HTML, the fundamental language of web development. HTML, or Hypertext Markup Language, is what powers the structure and content of websites. By building this HR Portal, we'll not only create something useful but also gain a solid understanding of HTML.

Who Is This For?

Whether you're completely new to web development or looking to strengthen your HTML skills, this journey is designed for you. We'll start from scratch, covering HTML basics, and gradually work our way up to more advanced concepts. This project is perfect for students, aspiring developers, or anyone curious about how websites are built.

What to Expect:

Explore HTML elements, create forms, tables and have a fully functional HR Portal that you can customize and expand upon.

1. HTML Headings:

In the <header>, you have an <h1> element with the text "Welcome to the Company HR Portal." This is an example of an HTML heading.

Code:- <https://pastebin.com/yHDpxFB3>

We have already added an `<h1>` element in the header for "Welcome to the Company HR Portal."

Next, we'll proceed to add HTML paragraphs.

2. HTML Paragraphs:

In the `<section id="employee-registration">`, there is a `<p>` element with the text "Register as an employee below." This is an HTML paragraph.

Code:- <https://pastebin.com/7znGey4X>

3. HTML Styles:

Inside the `<head>` section, there is an internal `<style>` element where you can define CSS styles to apply to your HTML elements. In this case, it sets the font family for the entire page to Arial, and it styles `<h1>` elements with a color.

Code:- <https://pastebin.com/LJaAgiqu>

4. HTML Formatting:

Inside the employee registration form, you use the `` element to make the word "Submit" bold for emphasis.

Code:- <https://pastebin.com/EyvGCK5r>

5. Common HTML Formatting Tags:

In addition to ``, you can use other common formatting tags like `` for emphasis, which is used in the job listings section.

Code: <https://pastebin.com/GN5qQxU6>

6. HTML Quotations:

There are no direct quotations in this project, but you can use the `<q>` element for short, inline quotations when needed.

Code: <https://pastebin.com/v63ntkuh>

7. HTML Comments:

Throughout the code, there are comments like `<!-- HTML Tables -->` and `<!-- HTML Lists – Unordered List -->`. These are HTML comments used to explain different sections of the code.

Code: <https://pastebin.com/Cjz43bCK>

8. HTML Colors:

Inside the internal `<style>` element, you set the color of `<h1>` elements with `color: #333;`. This defines the color using a hexadecimal value.

Code:- <https://pastebin.com/DciKLaWE>

9. Introduction to HTML and CSS:

The entire project is an example of HTML and CSS usage. HTML is used for structuring the content, and CSS is used for styling.

Code:- <https://pastebin.com/vCtkZ6Lb>

10. Benefits of Using HTML and CSS:

The benefits of using HTML and CSS are not explicitly mentioned, but you can discuss them during your teaching. Benefits include separation of content and presentation, ease of maintenance, and consistent styling.

Code: <https://pastebin.com/ixBaawFw>

11. Adding CSS to HTML Documents (Inline, Internal, External):

You have used internal CSS by placing `<style>` tags within the `<head>` section. External CSS can be linked using `<link rel="stylesheet" href="styles.css">`, but the actual external stylesheet is not shown in this code.

Code:- <https://pastebin.com/JzNBHd85>

12. HTML Links:

You have navigation links in the `<nav>` section with `<a>` elements pointing to different sections of the page using `href` attributes.

Code: <https://pastebin.com/UmrSgv0j>

13. HTML Images:

There are no images included in this code, but you can add images using the `` tag and `src` attribute when needed.

Code: <https://pastebin.com/Wz0ZQuBq>

We've added an image (replace "company_logo.png" with the actual image file path) inside the `<header>`. The `src` attribute specifies the image file's location, and the `alt` attribute provides alternative text for accessibility.

14. HTML Favicon:

You set the favicon for the webpage using `<link rel="icon" href="favicon.ico">`.

Code: <https://pastebin.com/yGMMcjX1>

You've already implemented a favicon using the `<link>` element with the `rel` attribute set to "icon" and the `href` attribute specifying the path to your favicon file (replace "favicon.ico" with the actual path).

15. HTML Page Title:

You set the title of the webpage using `<title>Company HR Portal</title>`.

Code: <https://pastebin.com/4PdVSpza>

You've set the page title using the `<title>` element within the `<head>` section. This title appears in the browser's title bar or tab.

16. HTML Tables:

You've created an HTML table inside the "Employee Directory" section to display employee information, including name, department, and contact info.

Code:- <https://pastebin.com/aFwgkw3S>

You've created an HTML table for the employee directory, including a header row (`<thead>`) and some sample data rows (`<tbody>`). You can add more rows for additional employees.

17. HTML Lists:

In the "Job Listings" section, you use an unordered list (``) to display a list of job titles as list items (``).

Code: <https://pastebin.com/weHwYWhx>

You've added an unordered list (``) to display job listings. Each list item (``) includes a job title and an "Apply Now" link. You can add more job listings as needed.

18. HTML Block & Inline:

Various HTML elements, like headings (`<h1>`, `<h2>`), paragraphs (`<p>`), and links (`<a>`), demonstrate block and inline-level elements.

Code: <https://pastebin.com/Bac4SmHS>

In this step, you've demonstrated the use of both block-level elements (`<p>`) and inline elements (``, ``, ``, ``) for text formatting and styling. You've also applied inline CSS to change text color.

19. HTML Classes:

Classes are not explicitly used in this code, but you can teach how to apply CSS classes to elements for styling consistency.

Code: <https://pastebin.com/AchVbxqa>

You've added a class (highlight) to one of the paragraphs to demonstrate how HTML classes work. You can define the class in your CSS file for styling.

20. HTML Id:

You've used the id attribute to create unique identifiers for different sections of the page, like <section id="employee-registration">.

Code: <https://pastebin.com/83QyaZPw>

You've added an id attribute to a <section> element. IDs are unique identifiers that can be used for CSS styling or JavaScript interaction. You can add content to this custom section.

21. HTML Iframes:

There are no iframes in this project, but you can introduce iframes as a way to embed external content.

Code: <https://pastebin.com/HsRfF5mq>

You've added an <iframe> element to embed a Google Map into your webpage. You can adjust the src attribute to embed a map of your desired location.

22. HTML JavaScript:

You've included an external JavaScript file using <script src="script.js"></script>.

Code: <https://pastebin.com/eJQBYj8p>

You've added JavaScript code to create an interactive counter. Clicking the "Increment" button will increase the count displayed on the webpage.

23. HTML File Paths:

File paths for stylesheets and scripts are specified in the href and src attributes, respectively.

Code: <https://pastebin.com/GLpWcLbt>

You've demonstrated how to link to a PDF file using a relative file path. Users can click the "Download PDF" link to download the PDF file.

24. HTML Head:

The <head> section contains essential meta-information and links to external resources

Code: <https://pastebin.com/S85nLxqF>

You've added a <meta> element to provide a meta description for your webpage. This can improve SEO and help users understand the page's content.

25. HTML Layout:

The project demonstrates a basic layout structure with header, sections, and a footer.

Code: <https://pastebin.com/Pxbki7JP>

You've structured your webpage content using `<div>` elements for different sections (header, main content, sidebar, and footer). This creates a basic layout for your HR portal.

26. HTML Responsive Design:

While not explicitly shown, you can teach how to make the design responsive using CSS media queries.

Code: <https://pastebin.com/FdWMm08M>

You've added a viewport meta tag to make your webpage responsive. This ensures that your site adjusts appropriately to different screen sizes, such as on mobile devices.

27. HTML Computer Code:

This project doesn't contain computer code examples, but you can include code snippets as needed.

Code: <https://pastebin.com/sgeJkYVB>

You've used the `<code>` element to display a code snippet within your content.

28. HTML Semantics:

Semantic HTML elements like `<header>`, `<nav>`, `<section>`, and `<footer>` are used to give meaning to the structure of the page.

Code: <https://pastebin.com/CPzkAAct>

You've used the `<article>` and `<section>` elements to improve the semantics and structure of your content.

29. HTML Style Guide:

Coding style and formatting are a matter of preference. You can establish your own style guide or follow existing conventions.

Code: <https://pastebin.com/faR0jNHa>

You've added a reference to a style guide within your webpage, making it easy for users to access styling guidelines.

30. HTML Entities, Symbols, and Emojis:

These are not explicitly used in this code, but you can introduce them when needed for special characters or symbols.

Code: <https://pastebin.com/zX2RckQw>

You've included special characters using HTML entities, symbols, and emojis to enhance your content.

31. HTML Charset:

You've set the character encoding for the document with <meta charset="UTF-8">.

Code: <https://pastebin.com/bmnCiH2p>

You've specified the character set (UTF-8) in the <meta> element to ensure proper character encoding for your webpage.

32. HTML URL Encode:

URL encoding is not demonstrated in this code, but you can explain it when discussing links with special characters.

Code: <https://pastebin.com/3jCPeR64>

You've demonstrated URL encoding to include special characters like & in links without causing issues.

34. HTML Forms, Form Attributes, and Input Types:

The "Employee Registration" section demonstrates HTML forms, form attributes (e.g., action, method), and input types (e.g., text, submit).

Code:- <https://pastebin.com/c4jzdAA3>

You've added a contact form to your webpage using the <form> element.

35. HTML Input Attributes:

The <input> element in the form uses the id, name, and required attributes.

Code: <https://pastebin.com/E5gh9b9Y>

You've added various attributes to form elements like placeholder, value, and required to enhance the form's functionality and user experience.

36. HTML Input Form Attributes:

Form attributes like action and method are used in the <form> element.

Code:- <https://pastebin.com/4Yr1JJZd>

You've added various form elements like text inputs, email inputs, password inputs, checkboxes, and range inputs to showcase different input types.

CODE:-

HTML:-

<https://pastebin.com/HcnSzgE6>

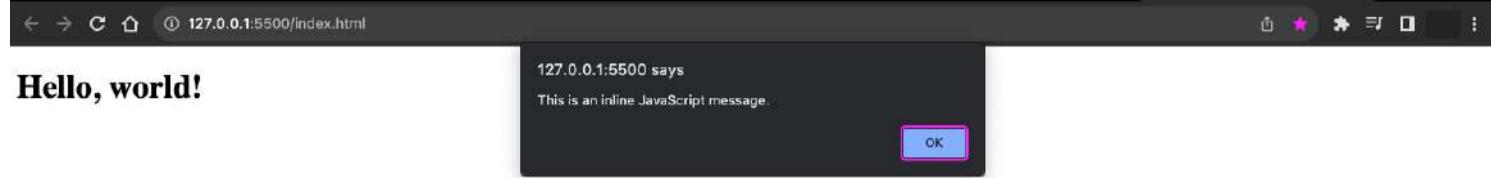
CSS :-

<https://pastebin.com/xhtx55Mw>

Example

Code:- <https://pastebin.com/cyUd76Z4>

Output



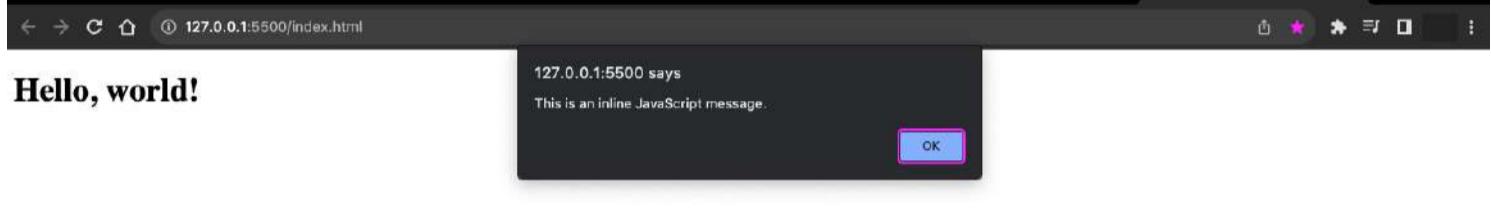
- **External JavaScript File:** For larger JavaScript code and better code organisation, it is preferable to place your JavaScript code in an external file with a .js extension. You can link the external JavaScript file to your HTML page using the <script> tag with the src attribute.

Example

Code:- <https://pastebin.com/GZX8NYCU>

script.js

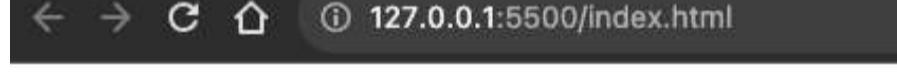
```
// External JavaScript code in script.js
alert("This is an external JavaScript message.");
```



Here's an example of a simple JavaScript function that changes the text color of a heading when a button is clicked:

Code:- <https://pastebin.com/cDZs4ZjM>

Output



Hello, world!

Change Color

When the "Change Color" button is clicked, the JavaScript function changeTextColor() is executed, and it changes the text color of the heading to red.



Hello, world!

Change Color

HTML layout

HTML layout refers to the way in which the content of a web page is organized and structured. It makes the website easy to navigate. For example, As you can see we have various contents on the page like heading, footer, the home page, etc in a structured way.

There are many different ways to layout a web page in HTML. Some of the most common methods include:

- **Using tables:** Tables are a simple way to layout a web page. They can be used to create rows and columns of content.
- **Using floats:** Floats are used to position elements on the page next to each other. They can be used to create a two-column layout or a sidebar layout.
- **Using flexbox:** Flexbox is a newer layout method that is becoming increasingly popular. It is more flexible than tables and floats, and it can be used to create more complex layouts.
- **Using CSS grid:** CSS grid is a newer layout method that is similar to flexbox. It is more powerful than flexbox, and it can be used to create even more complex layouts.

The best way to layout a web page depends on the specific needs of the page. If you need a simple layout, tables may be a good option. If you need a more flexible layout, flexbox or CSS grid may be a better choice.