Lesson 2: HTML Fundamentals

In this chapter, we will explore the fundamental concepts of HTML (Hypertext Markup Language) and how it is used to structure web content. HTML acts as the backbone of web pages, providing the necessary tags to define elements and organize the information presented to users. To write HTML code, you can use Integrated Development Environments (IDEs) such as Visual Studio Code, Sublime Text, or Atom.

HTML is a markup language that defines the structure and layout of web pages. It comprises a collection of tags, each serving a specific purpose in designating elements such as headings, paragraphs, lists, images, and links. With a good understanding of HTML, you can effectively structure web content and create visually appealing and accessible web pages. Now, let's embark on our journey to explore HTML fundamentals!

HTML Document Structure

The structure of an HTML document is essential for creating well-formed web pages. In this section, we'll explore the key components that form the foundation of an HTML document and provide a code example to illustrate the concepts.

At the beginning of an HTML document, we include the doctype declaration, which informs the browser about the version of HTML being used. This declaration ensures that the document is interpreted correctly.

Here's an example of the doctype declaration for HTML5:

```
<!DOCTYPE html>
```

Next, we have the **<html>** element, which serves as the root element of the document. It encapsulates the entire HTML content and provides the starting point for organizing the structure of the page.

```
<!DOCTYPE html>
<html>
  <!-- Rest of the document goes here -->
</html>
```

Within the **<html>** element, we distinguish between the **<head>** and **<body>** sections. The **<head>** section contains metadata and information about the document, such as the page title, character encoding, and external stylesheets. These elements contribute to search engine optimization and provide additional information to browsers and other tools.

```
<!DOCTYPE html>
<html>
  <head>
    <title>My Web Page</title>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="styles.css">
  </head>
  <body>
    <!-- Content of the web page goes here -->
  </body>
</html>
```

On the other hand, the **<body>** section represents the visible content of the web page. It contains all the elements that are rendered and displayed to users, including text, images, links, and interactive components. The content within the **<body>** section is what users see and interact with when visiting a web page.

Additionally, we'll cover the inclusion of metadata within an HTML document. Metadata includes the page title, which appears on the browser's title bar or tab, and helps identify the content of the page. We'll also explore how to specify the character encoding to ensure that the browser interprets special characters correctly. Moreover, we'll discuss how to link external stylesheets to apply consistent styles and layouts across multiple pages.

Understanding the structure of an HTML document allows us to organize and present web content effectively. By properly utilizing the doctype declaration, **<html>** element, **<head>** and **<body>** sections, and including metadata, we ensure that our web pages are structured correctly and provide a solid foundation for further development.

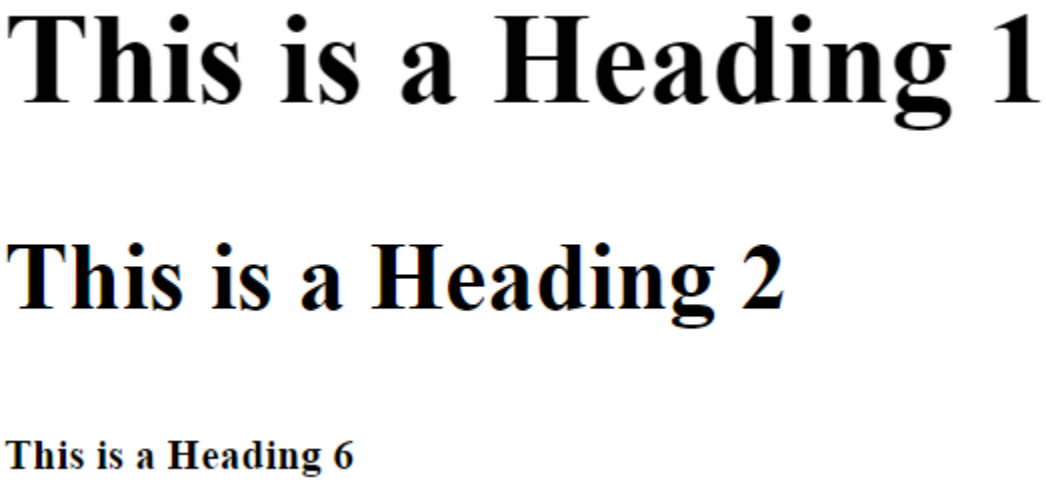
Text Formatting and Structure Tags

In this section, we'll dive into the various tags available in HTML for text formatting and structuring content. These tags allow you to organize and present your text in a well-organized and visually appealing manner. Let's explore some of the essential tags for text formatting:

Headings:
HTML provides six levels of headings, ranging from **<h1>** (the highest level) to **<h6>** (the lowest level). Headings are used to define the hierarchical structure of your content, with **<h1>** being the main heading and **<h6>** being the least prominent.

```
<h1>This is a Heading 1</h1>
<h2>This is a Heading 2</h2>
<!-- ... -->
<h6>This is a Heading 6</h6>
```

Output:



Paragraphs:
The **<p>** tag is used to define paragraphs. It provides a logical grouping of text content.

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

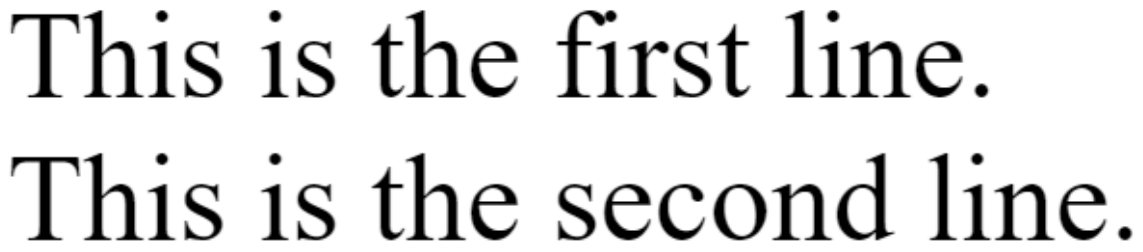
Output:

Line Breaks:

The **
** tag is used to create line breaks within a paragraph or other inline elements. It is a self-closing tag and does not require a closing tag.

```
<p>This is the first line.<br>This is the second line.</p>
```

Output:

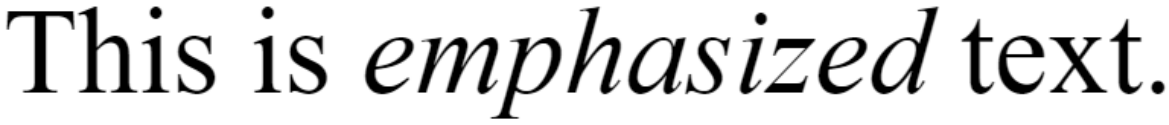


Emphasis:

The **** tag is used to emphasize text, typically rendered in italics by default. It indicates emphasis without conveying any specific semantic meaning.

```
<p>This is <em> emphasized </em> text.</p>
```

Output:



Strong:

The **** tag is used to indicate strong importance or emphasis, typically rendered in bold by default. It carries a stronger emphasis than the **** tag.

```
<p>This is <strong> strong </strong> text.</p>
```

Output:

This is **strong** text.

These are just a few examples of the tags available for text formatting and structuring content in HTML. By using these tags effectively, you can create visually appealing and reader-friendly web content. Experiment with different tags and explore their CSS properties to further enhance the visual presentation of your text.

Creating Forms, Tables, and Multimedia Elements

Forms are an essential part of web development as they allow users to input data and interact with websites. In HTML, we use the **<form>** element to create forms.

Let's take a look at an example of a basic form that includes input fields and a submit button:

```
<form>
    <label for="name">Name:</label>
    <input type="text" id="name" name="name" placeholder="Enter your name">

    <label for="email">Email:</label>
    <input type="email" id="email" name="email" placeholder="Enter your email">

    <input type="submit" value="Submit">
  </form>
```

Output:

Name: Enter your name Email: Enter your email Submit

In the above example, we use the **<label>** element to provide a description for each input field. The for attribute of the **<label>** element matches the id attribute of the corresponding input field, creating a connection between the label and the input field. This improves accessibility and usability.

We use the **<input>** element to create various types of input fields, such as text fields, email fields, checkboxes, radio buttons, and more. The type attribute specifies the type of input field. In the example above, we use the type="text" and type="email" attributes to create text and email input fields, respectively.

The **<input type="submit">** element creates a submit button that allows users to submit the form. You can customize the text on the button by setting the value attribute.

Organizing Data with HTML Tables

HTML tables are used to organize and present data in a tabular format. They are created using the **<table>**, **<tr>**, **<th>**, and **<td>** elements.

Let's see an example of a simple table:

```
<table>
    <tr>
      <th>Name</th>
      <th>Age</th>
      <th>Email</th>
    </tr>
    <tr>
      <td>John Doe</td>
      <td>25</td>
      <td>johndoe@example.com</td>
    </tr>
    <tr>
      <td>Jane Smith</td>
      <td>30</td>
      <td>janesmith@example.com</td>
    </tr>
  </table>
```

Output:

Name	Age	Email
John Doe	25	johndoe@example.com
Jane Smith	30	janesmith@example.com

In the example above, we use the **<table>** element to create the table. The table rows are defined using the **<tr>** element, and the table headers are defined using the **<th>** element within the **<tr>** element. The table data cells are defined using the **<td>** element within the **<tr>** element.

By adding more rows and cells, you can expand the table to display more data. You can also use CSS to style the table, apply alternate row colors, and customize the table's appearance.

Incorporating Multimedia with HTML

HTML provides elements for embedding multimedia content, such as images, audio, and video.

Let's take a look at an example of including an image in an HTML document:

```
<img src="image.jpg" alt="Description of the image">
```

In the above example, we use the **<img>** element to embed an image in the web page. The src attribute specifies the path to the image file, and the alt attribute provides alternative text for the image. The alternative text is important for accessibility, as it describes the image for users who cannot see it.

Similarly, HTML provides the **<audio>** and **<video>** elements for embedding audio and video content, respectively. You can specify the source file using the src attribute and include alternative content between the opening and closing tags in case the audio or video cannot be played.