Basics of HTML (HyperText Markup Language)

HTML is the standard markup language used to create the structure of web pages. It defines the elements that make up a webpage, such as headings, paragraphs, images, links, and more. Every webpage you see on the internet is built using HTML as its foundation.

Key Concepts of HTML

1. Structure of an HTML Document

An HTML document is made up of **elements** that are enclosed in **tags**. The basic structure looks like this:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    <h1>Hello, World!</h1>
    <p>This is a simple HTML page.</p>
  </body>
</html>
```

- <!DOCTYPE html>: Declares the document as an HTML5 document.
- <html>: The root element that contains all other HTML elements.
- <head>: Contains metadata about the document (e.g., title, links to stylesheets, etc.).
- <title>: The title displayed in the browser tab.
- <body>: Contains the content that is displayed on the webpage.

---

2. HTML Tags and Elements

- **Tag**: The name inside the angle brackets (< >). Tags usually come in pairs:
  - Opening tag: <p>
  - Closing tag: </p>
- **Element**: The complete structure, including the opening tag, content, and closing tag:
- <p>This is a paragraph.</p>

---

3. Common HTML Tags

Here are some of the most frequently used HTML tags:

Tag	Description	Example
<h1> to <h6>	Headings (h1 is largest, h6 is smallest)	<h1>Main Heading</h1>
<p>	Paragraph	<p>This is a paragraph.</p>
<a>	Hyperlink	<a href="https://example.com">Click here</a>
<img>	Image	<img src="image.jpg" alt="Description">
<ul>	Unordered list	<ul><li>Item 1</li><li>Item 2</li></ul>
<ol>	Ordered list	<ol><li>First</li><li>Second</li></ol>
<li>	List item	<li>Item</li>
<table>	Table	<table><tr><td>Data</td></tr></table>
<div>	Division/block (used for grouping)	<div>This is a container.</div>
<span>	Inline container (for styling text)	<span style="color:red;">Red text</span>
<form>	Form for user input	<form><input type="text"></form>
<input>	Input field	<input type="text" placeholder="Enter text">

---

4. Attributes in HTML

Attributes provide additional information about an element. They are written inside the opening tag.

Example of an <img> tag with attributes:

<img src="image.jpg" alt="A description of the image" width="300" height="200">

- src: Specifies the image source.
- alt: Provides alternative text if the image doesn't load.
- width and height: Define the size of the image.

Example of a hyperlink (<a> tag):

<a href="https://example.com" target="_blank">Visit Example</a>

- href: URL to navigate to when clicked.
- target="_blank": Opens the link in a new tab.

---

5. Nesting Elements

HTML elements can be nested inside one another, but they must be properly closed.

Correct Example:

```
<div>
  <h1>Heading</h1>
  <p>This is a paragraph inside a div.</p>
</div>
```

Incorrect Example:

```
<div>
  <h1>Heading
  <p>This is incorrect nesting.</h1>
</p>
```

---

6. Comments in HTML

Comments are notes in the code that are ignored by the browser. They are useful for explaining or organizing your code.

```
<!-- This is a comment -->
<p>This is visible on the webpage.</p>
```

Basic Example of an HTML Webpage

Here’s an example of a simple webpage with text, an image, and a link:

```
<!DOCTYPE html>
<html>
<head>
  <title>My First Webpage</title>
</head>
<body>
  <h1>Welcome to My Website</h1>
  <p>Hello! This is my first webpage. I'm learning HTML.</p>

  <img src="myphoto.jpg" alt="My photo" width="300">
  <p>Check out my favorite website:</p>
  <a href="https://example.com" target="_blank">Click here</a>
</body>
</html>
```

How HTML Works with CSS and JavaScript

1. **HTML**: Creates the structure (the skeleton of the webpage).
2. **CSS**: Styles the webpage (e.g., colors, layouts, fonts).
3. **JavaScript**: Adds interactivity (e.g., button clicks, animations).

For example:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    body { background-color: lightblue; }
    h1 { color: white; text-align: center; }
  </style>
</head>
<body>
  <h1>Welcome!</h1>
  <p>This page has some basic styling using CSS.</p>
</body>
</html>
```

Best Practices for Writing HTML

1. Use **semantic tags** (e.g., <header>, <footer>, <main>) for better accessibility and SEO.
2. Properly **indent your code** to make it more readable.
3. Always use **closing tags** where required.
4. Add **alt text** for images to improve accessibility.
5. Use meaningful **file names** and **folder structures** for organizing your project.
