HTML Lists

HTML Lists are used to specify lists of information. All lists may contain one or more list elements. There are three different types of *HTML lists*:

1. Ordered List (Numbered List)
2. Unordered List (Bulleted List)
3. Description Lists (Definition List)

Ordered List

In the ordered HTML lists, all the list items are marked with numbers by default.

It has the following syntax:

```
<ol>
 <li>Amazon</li>
<li>Google</li>
 <li>Meta</li>
 <li>Oracle</li>
</ol>
```

Attribute:

1. reversed
2. start: number
3. type (with css list-style-type): 1 | A | a | I | i

Unordered List

In HTML Unordered list, all the list items are marked with bullets.

It has the following syntax:

```
<ul>
 <li>Amazon</li>
<li>Google</li>
 <li>Meta</li>
 <li>Oracle</li>
</ul>
```

Description *List*

A description list is a list of terms, with a description of each term.

The `<dl>` tag defines the description list, the `<dt>` tag defines the term (name), and the `<dd>` tag describes each term. It has the following syntax:

```
<dl>
 <dt>Amazon</dt>
<dd> -focus on e-commerce</dd>
<dd> -focus on cloud computing,</dd>
<dd> -focus on online advertising</dd>
<dt>Google</dt>
<dd> -focus on search engine technology</dd>
<dd> -focus on cloud computing</dd>
 <dt>Meta</dt>
<dd> -owns facebook</dd>
<dd> -owns instagram</dd>
 <dt>Oracle</dt>
<dd>It is a multinational technology corporation</dd>
</ul>
```

Html Grouping Elements

Html Grouping element is used to group other HTML content together.

There are a number of additional tags that allow us to group related HTML content together. Some examples include:

`<div>, <header>, <section>, <article>, <footer>, <main>, <span>, <aside>`

All of these HTML elements have no effect on the HTML content itself, but instead group and contain other elements.

DIV

The `<div>` tag defines a division or a section in an HTML document.

The `<div>` tag is used as a container for HTML elements - which is then styled with CSS or manipulated with JavaScript.

The `<div>` tag is easily styled by using the class or id attribute.

Any sort of content can be put inside the `<div>` tag!

```
<div>
<h2>Heading for div</h2>
</div>
```

SPAN

The `<span>` tag is an inline container used to mark up a part of a text, or a part of a document.

The `<span>` tag is easily styled by CSS or manipulated with JavaScript using the class or id attribute.

The `<span>` tag is much like the element, but is a block-level element and `<span>` is an inline element.

Article

An **article** defines a complete or an *Self Contained composition* in a webpage. An **Article** could be a *Blog post*, *forum*, *newspaper article*, *an independent content*, *user comment*.

Tip: The `<aside>` content is often placed as a sidebar in a document.

An **article** must have an heading or subheading, i.e h1-h6

```
<article>    <h2>Heading for Article</h2></article>
```

Section

An **Section** represents an generic section of a document. **Section** could be various sections of an article with heading.

An section can include various chapters of a book or various webpage sections, like *introductions*, *content*, *contact info* etc.

```
<article>
<h1>Article Heading</h1>
<section>
   <h2>Subheading 1 for Section</h2>
</section>

<section>
   <h2>Subheading 2 for Section</h2>
</section>
</article>
```

Header

Defines a **header** for a document, section or an article. **Header** can be used more than once on a single webpage. But try to use single **header** in a section, or an article.

```
<section>
 <header>
   <h2>Header for Section<h2>
 </header>
 <p>Section Content Goes Here...</p>
</section>
```

NOTE: You can have several `<header>` elements in one HTML document. However, `<header>` cannot be placed within a `<footer>`, `<address>` or another `<header>` element.

Footer

Footer tag defines footer of an document, section or an article. A webpage can have multiple footers, but in separate sections.

```
<footer>
   <p>Content for footer<p>
</footer>
```

A `<footer>` element typically contains:

- authorship information
- copyright information
- contact information
- sitemap
- back to top links
- related documents

Main

The `<main>` tag specifies the main content of a document.

The content inside the `<main>` element should be unique to the document. It should not contain any content that is repeated across documents such as sidebars, navigation links, copyright information, site logos, and search forms.

Note: There must not be more than one `<main>` element in a document. The `<main>` element must NOT be a descendant of an <article>, <aside>, <footer>, <header>, or <nav> element.

HTML Block and Inline Elements

Every HTML element has a default display value, depending on what type of element it is.

Block-level Elements

A block-level element always starts on a new line, and the browsers automatically add some space (a margin) before and after the element.

A block-level element always takes up the full width available (stretches out to the left and right as far as it can).

Two commonly used block elements are: `<p>` and `<div>`.

The `<p>` element defines a paragraph in an HTML document.

The `<div>` element defines a division or a section in an HTML document.

Inline Elements

An inline element does not start on a new line.

An inline element only takes up as much width as necessary.

HTML Semantic Elements

A semantic element clearly describes its meaning to both the browser and the developer.

Why use semantic elements?

- First, it is much **easier to read**.
- It has **greater accessibility**
- **SEO**

Screen readers rely on semantically-rich HTML to process web pages and help users with low vision navigate them. Semantic tags tell screen reader users where they are and how they can interact with the current page.

Like with assistive technologies, semantic markup helps search engine crawlers navigate your page to better understand its structure and contents. Semantic HTML is a great way to show search engines what to index and get the most from your crawl budget.

HTML Form Elements

HTML Input Types

Here are the different input types you can use in HTML.

The `<input>` element can be displayed in several ways, depending on the type attribute:

<input type="button">

<input type="checkbox">

<input type="color">

<input type="date">

<input type="datetime-local">

<input type="email">

<input type="file">

<input type="hidden">

<input type="image">

<input type="month">

<input type="number">

<input type="password">

<input type="radio">

<input type="range">

<input type="reset">

<input type="search">

<input type="submit">

<input type="tel">

<input type="text">

<input type="time">

<input type="url">

<input type="week">

The <label> Element

The `<label>` element defines a label for several form elements.

The `<label>` element is useful for screen-reader users, because the screen-reader will read out loud the label when the user focus on the input element.

The `<label>` element also help users who have difficulty clicking on very small regions (such as radio buttons or checkboxes) - because when the user clicks the text within the `<label>` element, it toggles the radio button/checkbox.

The `for` attribute of the `<label>` tag should be equal to the `id` attribute of the `<input>` element to bind them together.

The <select> Element

The `<select>` element defines a drop-down list.

Use the `multiple` attribute to allow the user to select more than one value

The <textarea> Element

The `<textarea>` element defines a multi-line input field (a text area)

The <button> Element

The `<button>` element defines a clickable button

The <fieldset> and <legend> Elements

The `<fieldset>` element is used to group related data in a form.

The `<legend>` element defines a caption for the `<fieldset>` element.