Block-level vs Inline Elements in HTML

Block-level Elements

- **Definition**: Block-level elements always start on a new line and take up the full width of their parent container.
- **Behavior**: Browsers automatically add a line break before and after block elements.
- **Use**: They are used to structure the layout into larger sections.
- **Examples**:
  - `<div>`
  - `<p>`
  - `<h1>` to `<h6>`
  - `<section>`, `<article>`
  - `<ul>`, `<ol>`, `<li>`
  - `<form>`, `<table>`

Example:

```
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
```

Both `<p>` elements appear on separate lines.

Inline Elements

- **Definition**: Inline elements do not start on a new line. They only take up as much width as their content.
- **Behavior**: They sit inside a line along with text or other inline elements.
- **Use**: They are used to style or format parts of text.
- **Examples**:
  - `<span>`
  - `<a>`
  - `<strong>`, `<em>`
  - `<img>`
  - `<label>`
  - `<b>`, `<i>`

Example:

```
<p>This is <span>inline text</span> inside a paragraph.</p>
```

The word *inline text* appears within the same line of the paragraph.

Metadata & Scripting Tags in HTML

These tags provide **information about the webpage**, connect external resources, and allow embedding of **CSS or JavaScript**. Most of them are used inside the `<head>` section of an HTML document.

1. `<head>`

The `<head>` tag contains **metadata** (information about the document) that is not displayed directly on the page.

Syntax:

```
<head>
  <title>My Website</title>
  <meta charset="UTF-8">
</head>
```

2. <title>

Specifies the **title of the webpage**, shown in the browser tab and used by search engines.

Syntax:

```
<title>HTML Notes</title>
```

3. <meta>

Provides **metadata** like description, keywords, author, charset, and viewport settings.

Syntax:

```
<meta charset="UTF-8">
<meta name="description" content="Learn HTML with notes">
<meta name="keywords" content="HTML, CSS, JavaScript">
<meta name="author" content="Neeraj Katheriya">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="refresh" content="10; url=https://example.com">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
```

Most Used Attributes:

- **`charset`** → Defines character encoding (e.g., UTF-8).
- **`name`** → Type of metadata (`description`, `keywords`, `author`, `viewport`).
- **`content`** → Value of the metadata.
- **`http-equiv`** → For HTTP headers (e.g., `refresh`, `X-UA-Compatible`).

5. <style>

Used to write **internal CSS** inside an HTML document.

Syntax:

```
<style>
  body { background-color: lightblue; }
</style>
```

6. <script>

Used to write **JavaScript code** or link an external JS file.

Syntax (Internal JS):

```
<script>
  alert("Hello World!");
</script>
```

Syntax (External JS):

```
<script src="app.js"></script>
```

Most Used Attributes:

- **`src`** → Path of external JS file.
- **`type`** → Type of script (`text/javascript`, optional in HTML5).

7. <noscript>

Defines **fallback content** for browsers that do not support JavaScript or if JS is disabled.

Syntax:

```
<noscript>
  <p>JavaScript is disabled in your browser.</p>
</noscript>
```

What is SEO?

SEO stands for **Search Engine Optimization**.
It is the process of **improving a website** so that it appears **higher in search engine results** (like Google, Bing, Yahoo).

When someone searches for a keyword related to your site, good SEO helps your website appear on the **first page** instead of being hidden on later pages.

Why SEO is Important?

- Increases **website visibility**
- Brings **organic (free) traffic**
- Builds **trust & credibility**
- Better **user experience**
- Helps compete with other websites

Extra HTML Tags

1. <dialog>
    - Defines a popup dialog box.
    - **Attributes**:
        - `open` → makes dialog visible.

    Example:

    ```
    <dialog open>
      <p>This is a dialog box</p>
    </dialog>
    ```

2. <menu>
    - Defines a list of commands or options (rarely used).

3. <progress> Tag

- Represents the **progress of a task**.
- Mostly used for **loading bars or task completion**.

Attributes:

- `value` → current progress (number).
- `max` → maximum value (default is 1).

Example:

```
<label for="task">Task Progress:</label>
<progress id="task" value="60" max="100"></progress>
```

Shows progress as **60% completed**.

4. <meter> Tag

- Represents a **measurement within a known range** (e.g., disk usage,temperature, score).

Attributes:

- `value` → current value.
- `min` → minimum value.
- `max` → maximum value.
- `low`, `high` → define ranges.
- `optimum` → ideal value.

Example:

```
<p>Disk Usage: <meter value="4" min="0" max="10">4 out of 10</meter></p>
```

5. <dl>, <dt>, <dd> (Definition List)

- Used for **glossaries, FAQs, or descriptions**.

Tags:

- `<dl>` → Definition List (container).
- `<dt>` → Definition Term.
- `<dd>` → Definition Description.

Example:

```
<dl>
  <dt>HTML</dt>
  <dd>HyperText Markup Language</dd>

  <dt>CSS</dt>
  <dd>Cascading Style Sheets</dd>
</dl>
```

6. <fieldset> and <legend>

- `<fieldset>` ka use **form ke input elements ko group karne** ke liye hota hai.
- Browser us group ke around ek **border box** bana deta hai.
- Form ko **organized aur readable** banata hai.

7. <legend> Tag

- `<legend>` tag **fieldset ka title/caption** dene ke liye use hota hai.
- Yeh box ke top-left corner me display hota hai.
- Ek `<fieldset>` me **sirf ek `<legend>`** allowed hota hai.

Example:

```
<form>
  <fieldset>
    <legend>Personal Information</legend>
    Name: <input type="text"><br>
    Age: <input type="number"><br>
  </fieldset>

  <fieldset>
    <legend>Contact Details</legend>
    Email: <input type="email"><br>
    Phone: <input type="tel"><br>
  </fieldset>
</form>
```

8. <svg> Tag (Scalable Vector Graphics)

- Used to draw **shapes, graphics, and icons** directly in HTML.
- Scales without losing quality.

Example:

```
<svg width="400" height="250" viewBox="0 0 400 250" aria-label="SVG
Shapes">
      <!-- rectangle -->
      <rect x="10" y="10" width="120" height="80" fill="#ffa94d"
stroke="#222" stroke-width="2"></rect>

      <!-- circle -->
      <circle cx="220" cy="50" r="40" fill="#74c0fc" stroke="#222" stroke-
width="2"></circle>

      <!-- line -->
      <line x1="10" y1="130" x2="390" y2="130" stroke="#fa5252" stroke-
width="4"></line>

      <!-- triangle -->
      <polygon points="300,30 360,110 240,110" fill="#69db7c" stroke="#222"
stroke-width="2"></polygon>

      <!-- text -->
      <text x="10" y="200" font-size="24" fill="#2f9e44">Hello SVG</text>
</svg>
```