

# HTML Basics

## Learning Objective

### Introduction

HTML, or Hypertext Markup Language, is the backbone of the web. It is a markup language that structures the content of web pages, defining the elements, layout, and formatting.

Focus: Basics of HTML, Different HTML Tags, and Attributes

Prerequisites: VS Code IDE

### Theme

Airbnb is a popular online marketplace for booking accommodations and experiences worldwide. HTML plays a crucial role in shaping the user interface and functionality of Airbnb's website. From displaying property listings, interactive search filters, and booking calendars to user reviews, host profiles, and secure payment options, HTML is instrumental in structuring and presenting the vast amount of information on the platform.

By leveraging HTML effectively, Airbnb ensures a visually appealing and user-friendly experience for individuals seeking accommodations or hosting their properties. HTML's flexibility allows Airbnb to cater to diverse user needs and provide a seamless browsing and booking process for travelers around the globe.

### Primary Goals

- Gain a solid understanding of HTML, its purpose, and how it is used to structure web pages.
- Become familiar with fundamental HTML tags like `<!DOCTYPE html>`, `<html>`, `<head>`, `<body>` etc.
- Apply the concepts learned in each step to implement a project, gradually building a webpage based on the provided project implementation steps.
- Learn how to use appropriate HTML tags to structure different parts of a webpage, such as the header, main content, sections, and footer.

- Explore additional HTML elements and features like dropdown menus to enhance the user experience and interactivity of the webpage.

# HTML Basics

## Introduction to HTML

HTML stands for Hypertext Markup Language. It is the standard markup language used for creating web pages and applications. HTML provides a structured way to organize and format the content of a webpage, including text, images, videos, links, forms, and more.

HTML is composed of a series of elements or tags, which are enclosed within angle brackets (< and >). These tags define the structure and semantics of the content. For example, the <h1> tag represents a heading, the <p> tag represents a paragraph, and the <img> tag represents an image.

Here's a simple example of an HTML document structure:

```
<!DOCTYPE html>
<html>
<head>
  <title>My First Web Page</title>
</head>
<body>
  <h1>Welcome to My Web Page</h1>
  <p>This is a paragraph of text.</p>
  
  <a href="https://www.example.com">Visit Example.com</a>
</body>
</html>
```

In this example, we have an HTML document enclosed within the <html> tag. The document is divided into two main sections: the <head> section, which contains meta-information about the document, such as the title displayed in the browser's title bar, and the <body> section, which contains the visible content of the webpage.

Inside the <body> section, we have various HTML elements. The <h1> tag represents a heading, the <p> tag represents a paragraph, the <img> tag displays an image, and the <a> tag creates a hyperlink to another web page. We will study them thoroughly.

HTML is a markup language, meaning it uses tags to structure and format the content, but it does not directly control the visual appearance of the webpage. The presentation

and styling of HTML elements are typically handled using CSS (Cascading Style Sheets) and JavaScript for interactivity.

By combining HTML, CSS, and JavaScript, web developers can create interactive and visually appealing websites and web applications that can be accessed and displayed in web browsers on various devices. HTML is a fundamental building block of the web and serves as the backbone for creating web pages and delivering content on the internet.

## HTML `<!DOCTYPE html>` Declaration

Every HTML document should start with a `<!DOCTYPE>` declaration.

So as you know, this declaration isn't an HTML tag. It's more like a heads-up to the browser, letting it know what type of document to expect.

In HTML 5, the declaration is simple:

```
<!DOCTYPE html>
```

```
<!-- HTML elements go here -->
```

## Comments in HTML

In HTML, you can add comments to your code to provide explanations, notes, or reminders for yourself or other developers. Comments are not displayed in the browser when the page is rendered and are ignored by the HTML parser. They are purely for human readability. Here's how you can add comments in HTML:

- **Single-line comments:** To add a comment that spans a single line, you can use the `<!-- -->` syntax. Anything between these opening and closing comment tags will be treated as a comment.
  - Here's an example:

```
<!-- This is a single-line comment -->
```

- **Multi-line comments:** If you want to add comments that span multiple lines, you can use a combination of `<!--` at the beginning and `-->` at the end.
  - Here's an example:

```
<!--  
This is a multi-line comment.  
It can span multiple lines.  
-->
```

Note

# <html> tag

The `<html>` tag is the root element of an HTML document. It holds everything together and represents the whole web page. Remember to close it with `</html>` to keep things in order.

Here's the deal with the `<html>` tag:

- **Nesting:** All the other HTML elements, like `<head>` and `<body>`, hang out inside the `<html>` tag. The `<head>` usually comes right after the `<html>` tag, and the `<body>` follows the `<head>`.
- **Structure Power:** Whatever you put between the `<html>` tags determines how your web page looks and behaves. You can add headers, paragraphs, lists, images, links, and all that jazz.
- **HTML Language Alert:** With the `<html>` tag, you can specify the HTML version and language. For example, `<html lang="en">` tells the browser and search engines that it's English content they're dealing with.

Here's an example of a basic HTML structure using the `<html>` tag:

```
<!DOCTYPE html>
<html lang="en">
```

```
<!-- More HTML elements go here -->
```

```
</html>
```

The `<html>` tag serves as the foundation for structuring and organizing the elements that make up a web page, allowing browsers to interpret and render the content correctly.

Project Implementation Step 1:

## <head> tag

The `<head>` tag is like the behind-the-scenes superstar of an HTML document. It contains important information and instructions for the browser, shaping how your web page is presented and behaves.

Here's the deal with the `<head>` tag:

- **Meta-information:** Within the `<head>` tags, you can include elements like `<title>`, `<meta>`, `<link>`, and `<style>`. These elements provide crucial details and instructions to the browser, such as the document title, character encoding, linked stylesheets, and more.

- This meta tag specifies the character encoding for the HTML document. In this case, it is set to "UTF-8," a widely-used character encoding standard that supports a wide range of characters and alphabets. UTF-8 ensures the webpage can properly display and handle different languages and special characters.
- A favicon, short for "favorite icon," is a small icon associated with a website displayed in the browser's address bar, next to the page title, and in bookmarks or tabs. It helps users quickly identify and recognize a website.
- `<meta name="viewport" content="width=device-width, initial-scale=1.0" />`: This meta tag is related to responsive web design and is used to control the initial layout and scaling of the webpage on different devices. It includes two properties:
  - `width=device-width`: This property sets the width of the webpage to the width of the device's screen. It ensures that the webpage adjusts its layout to fit the screen size, allowing for a better user experience on various devices.
  - `initial-scale=1.0`: This property sets the initial zoom level when the webpage is first loaded. A value of 1.0 means that the webpage is displayed at a 1:1 scale without any zooming. This ensures that the webpage appears at an optimal size when accessed on different devices.
- Document Title: One key element in the `<head>` section is `<title>`. It lets you set the title that appears in the browser's title bar or tab. It's like naming your web page!
- External Resources and Scripts: The `<head>` section is where you can link external resources, such as CSS stylesheets and JavaScript files, using the `<link>` and `<script>` elements. These resources enhance the presentation and functionality of your web page.
- SEO and Accessibility Boost: You can also include meta tags, like `<meta name="description" content="...">`, in the `<head>` section to provide a description of your web page for search engines. It's an opportunity to improve search engine optimization (SEO) and make your page more accessible.

Here's an example of an HTML structure with the `<head>` tag:

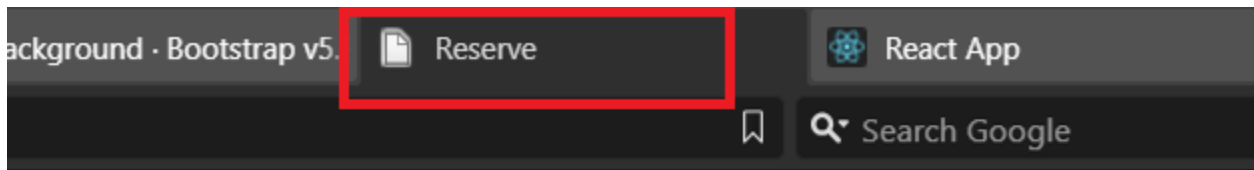
```
<head>
```

```
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <link rel="shortcut icon" href="favicon.ico" />
<title>Title of the document</title>

</head>
```

In this example, the `<head>` section contains the `<meta>` element to specify the character encoding, the `<title>` element for the document title, and the `<link>` and `<script>` elements to link external CSS and JavaScript files, respectively.

Project Implementation Step 2:



## `<body>` tag

The `<body>` tag is where the real action happens in an HTML document. It represents the content users see and interact with on your web page. It's where you bring together all the elements to create a visually appealing, interactive, and engaging experience for your website visitors. Don't forget to close it with `</body>` to keep things organized.

Here's the scoop on the `<body>` tag:

- **Content:** All the visible elements that make up your web page, such as text, images, headings, paragraphs, links, and more, reside within the `<body>` tags. This is where you bring your web page to life!
- **Page Structure:** The order and arrangement of elements within the `<body>` tag define the structure and layout of your web page. You can use HTML elements like `<header>`, `<nav>`, `<main>`, `<section>`, and `<footer>` to organize and structure your content for better readability and accessibility.
- **Interactive Elements:** The `<body>` section is where you can include interactive elements like forms, buttons, input fields, and multimedia content such as videos

and audio. These elements allow users to interact with your web page and enhance the user experience.

- **Styling and Scripting:** You can also include inline styles or link external CSS stylesheets within the `<body>` section to define the visual appearance of your content. Additionally, JavaScript code can be placed within `<script>` tags in the `<body>` section to add interactivity and dynamic behavior to your web page.

Here's an example of an HTML structure with the `<body>` tag:

```
<body>
  <header>
    <!-- More HTML elements go here -->
  </header>

  <main>
    <!-- More HTML elements go here -->
  </main>

  <footer>
    <!-- More HTML elements go here -->
  </footer>
</body>
```

In this example, the `<body>` section contains the visible content of the web page, such as the `<header>` section, a `<main>` section, and a `<footer>` section at the end.

## Semantic Elements

Semantic elements in HTML are tags that provide meaning and structure to the content within a webpage. They help to convey the purpose and significance of different sections or elements of a webpage, making the code more readable, accessible, and search engine-friendly. Here are some commonly used semantic elements in HTML:

`<header>`, `<nav>`, `<main>`, `<section>`, `<aside>`, `<footer>`.

In the non-semantic example, `<div>` is a generic container element that does not provide any specific meaning or structure to the content. It is often used for grouping or styling purposes. While it can be styled to resemble a heading visually, it does not carry the semantic significance of a heading, making it less informative to assistive technologies and search engines.

By using semantic elements, you make your HTML more meaningful and accessible, enhancing the overall user experience and improving the search engine optimization of your website.

Project Implementation Step 3:

## <header> tag

The `<header>` tag is an HTML semantic element that represents the introductory section or a container for the header of a webpage or a section within a webpage. It typically contains information considered introductory or related to the overall content of the webpage. Here's a breakdown of its usage and characteristics:

The `<header>` element is usually placed at the top of a webpage or within a section to introduce the main content or provide a consistent header across multiple sections.

The `<header>` tag carries semantic meaning, signaling that the enclosed content is related to the introductory or navigational aspects of the webpage. Assistive technologies, search engines, and developers can interpret this information to understand the significance of the content within the header.

```
<header>
  <nav>
    <!-- More HTML elements go here -->
  </nav>
</header>
```

## <nav> tag

In HTML, the `<nav>` tag is used to define a section of a webpage that contains navigation links. It is part of the semantic HTML5 markup, which provides a clearer structure and meaning to the content of a webpage. The `<nav>` element is typically used to group together a set of navigation-related links.

Here's an explanation of the elements commonly used within a `<nav>` tag:

- **<a> (Anchor) Tag:** The `<a>` tag is used to create a hyperlink. It is commonly used within the `<nav>` element to define individual navigation links. The anchor tag is represented as `<a>` and `</a>` with the link text between them.
- **href Attribute:** The `href` attribute is used within the `<a>` tag to specify the URL (Uniform Resource Locator) that the link should navigate to when clicked. It is assigned a value, such as a web page URL or an anchor within the same page. For example, `href="http://example.com"` would link to the webpage at ["http://example.com"](http://example.com).
- **target Attribute:** The `target` attribute is used within the `<a>` tag to specify how the linked webpage should open when clicked. The attribute value determines the target window or tab for the link. Common values include:



- `_blank`: Opens the linked webpage in a new browser tab or window.
- `_self`: Opens the linked webpage in the same window or tab.
- `_parent`: Opens the linked webpage in the parent frame or window.
- `_top`: Opens the linked webpage in the full body of the window, canceling any frames.
- A custom name: Opens the linked webpage in a specific named target, which can be defined in the `target` attribute of another element, such as a frame or iframe.

By using these elements together, the `<nav>` tag provides a structured way to group navigation links (`<a>` tags) and specify their destinations (`href` attribute) and how they should open (`target` attribute). This helps create a clear and meaningful navigation structure for website visitors.

```
<nav>
  <a href="http://example.com" target="_blank">Home</a>
  <a href="http://example.com/about" target="_self">About Us</a>
  <a href="#services" target="_self">Our Services</a>
</nav>
```

## <img> tag

The `<img>` tag is an HTML element used to insert an image into a webpage. It is a self-closing tag, meaning it does not require a closing tag.

The `<img>` tag has several attributes that define the properties and behavior of the image:

- `src`: This attribute specifies the URL or file path of the image to be displayed. It can be a relative or absolute path.
- `alt`: The `alt` attribute provides alternative text for the image. It is displayed when the image fails to load or for accessibility purposes to describe the image to users with visual impairments.
- `width` & `height`: These attributes define the width and height dimensions of the image in pixels. They can be used to set the image's display size on the webpage.
- `title`: The `title` attribute provides additional information about the image. When a user hovers over the image, the title text is displayed as a tooltip.

```

```

## <div> tag

The `<div>` tag in HTML is a fundamental element used to create a division or section within a webpage. It serves as a container or wrapper that allows you to group and organize other HTML elements, such as text, images, forms, or other HTML tags, together as a cohesive unit.

The `<div>` tag itself does not have any inherent semantic meaning, as it is a generic container. It is primarily used for styling and layout purposes or to group related elements for scripting or styling purposes.

**Note:** You can apply CSS styles to a `<div>` element using classes or IDs to control its appearance and behavior.

```
<div>
  <!-- More HTML elements go here -->
</div>
```

Project Implementation Step 4:



Reserve

[Tickets](#) [Contact Us](#) [Login](#) [Register](#)

## <main> tag

The `<main>` tag is a semantic HTML5 element used to define the main content area of a webpage. It represents the primary content of a document or a section within a document. The `<main>` element should typically be unique within a webpage and not be nested within other `<main>` elements.

The purpose of the `<main>` tag is to provide clarity and structure to the HTML markup by identifying the main content section that is relevant to the page's topic or purpose. It helps search engines, screen readers, and other assistive technologies to understand and navigate the primary content of the webpage more easily.

Here's an example of how the `<main>` tag can be used:

```
<main>
  <div>
    <!-- More HTML elements go here -->
  </div>
</main>
```

## `<section>` tag

In HTML, the `<section>` tag is used to define a section or a distinct grouping of content within a webpage. It is a semantic element that helps to structure and organize the content of a webpage into meaningful sections.

The `<section>` tag is typically used when there is a logical division or theme within the content.

For example: A webpage about a company may have sections for "About Us," "Services," and "Contact Information." Each of these sections can be represented using the `<section>` tag.

```
<section>
  <!-- More HTML elements go here -->
</section>
```

Using the `<section>` tag helps developers and search engines understand the structure and hierarchy of the webpage. It provides a way to group related content together and allows for easier styling and targeting of specific sections using CSS or JavaScript.

It's important to note that the `<section>` tag does not imply any specific styling or formatting. It is a generic container element, and its meaning and purpose should be determined based on the content it contains.

## Dropdown in HTML

In HTML, a dropdown, also known as a select or dropdown menu, is a form element that allows users to select one option from a list of predefined options. When the user clicks on the dropdown, a list of options is displayed, and they can choose one option from the list. The selected option is displayed as the default or current selection in the dropdown.

To create a dropdown in HTML, you use the `<select>` element along with the `<option>` elements to define the available options.

### `<select>` Element:

The `<select>` element represents a control that provides a list of options for the user. It acts as a container for the `<option>` elements and defines the dropdown menu.

Attributes of the `<select>` element:

- **name**: Specifies the name of the select element, which is used to identify the selected option when the form is submitted.
- **multiple**: If specified, allows the user to select multiple options at once.
- **size**: Specifies the number of visible options in the dropdown menu.

Here's an example of how the `<select>` element can be used:

```
<select name="fruits">
  <option value="apple">Apple</option>
  <option value="banana">Banana</option>
  <option value="orange">Orange</option>
</select>
```

## `<option>` Element:

The `<option>` element is used to define the individual options within the `<select>` element. Each `<option>` represents a choice that the user can select from the dropdown menu.

Attributes of the `<option>` element:

- **value**: Specifies the value associated with the option. This value is sent to the server when the form is submitted.
- **selected**: Specifies that the option should be pre-selected when the page loads.

Here's an example of how the `<option>` element can be used:

```
<select name="fruits">
  <option value="apple">Apple</option>
  <option value="banana" selected>Banana</option>
  <option value="orange">Orange</option>
</select>
```

In the above example, the "Banana" option is pre-selected because of the **selected** attribute. When the form is submitted, the selected option's **value** attribute (in this case, "banana") will be sent to the server.

Project Implementation Step 5:

From:

Delhi ▼

To:

Delhi ▼

## `<p>` tag

In HTML, the `<p>` tag is used to define a paragraph element. It is a block-level element that represents a self-contained unit of text or content.

When the `<p>` tag is used, it typically denotes a new paragraph of text within a larger document. By default, paragraphs are rendered with some space before and after them, creating a visual separation between different paragraphs.

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

The `<p>` tag can be used to structure and organize text content on a webpage, making it more readable and structured for both users and search engines. It is commonly used in blog posts, articles, and other textual content on websites.

## `<h1>` to `<h6>` tags

In HTML, `<h1>` to `<h6>` are heading tags used to define different levels of headings in a document. These tags are part of the structural elements in HTML and play a crucial role in organizing and visually formatting the content.

The `<h1>` tag represents the highest level of heading and is typically used for the main title or heading of a webpage. It carries the most significance and is visually displayed as the largest and most prominent heading.

The hierarchy continues with `<h2>` being the second-level heading, `<h3>` as the third level, and so on, down to `<h6>`, which represents the lowest level of heading. As the heading level increases (from `<h1>` to `<h6>`), the font size and visual prominence typically decrease, reflecting the diminishing importance of the heading structure.

```
<h1>Heading 1</h1>
```

```
<h2>Heading 2</h2>
```

```
<h3>Heading 3</h3>
```

```
<h4>Heading 4</h4>
```

```
<h5>Heading 5</h5>
```

```
<h6>Heading 6</h6>
```

# Heading 1

## Heading 2

### Heading 3

#### Heading 4

##### Heading 5

###### Heading 6

Here's an overview of the general usage and typical scenarios for each heading level:

- `<h1>`: Typically used for the main title of the webpage. There should generally be only one `<h1>` per page.

- `<h2>`: Used for major section headings or subheadings within the main content area.
- `<h3>` to `<h6>`: Used for subsections or headings of decreasing importance within the content. These tags are often used to create hierarchical structures, with `<h3>` being a subsection of `<h2>`, `<h4>` being a subsection of `<h3>`, and so on.

It's important to use heading tags appropriately to maintain a logical and meaningful structure for your content. This not only helps with visual styling but also improves accessibility and search engine optimization by providing a clear hierarchy of information.

## <button> tag

The `<button>` tag in HTML is used to create a clickable button on a webpage. It is a versatile element that can be used for various purposes, such as submitting forms, triggering JavaScript functions, or navigating to another page. The `<button>` tag can have different attributes to specify its behavior and appearance.

Here's an example code to illustrate the usage of the `<button>` tag with some common attributes:

- `type`: Specify the type of the button. It can take different values.

```
<button type="button">Learn More</button> <!-- Regular button - - >
<button type="submit">Submit</button> <!-- Submit button in a form - ->
<button type="reset">Reset</button> <!-- Reset button in a form - ->
```

- `disabled`: Disables the button, making it unclickable and visually inactive.

```
<button disabled>Disabled Button</button>
```

- `value`: Specify the value associated with the button when submitting a form.

```
<button type="submit" value="submit">Submit</button>
```

Project Implementation Step 6:



Reserve

[Tickets](#) [Contact Us](#) [Login](#) [Register](#)

From:

Delhi ▾

To:

Delhi ▾

Travel with world's largest bus services



2000+

Bus Collection



2 Million+

Happy customers globally



5000+

Tickets book everyday

**Here's what a few of our customers have to say**

**Vatsal Agarwal**

Customer since 2020

4.9 \*

Amazing travel experience with reserve. Excellent staff

**Rohan Sharma**

Customer since 2022

4.3 \*

Amazing service. Always a best time with reserve.

**Virat Kohli**

Customer since 2019

4.5 \*

Bus was clean and the journey was smooth. Reached on time.

**<footer> tags**

By Nehal Kadyan

In HTML, the `<footer>` tag is used to define the footer section of a webpage. It represents the bottommost part of a document or a specific section within a document, typically containing information related to the document itself or the website.

The `<footer>` tag is a semantic element that helps convey the structure and meaning of the content to both browsers and search engines. It is commonly placed inside the `<body>` element of an HTML document, typically after the main content.

The footer section often includes information such as copyright notices, legal disclaimers, contact details, links to relevant pages or documents, and navigation menus specific to the footer. It provides a consistent location for this information across multiple pages of a website.

Here's an example of how the `<footer>` tag can be used in HTML:

```
<footer>  
  <!-- More HTML elements go here -->  
</footer>
```

Including a `<footer>` tag in an HTML document helps improve its structure, accessibility, and user experience by separating the footer section from the main content and providing a designated area for important information related to the webpage or website.

Project Implementation Step 7:





Reserve

[Tickets](#) [Contact Us](#) [Login](#) [Register](#)

From:

Delhi

To:

Delhi

Travel with world's largest bus services



2000+

Bus Collection



2 Million+

Happy customers globally



5000+

Tickets book everyday

Here's what a few of our customers have to say

**Vatsal Agarwal**

Customer since 2020

4.9 \*

Amazing travel experience with reserve. Excellent staff

**Rohan Sharma**

Customer since 2022

4.3 \*

Amazing service. Always a best time with reserve.

**Virat Kohli**

Customer since 2019

4.5 \*

Bus was clean and the journey was smooth. Reached on time.

© 2023 AlmaBetter. All rights reserved

Follow Us



# Summary

## What did we learn?

Sure, here's the text with backticks added before and after the HTML elements and tags:

- HTML is the standard markup language used for creating web pages and applications. It provides a structured way to organize and format content on a webpage.
- HTML documents start with the `<!DOCTYPE html>` declaration to specify the HTML version.
- Comments can be added in HTML using `<!-- -->` syntax.
- The `<html>` tag is the root element of an HTML document and contains other elements.
- The `<head>` tag contains meta-information, such as the document title and linked stylesheets.
- The `<body>` tag represents the visible content of the webpage and includes interactive elements.
- Semantic elements like `<header>`, `<nav>`, `<main>`, `<section>`, `<aside>`, and `<footer>` provide meaning and structure to the content.
- Insert images into a webpage using the `<img>` tag. Group and organize HTML elements with the `<div>` tag.
- Create dropdown menus using the `<select>` and `<option>` tags. Define paragraphs of text or content with the `<p>` tag.
- Use headings of different levels (`<h1>` to `<h6>`) for structuring content. Create clickable buttons with the `<button>` tag.

## Shortcomings & Challenges

- Different web browsers may interpret HTML code differently, leading to inconsistencies in how webpages are displayed.

- Ensuring web content is accessible to users with disabilities is an important challenge.
- Creating webpages that adapt and display well on different devices and screen sizes can be challenging.
- Search engine optimization requires careful consideration of HTML structure, meta tags, headings, and content organization.

## Best practices to follow

- Adhering to HTML standards and validating HTML code helps ensure proper rendering across different browsers and devices.
- Utilize semantic elements (`<header>`, `<nav>`, `<main>`, etc.) to provide clear structure and meaning to the content. This improves accessibility and helps search engines understand the page's content.
- Maintain consistent code formatting and indentation to improve code readability and maintainability.
- Include descriptive alt attributes for images to provide alternative text for users who cannot see them. This improves accessibility and helps with SEO.
- Use heading tags (`<h1>` to `<h6>`) in a logical hierarchy to structure content. Use `<h1>` for the main page heading and subsequent headings for sub-sections.
- Optimize HTML structure, meta tags, and content to improve search engine visibility. Use relevant keywords, descriptive titles, and meta descriptions.

## Enhance your knowledge

Supercharge your knowledge by exploring the resources provided! 

- Text-level Semantic Elements:  
<https://www.w3.org/TR/2014/REC-html5-20141028/text-level-semantics.html>
- Inline and Block Elements:  
<https://medium.com/ux-art/html-block-and-inline-elements-b65775e86599>