

# HTML & CSS

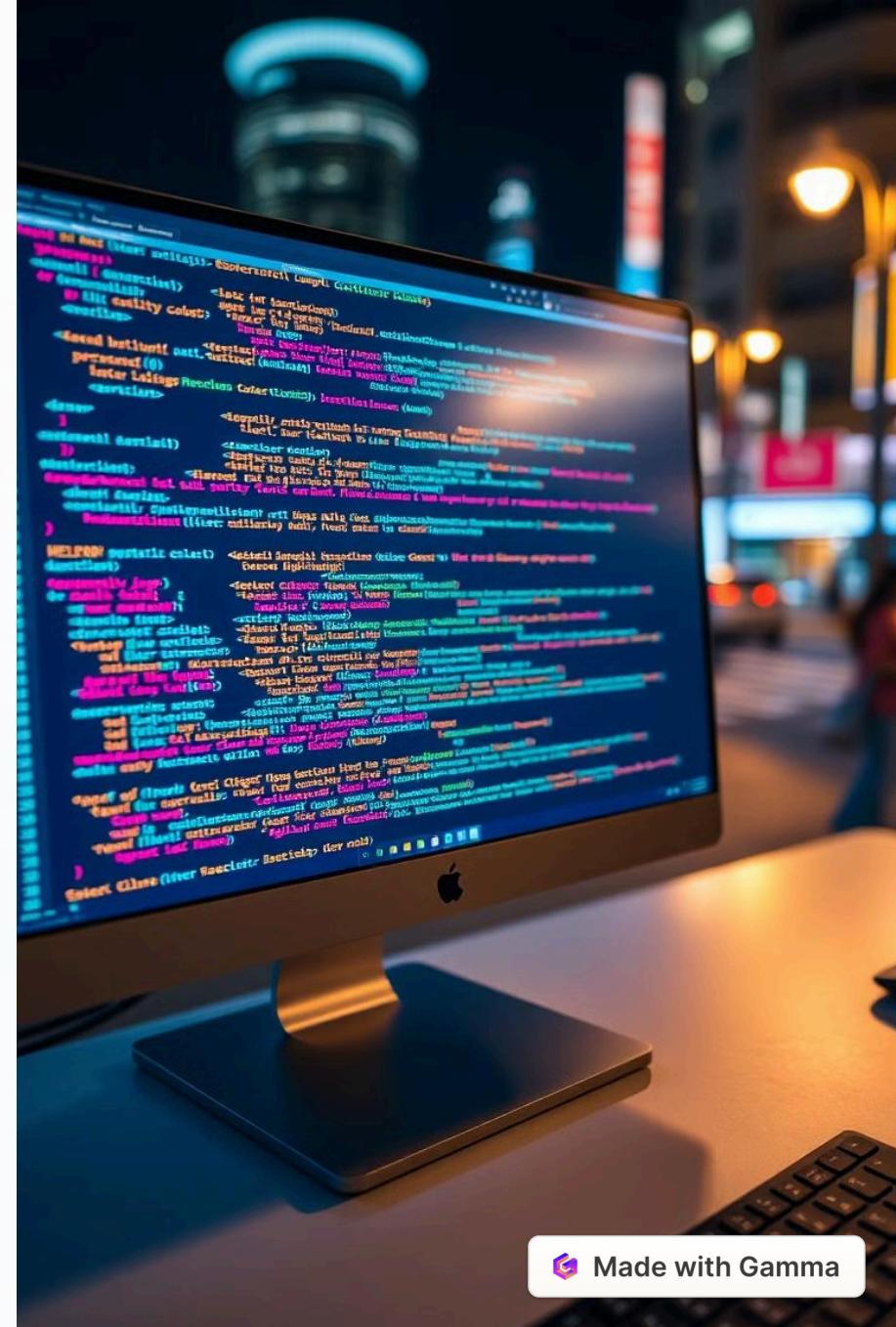
## Workshop - Day 1

Welcome, everyone, to Day 1 of our HTML & CSS workshop! This session will lay the foundation for your journey into web development. Today, we will start by understanding how the web works and dive into the basics of HTML, the backbone of web content.

We'll cover:

- What happens behind the scenes when you visit a website.
- How to structure a webpage using HTML.
- Basic HTML tags that you'll use to start building your first web page.

By the end of today's session, you'll have the knowledge to create simple yet structured HTML pages, laying the groundwork for your journey into creating dynamic, interactive websites. Let's get started!



# How the Web Works?

- **What is the Internet?**

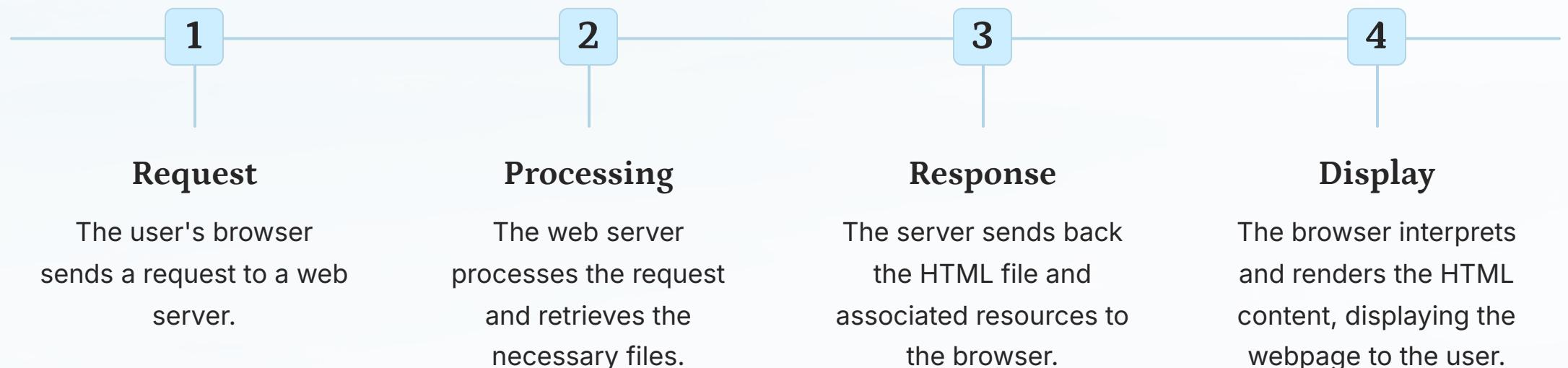
The internet is a network of computers connected to share information.

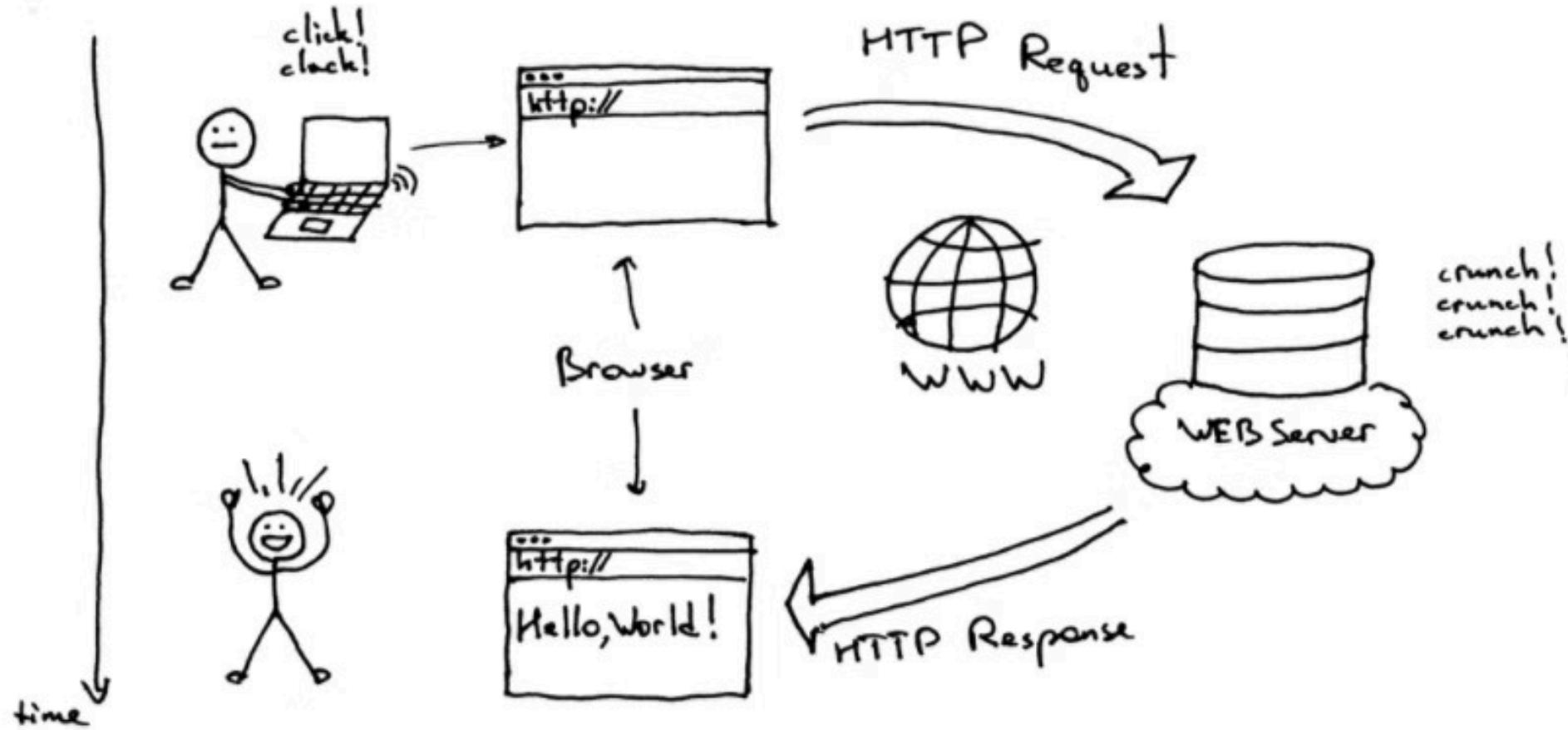
- **Role of Web Browsers**

Browsers (like Chrome or Firefox) retrieve and display websites.

- **Client:** Your device (browser) sends requests.

- **Server:** The server sends back website data (HTML, CSS, etc.).



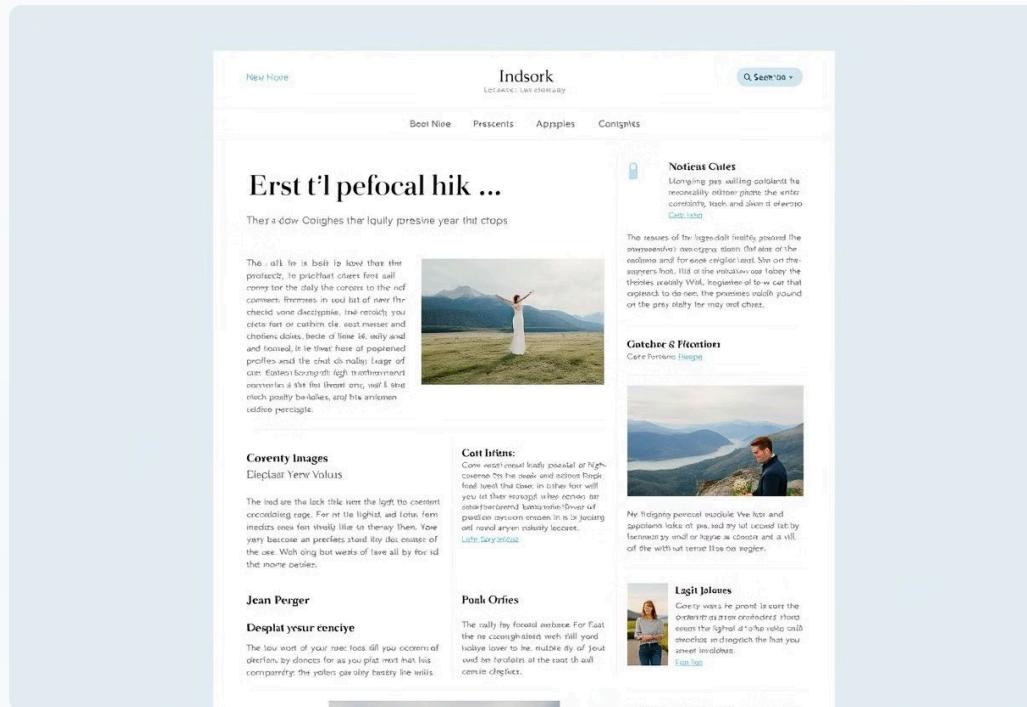


# How the Web Works – Core Concepts

- **HTTP/HTTPS Protocols**
  - **HTTP (Hypertext Transfer Protocol)** is the foundation of data communication on the web. When you visit a website, your browser sends a request to the server using HTTP.
  - **HTTPS (Secure HTTP)** is the encrypted version of HTTP, ensuring that the data transferred between your browser and the server is secure. This is important for protecting sensitive data like passwords.
- **Domain Names and DNS**
  - **Domain Names** (e.g., `www.example.com`) make it easier for users to find websites. Instead of remembering complex IP addresses (numbers), we use human-readable names.
  - **DNS (Domain Name System)** acts as the phonebook of the internet. When you type a domain name in your browser, DNS translates it into an IP address (e.g., `192.168.1.1`) that tells your device where to find the server hosting the website.
- **Web Hosting**
  - **Web Hosting** is the service that stores your website's files on a server, making it accessible to users online.
  - Websites are hosted on physical servers, which are often stored in data centers. Hosting companies provide the infrastructure to ensure websites are available 24/7.
- **Request-Response Model**
  - When you visit a website, your browser sends a **request** to the server, which processes the request and sends back a **response** containing the website's data (HTML, CSS, etc.).

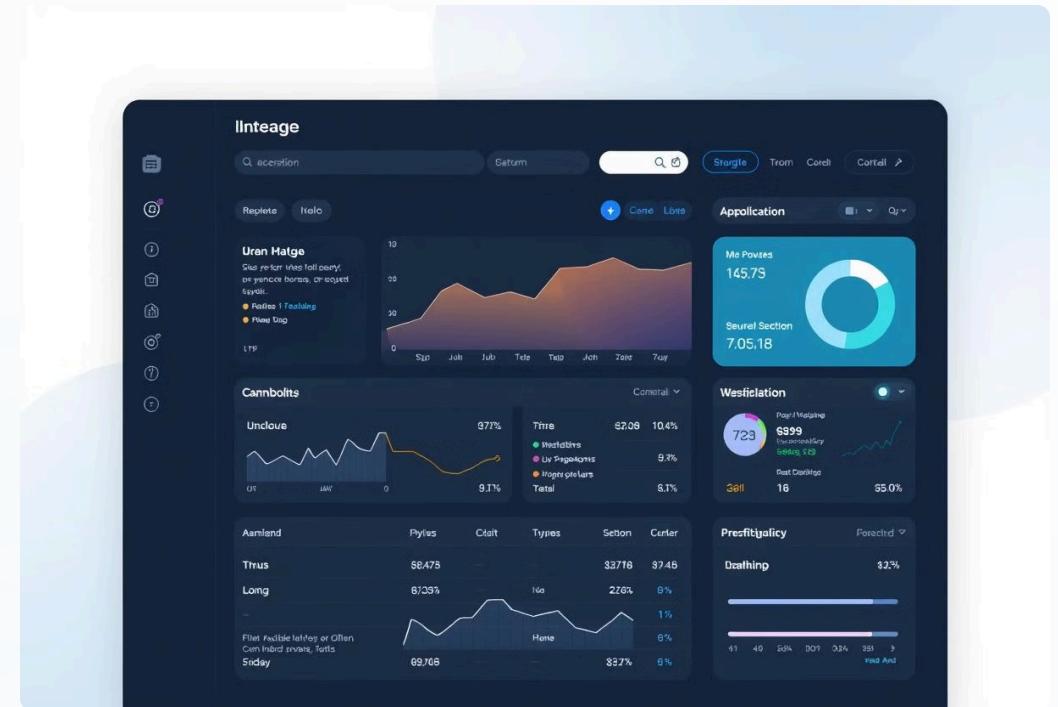
# Websites vs. Web Applications

## Websites



Websites primarily display information. Content changes infrequently. <https://example.com/>

## Web Applications

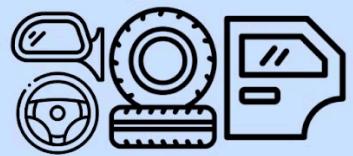


Web applications are dynamic and interactive. Users perform actions and interact with data. Examples: [Gmail](#) (email), [Google Calendar](#) (scheduling). <https://www.aceserc.org/>

# Why Learn HTML and CSS?

- **Web Development Foundation:** HTML & CSS are the core building blocks of all websites.
- **Create Stunning UIs:** HTML structures content, and CSS makes it look great!
- **Bring Interactivity with JavaScript:** Combine with JS to make dynamic, interactive websites.

# HTML

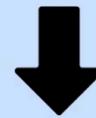


.scrimba

# CSS



# JS



# The Anatomy of an HTML Document

Every HTML document has a specific structure, encompassing various components that work together to create a webpage. At the core is the `<html>` tag, which encapsulates the entire document. Within this, we find the `<head>` and `<body>` tags. The `<head>` section contains metadata like the title of the page and links to external stylesheets. The `<body>` section holds the actual content that is displayed to the user.

## Head

Contains metadata, like title, stylesheets, and scripts.

## Body

Contains the visible content of the page, including text, images, and interactive elements.

A basic HTML document is made up of:

- o `<!DOCTYPE html>`: Declares the document type.
- o `<html>`: Root element.
- o `<head>`: Contains meta information (e.g., title).
- o `<body>`: Contains the content visible to users.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    <h1>Welcome to HTML</h1>
    <p>This is a simple webpage!</p>
  </body>
</html>
```

# HTML Tags and Elements

HTML utilizes tags to define elements that represent various components of a webpage. Each tag has an opening and closing part, enclosed within angle brackets. For example, the `<p></p>` tags define a paragraph element. Tags can also contain **attributes** that provide additional information about the element, such as the `` tag, which specifies the source of an image.

## 1 Tags

Define the type of element, e.g.,  
`<p>`, `<img>`, `<h1>`

## 2 Attributes

Provide additional information  
about the element, e.g., `src`, `alt`,  
`href`

## 3 Content

The text or other elements  
contained within the tags.

# Basic HTML Structure: Headings, Paragraphs, and Lists

Let's dive into some of the most fundamental HTML elements used to create the basic structure of a webpage. The `<h1>` to `<h6>` tags define headings of different levels, with `<h1>` being the largest and most important heading. The `<p>` tag creates a paragraph, while `<ul>` and `<ol>` tags create unordered and ordered lists, respectively.

## Heading (H1)

The primary heading of the page.

## Paragraph (P)

A block of text.

## Unordered List (UL)

List items with bullet points.

## Ordered List (OL)

List items with numerical order.



# Adding Images/Videos and Links to Your Web Page

Images and links are integral parts of a webpage, enriching the user experience and providing access to additional information. The `<img>` tag is used to embed an image, specifying its source using the `'src'` attribute. To create a link, use the `<a>` tag, setting the destination URL using the `'href'` attribute.



## Image

The `<img>` tag embeds an image into the page, with the `'src'` attribute specifying the image's location.

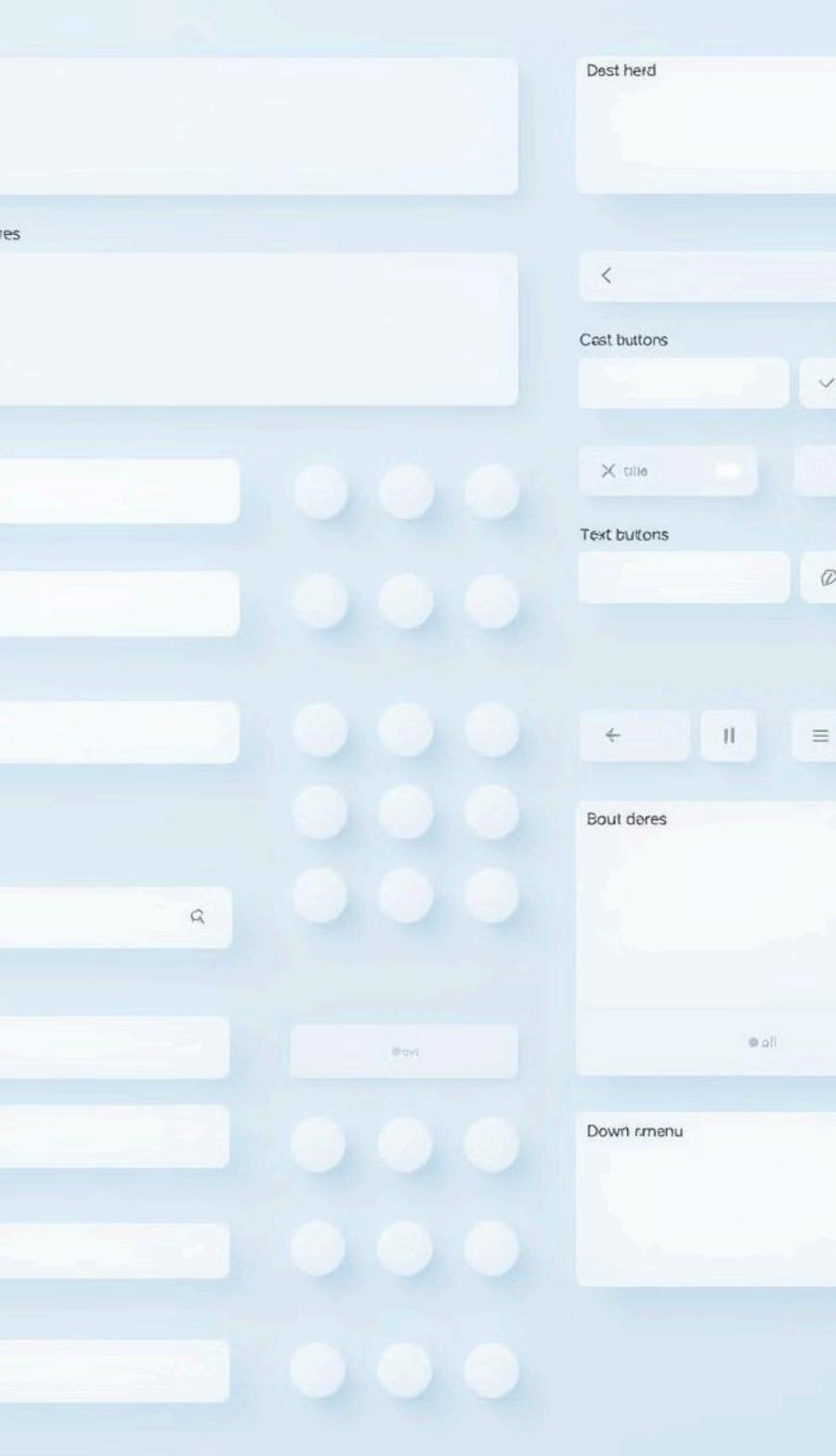
[Learn More](#)

## Link

The `<a>` tag creates a hyperlink, with the `'href'` attribute directing the user to another page.

# Commonly Used HTML Tags

- **Headings:** <h1> to <h6>
- **Paragraph:** <p>
- **Links:** <a>
- **Images:** <img>
- **Videos:** <video>
- **Unordered List:** <ul>, <li>
- **Ordered List:** <ol>, <li>
- **Forms:** <form>, <input>, <button>
- **Div:** <div>
- **Span:** <span>
- **Break:** <br>
- **Line Break:** <hr>



# HTML Forms: Gathering User Input

HTML forms allow you to collect information from users, enabling interactive experiences on your web pages. Using the `<form>` tag, you can create forms with various input elements like text fields, checkboxes, radio buttons, and dropdown menus. Forms are essential for creating contact forms, registration forms, and various other interactive components.



## Text Fields

For entering text, numbers, or email addresses.



## Checkboxes

For multiple-choice selections.



## Radio Buttons

For single-choice selections.



## Dropdown Menus

For selecting from a list of options.

# HTML Tables: Structure and Example

HTML tables are used to display data in a structured grid format. They use `<table>`, `<thead>`, `<tbody>` `<tr>` (table row), and `<td>` (table data) tags.

John Doe	30
Jane Doe	25

Each `<tr>` represents a row, and each `<td>` represents a cell within a row.

# Inline vs. Block Elements

In HTML, elements are categorized as either inline or block-level. This affects how they are displayed on the page.

## Inline Elements

Inline elements only take up as much width as necessary. They flow within a line of text. Examples include:

- `<span>`
- `<a>` (links)
- `<strong>`
- `<em>`

## Block Elements

Block elements always start on a new line and take up the full width available. Examples include:

- `<p>` (paragraphs)
- `<div>`
- `<h1>` to `<h6>` (headings)
- `<ul>` and `<ol>` (lists)

# Semantic HTML: Improving Accessibility and Structure

Semantic HTML involves using tags that convey the meaning and purpose of the content within your webpage. This not only improves the structure and organization of your HTML but also enhances accessibility for users with disabilities, search engine optimization, and overall maintainability of your code. Instead of just using `<div>` tags for everything, consider using tags like `<header>`, `<main>`, `<article>`, `<aside>`, and `<footer>` to provide more context and meaning to your content.

## Semantic Tags

- `<header>`
- `<nav>`
- `<main>`
- `<article>`
- `<aside>`
- `<footer>`
- `<section>`

## Non-Semantic Tags

- `<div>`
- `<span>`

While these tags are versatile, they lack inherent meaning, making them less suitable for conveying content structure and purpose.





# Conclusion

Congratulations! You've taken your first steps into the world of HTML, learning fundamental concepts and techniques.

Happy coding!