



# HTML Fundamentals and Concepts

This document provides a comprehensive overview of fundamental HTML concepts, including basic elements, attributes, block vs. inline elements, tables, forms, and accessibility features like ARIA.

## 1. Introduction to HTML

### What is HTML?

HTML (HyperText Markup Language) is the standard markup language for documents designed to be displayed in a web browser. It describes the structure of a web page semantically and originally included cues for the appearance of the document.

### Tags, Elements, and Attributes

- **Tag:** Tags are the fundamental building blocks of HTML. They consist of the element name surrounded by angle brackets (e.g., `<p>`).
- **Element:** An HTML element is defined by a starting tag, some content, and an ending tag. The entire structure (e.g., `<p>This is a paragraph.</p>`) is the element.
- **Attribute:** Attributes provide additional information about HTML elements. They are always specified in the start tag and usually come in name/value pairs (e.g., `id="main"` or `class="highlight"`).

### Basic HTML Structure

Here is the code for the most basic HTML document:

```
<!DOCTYPE html><html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<title>My Basic HTML Page</title>
</head>
<body>
  <h1>Hello, World!</h1>
  <p>This is my first web page.</p>
</body>
</html>
'''
```

## 2. Core Text Elements

### Heading Tags

Heading tags are used to define the structure and hierarchy of content. HTML provides six levels of headings, from `<h1>` (most important) to `<h6>` (least important).

Tag	Purpose
<code>&lt;h1&gt;</code>	Main title/most important heading
<code>&lt;h2&gt;</code>	Section heading
<code>&lt;h3&gt;</code>	Sub-section heading
<code>&lt;h6&gt;</code>	Least important heading

**Code Example:**  
`<h1>Main Document Title</h1>`  
`<h2>A Section Header</h2>`  
`<h3>A Sub-section Header</h3>`  
`<p>Some content under the sub-section.</p>`  
`'''`

### Paragraph Tags

The `<p>` tag is used to define a paragraph of text.

**Code Example:**  
`<p>This is the first paragraph of text on the page. It contains a lot of important information.</p>`

<p>This is a second, separate paragraph. Web browsers automatically add some space before and after paragraph elements.</p>

### 3. Block vs. Inline Elements

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

#### Block-Level Elements

- Block-level elements start on a new line and take up the full width available.
- Examples: <h1>, <p>, <div>, <ul>, <table>.

#### Inline Elements

- Inline elements do not start on a new line; they only take up as much width as necessary for their content.
- Examples: <a>, <span>, <strong>, <em>, <img>.

#### Types of Tags and Nested Elements

- **Paired Tags:** Most tags are paired (or container tags), requiring both an opening and a closing tag (e.g., <p> . . . </p>).
- **Empty Tags (Self-Closing):** These tags do not have an ending tag and do not wrap content. They perform a specific function (e.g., <br>, <hr>, <img>).

#### Nested Element Example:

Elements can be placed inside other elements; this is called nesting.<p>This paragraph contains a <strong>strong element</strong> which makes the text bold.</p>



<

This is the outer paragraph element. It contains:

<ul>

<li>An unordered list.</li>

<li>

A nested paragraph inside a list item.

<p>

This paragraph is **deeply nested**. It contains <em>emphasis</em> and

a <a href="#">link element</a>.

</p>

</li>

</ul>

The list is now closed. This is still the outer paragraph.

</p>

## 4. HTML Tables

Tables are used to display tabular data. Key tags include `<table>`, `<tr>` (table row), `<th>` (table header), and `<td>` (table data/cell).

The following code demonstrates a simple table structure.

Header 1	Header 2
Data A1	Data A2
Data B1	Data B2

**Code Example:**`<table>`

```
<thead>
```

```
<tr>
```

```
<th>Name</th>
```

```
<th>Email</th>
```

```
<th>Role</th>
```

```
</tr>
```

```
</thead>
```

```
<tbody>
```

```
<tr>
```

```
<td>Alice</td>
```

```
<td>alice@example.com</td>
```

```
        <td>Developer</td>

    </tr>

    <tr>

        <td>Bob</td>

        <td>bob@example.com</td>

        <td>Designer</td>

    </tr>

</tbody>

</table>
```

## 5. HTML Forms

Forms are essential for collecting user input. The main tags are `<form>`, `<input>`, `<textarea>`, and `<button>`.

### Form Tags and Input Boxes

The `<input>` tag is the most versatile form element, supporting various types like text, number, password, checkbox, and radio.

**Code Example:** `<form action="/submit-data" method="POST">`

```
<!-- Text input box →
```

```
<label for="username">Username:</label>
```

```
<input type="text" id="username" name="username" required>
```

```
<br>
```

```
<!-- Password input box →
```

```
<label for="password">Password:</label>
```

```
<input type="password" id="password" name="password">

<br>

<!-- Submit button →

<button type="submit">Log In</button>

</form>
```

## Practice Questions (Forms)

1. How would you create a text input box that only accepts email addresses, and is required? (Hint: check the `type` attribute).
2. What is the purpose of the `action` attribute in the `<form>` tag?
3. Which HTML element is used to group related elements in a form and provide a caption for the group?

## 6. Multimedia and Other Basic Tags

### Image Tags

The `<img>` tag is an empty/self-closing tag used to embed an image into the document. The two most critical attributes are `src` (source) and `alt` (alternative text).

**Code Example:**`<span type="image">A small icon representing a digital photograph.</span>`

```

```

### Video Tags

The `<video>` tag embeds a video player. Attributes like `controls`, `autoplay`, and `loop` are commonly used.

**Code Example (with source):**`<video controls width="600">`

```
<source src="path/to/video.mp4" type="video/mp4">
```

```
<source src="path/to/video.webm" type="video/webm">
```

Your browser does not support the video tag.

</video>

## Basic Tags and Sources

The following table lists several other essential basic HTML tags.

Tag	Description
<a>	Hyperlink (anchor)
 	Line break (empty tag)
<hr>	Thematic break/horizontal rule (empty tag)
<ul>	Unordered list
<ol>	Ordered list
<li>	List item
<span>	Generic inline container
<div>	Generic block container

## 7. ARIA Concept (Accessibility)

### What is ARIA?

ARIA (Accessible Rich Internet Applications) is a set of attributes that you can add to HTML elements to improve accessibility for users with disabilities who use assistive technologies like screen readers. ARIA does not change the appearance of an element, only the semantic meaning communicated to assistive technology.

### Example: Making a Custom Button Accessible

If you use a generic `<div>` element as a button (which is generally discouraged, but common in custom UI libraries), a screen reader won't know it's interactive. ARIA attributes fix this.

**Code Example (Easy to Understand):** <!-- Without ARIA, a screen reader just sees a generic box →

```
<div id="custom-button-1">Click Me</div>
```

<!-- With ARIA, the screen reader knows this box is a button →

```
<div
```

```
  id="custom-button-2"
```

```
  role="button"
```

```
  aria-pressed="false"
```

```
  tabindex="0">
```

```
  Click Me
```

```
</div>
```

### Explanation:

- **role="button"**: Tells the screen reader that this `<div>` should be treated like a standard button element.
- **aria-pressed="false"**: Indicates the current state of the button (in this case, it is not currently pressed/toggled). This is useful for toggle buttons.
- **tabindex="0"**: Makes the element focusable via the keyboard, which is essential for keyboard-only users.

ARIA primarily uses three concepts: **roles** (what is this element?), **states** (what is its condition?), and **properties** (what is its nature?).