**HTML Lists: Key Notes**

**📌 Overview**

Lists in HTML help structure content in an organized way. There are two main types:

1. **Unordered Lists (<ul>)** → Bullet points
2. **Ordered Lists (<ol>)** → Numbered items

**📍 Unordered List (<ul>)**

Used when order doesn’t matter (e.g., shopping lists, features).

**Example:**

html

CopyEdit

<ul>

<li>Milk</li>

<li>Eggs</li>

<li>Bread</li>

</ul>

🔹 Displays as bullet points.

**📍 Ordered List (<ol>)**

Used when order matters (e.g., step-by-step instructions).

**Example:**

<ol>

<li>Preheat oven</li>

<li>Mix ingredients</li>

<li>Bake for 20 minutes</li>

</ol>

🔹 Displays with numbers.

**✅ Best Practices**

* **Each list item must be inside <li> tags.**
* **Use <ul> for unordered, <ol> for ordered lists.**
* **Indent properly for readability.**

**🚀 Challenge: Format This List**

Convert this text into a properly formatted HTML list:

To-Do List:

- Buy groceries

- Walk the dog

- Finish project

# ****Nesting and Indentation in HTML Lists****

## ****1. Introduction****

* Lists can be **nested** inside other lists to create a hierarchical structure.
* Proper **indentation** improves readability and helps in debugging.

## ****2. Nesting Lists****

We can nest **ordered (<ol>)** and **unordered (<ul>)** lists inside another list item (<li>).

### ****Example: Simple Nested List****

#### **Code:**

<ul>

<li>Fruits

<ul>

<li>Apple</li>

<li>Banana</li>

<li>Cherry</li>

</ul>

</li>

<li>Vegetables

<ul>

<li>Carrot</li>

<li>Broccoli</li>

</ul>

</li>

</ul>

#### **Output:**

* Fruits
  + Apple
  + Banana
  + Cherry
* Vegetables
  + Carrot
  + Broccoli

## ****3. Complex Nested Lists****

Lists can be nested multiple levels deep, mixing <ul> and <ol>.

### ****Example: Complex Nested List****

#### **Code:**

<ul>

<li>A</li>

<li>B

<ol>

<li>B1</li>

<li>B2

<ul>

<li>B2a

<ul>

<li>B2aa</li>

<li>B2ab</li>

</ul>

</li>

<li>B2b</li>

<li>B2c</li>

</ul>

</li>

<li>B3

<ol>

<li>B31</li>

<li>B32</li>

</ol>

</li>

</ol>

</li>

<li>C</li>

</ul>

#### **Output:**

* A
* B
  1. B1
  2. B2
     + B2a
       - B2aa
       - B2ab
     + B2b
     + B2c
  3. B3
     1. B31
     2. B32
* C

## ****4. Importance of Indentation****

* Helps in **reading** and **understanding** nested structures.
* VS Code automatically corrects indentation when saving (Ctrl+S / Cmd+S).
* Proper nesting avoids missing closing tags and improves **debugging**.

### ****Common Mistake: Missing Closing Tag****

<ul>

<li>Item 1

<ul>

<li>Sub-item 1</li>

<li>Sub-item 2

</li> <!-- Incorrect closing tag placement -->

</ul>

**Error:** List structure is broken; proper indentation can help find mistakes.

## ****5. Summary****

✅ **Nesting lists** allow hierarchical structures.  
✅ **Indentation** improves readability and debugging.  
✅ Always **match opening and closing tags**.

**HTML Attributes & The Anchor (<a>) Element**

**1. Understanding Attributes**

* Attributes provide **additional information** about an HTML element.
* They are always written inside the **opening tag** of an element.
* Structure:

html

CopyEdit

<element attribute="value">Content</element>

* Attributes consist of:
  + **Attribute name**
  + **Equal sign (=)**
  + **Value enclosed in quotes ("")**

**2. The Anchor (<a>) Element**

* The <a> element creates **hyperlinks** in HTML.
* Basic syntax:

html

CopyEdit

<a href="https://example.com">Click Here</a>

* Key attribute:
  + **href** → Specifies the URL the link points to.
  + Without href, the anchor tag **won't function as a hyperlink**.

**3. Example Usage**

html

CopyEdit

<a href="https://www.google.com">Go to Google</a>

* The text "Go to Google" will be displayed as a clickable link.
* Clicking it redirects the user to https://www.google.com.

**4. Global vs. Specific Attributes**

* **Specific Attributes**: Used only for certain elements.
  + Example: href (specific to <a>).
* **Global Attributes**: Can be used on any HTML element.
  + Example: draggable="true" (makes an element draggable).

html

CopyEdit

<a href="https://example.com" draggable="true">Draggable Link</a>

**5. Creating a List of Favorite Websites**

* Use an **ordered list (<ol>)** to structure links:

html

CopyEdit

<h1>My Top 5 Favorite Websites</h1>

<ol>

<li><a href="https://www.google.com">Google</a></li>

<li><a href="https://www.youtube.com">YouTube</a></li>

<li><a href="https://www.github.com">GitHub</a></li>

<li><a href="https://www.reddit.com">Reddit</a></li>

<li><a href="https://www.wikipedia.org">Wikipedia</a></li>

</ol>

* Clicking a link opens the corresponding website.

**6. Using the start Attribute in <ol>**

* The start attribute allows numbering to begin from a **custom value**.

html

CopyEdit

<ol start="5">

<li>Google</li>

<li>YouTube</li>

</ol>

* This starts the list from **5 instead of 1**.

**7. Key Takeaways**

✅ **Attributes** provide extra details about HTML elements.  
✅ **The <a> element** requires the href attribute to create hyperlinks.  
✅ **Global attributes** like draggable can be used on any element.  
✅ **Lists** can be combined with anchor tags to create structured navigation.  
✅ **The start attribute** in <ol> can modify numbering.

**HTML Image Element – Quick Notes**

* **Purpose of <img> Tag:**
  + Embeds images in a webpage.
  + Syntax example: <img src="url" />
* **Key Attributes:**
  + **src (Source):**
    - Specifies the image’s URL.
    - Tells the browser where to fetch the image.
    - *Example:*

<img src="https://example.com/photo.jpg" />

* + **alt (Alternative Text):**
    - Provides a text description of the image.
    - Crucial for accessibility (e.g., screen readers for visually impaired users).
    - *Example:*

<img src="https://example.com/dog.jpg" alt="A playful dog in the park" />

* **Void Element / Self-Closing Tag:**
  + The <img> tag is a void element, meaning it doesn’t require a closing tag.
  + Other examples of void elements include <br> (line break) and <hr> (horizontal rule).
* **Using Placeholder Images:**
  + Services like [picsum.photos](https://picsum.photos/) provide random placeholder images.
  + You can specify the dimensions by appending the size in the URL.
  + *Example:*

<img src="https://picsum.photos/200/200" alt="Random placeholder image" />

* + - This example requests a 200x200 pixel image (often a random photo, like a forest).
* **Animated Images:**
  + GIF images work just like static images (JPEG, PNG) when used with <img>.
  + They automatically animate upon page load.
  + *Example:*

<img src="https://example.com/dog.gif" alt="Dog digging in the sand" />

* **Practice Exercise:**
  + **Task:** Determine if you’re a cat person or a dog person.
    - Create a heading using <h1> that states either "I'm a cat person" or "I'm a dog person."
    - Immediately after, add an image element using the corresponding image URL.
  + **Example Code:**

<h1>I'm a dog person</h1>

<img src="https://example.com/dog.gif" alt="A cute dog digging in the sand" />

* + This exercise reinforces:
    - How to create and structure an image element.
    - The importance of using the src and alt attributes correctly.
    - The concept of void elements in HTML.