HTML attributes

1. Global Attributes (Can be applied to almost all HTML elements)

- `class` – Specifies a class name for the element (used for styling or JavaScript).
- `id` – Specifies a unique ID for the element.
- `style` – Adds inline CSS styles to the element.
- `title` – Adds a tooltip to the element.
- `lang` – Specifies the language of the element's content.
- `dir` – Specifies the text direction (`ltr`, `rtl`, or `auto`).
- `data-*` – Custom data attributes for embedding extra data.
- `hidden` – Hides the element.
- `tabindex` – Specifies the tab order of an element.
- `accesskey` – Defines a shortcut key to activate or focus the element.
- `contenteditable` – Specifies if the content is editable (`true` or `false`).
- `draggable` – Specifies whether the element is draggable.
- `spellcheck` – Specifies whether to check the spelling and grammar of the element's content.
- `translate` – Specifies whether the element’s content should be translated.

2. Input and Form Attributes

- `type` – Specifies the type of input (e.g., `text`, `password`, `email`, `submit`, etc.).
- `name` – Specifies the name of the element (used in forms).
- `value` – Specifies the initial value of the element.
- `placeholder` – Displays placeholder text inside an input field.
- `required` – Specifies that the input must be filled out before submission.
- `readonly` – Specifies that the input is read-only.
- `disabled` – Disables the element.
- `maxlength` – Specifies the maximum number of characters allowed.
- `min` / `max` – Specifies the minimum or maximum value.
- `pattern` – Specifies a regular expression for input validation.
- `autocomplete` – Enables or disables autocomplete.
- `step` – Specifies the increment for numeric inputs.

3. Anchor (`<a>`) Attributes

- `href` – Specifies the URL of the link.
- `target` – Specifies where to open the linked document (`_self`, `_blank`, `_parent`, `_top`).

- `rel` – Specifies the relationship between the current document and the linked document.
- `download` – Specifies that the target should be downloaded.

4. Image (`<img>`) Attributes

- `src` – Specifies the image source.
- `alt` – Provides alternative text for the image.
- `width` / `height` – Specifies the width and height of the image.

5. Button Attributes

- `type` – Specifies the button type (`button`, `submit`, or `reset`).
- `disabled` – Disables the button.

6. Table Attributes

- `colspan` – Specifies the number of columns a cell should span.
- `rowspan` – Specifies the number of rows a cell should span.
- `border` – Specifies the width of the table border (deprecated in HTML5, use CSS instead).

7. Media Attributes (Audio/Video)

- `src` – Specifies the media file source.
- `controls` – Adds playback controls.
- `autoplay` – Starts playing the media automatically.
- `loop` – Specifies that the media will start over again when finished.
- `muted` – Specifies that the audio output should be muted.

8. Script Attributes

- `src` – Specifies the source file of the script.
- `type` – Specifies the script type (e.g., `text/javascript`).
- `defer` – Delays execution of the script until the HTML is fully parsed.
- `async` – Allows the script to execute asynchronously.

9. Meta Attributes

- `name` – Specifies the name of the metadata (e.g., `viewport`, `description`).
- `content` – Specifies the value of the metadata.
- `charset` – Specifies the character encoding (e.g., `UTF-8`).
- `http-equiv` – Provides HTTP header-like information (e.g., `refresh`, `Content-Type`).

10. ARIA (Accessibility) Attributes

- `aria-label` – Defines a label for the element.
- `aria-hidden` – Hides the element from assistive technologies.
- `role` – Defines the role of the element (e.g., `button`, `navigation`).

11. Event Attributes

- `onclick` – JavaScript executed when the element is clicked.
- `onchange` – Triggered when the value of an element changes.
- `onmouseover` – Triggered when the mouse pointer is over an element.
- `onkeydown` – Triggered when a key is pressed down.

Semantic Tags

Definition:
Semantic tags clearly describe their purpose and the content they hold. These tags improve accessibility, readability, and search engine optimization (SEO) by providing meaningful structure to a webpage.

Examples:

- `<header>`: Represents introductory content, such as a logo or navigation links.
- `<nav>`: Represents a section of navigation links.
- `<main>`: Represents the main content of the document.
- `<section>`: Defines a thematic grouping of content, often with a heading.
- `<article>`: Represents independent, self-contained content like blog posts or news articles.
- `<aside>`: Represents content tangentially related to the main content, like sidebars or callouts.
- `<footer>`: Represents the footer of a document or section.
- `<figure>`: Represents self-contained content, often with a `<figcaption>` for captions (e.g., images, diagrams).
- `<figcaption>`: Provides a caption or legend for a `<figure>`.
- `<time>`: Represents a specific point in time.
- `<mark>`: Highlights text to indicate its importance or relevance.
- `<address>`: Provides contact information.

Non-Semantic Tags

Definition:
Non-semantic tags do not convey any specific meaning about their content. They are often used as generic containers to style content or for layout purposes.

Examples:

- `<div>`: A generic container for grouping elements, primarily used for styling or layout with CSS.
- `<span>`: A generic inline container for text or inline elements, mainly used for styling.