

# Topic 3 — Lists in HTML : By Gagan Baghel

## Concepts Covered:

`<ul>` , `<ol>` , `<li>` , nested lists, `<dl>` , `<dt>` , `<dd>` , list types (`type` and `start` attributes).

## 🎯 Learning Objective:

To understand how to create and style different types of lists (unordered, ordered, nested, and descriptive) to organize content clearly and semantically.

## 💡 Practice Tasks (10 Total)

### Task 1: Create a Shopping List (Unordered List)

**Goal:** Learn to create a simple unordered list.

#### Instructions:

1. Create a file `shopping-list.html`.
2. Inside `<body>` , add `<h2>` — "My Shopping List".
3. Create an unordered list `<ul>` containing at least 5 items.
4. Use the `type` attribute (`disc` , `circle` , `square`) to test different bullet styles.
5. Add a comment describing which bullet style looks best.
6. output :

- Milk
- Bread
- Eggs
- Butter
- Apples

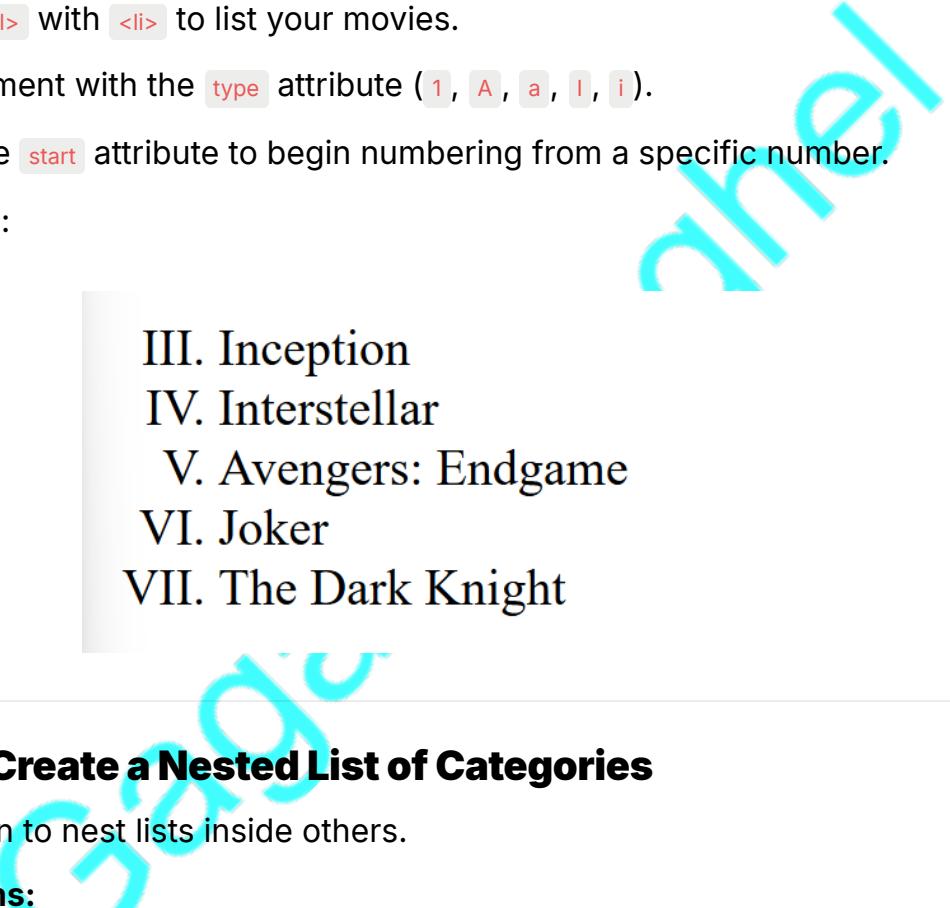
---

## Task 2: Create a Favorite Movies List (Ordered List)

**Goal:** Understand ordered lists and numbering.

**Instructions:**

1. Create `movies.html`.
2. Use `<h2>` — “Top 5 Favorite Movies”.
3. Use `<ol>` with `<li>` to list your movies.
4. Experiment with the `type` attribute (`1`, `A`, `a`, `I`, `i`).
5. Use the `start` attribute to begin numbering from a specific number.
6. output :

- 
- III. Inception
  - IV. Interstellar
  - V. Avengers: Endgame
  - VI. Joker
  - VII. The Dark Knight

---

## Task 3: Create a Nested List of Categories

**Goal:** Learn to nest lists inside others.

**Instructions:**

1. Create `categories.html`.
2. Create a main unordered list for categories: “Fruits”, “Vegetables”, “Snacks”.
3. Under each category, nest another list of 3–4 items using `<ul>`.
4. Experiment with different bullet styles for inner lists.
5. Add comments to explain parent-child list relationships.
6. output :

- Fruits
  - Apple
  - Banana
  - Orange
- Vegetables
  - Carrot
  - Potato
  - Tomato
- Snacks
  - Chips
  - Cookies
  - Popcorn

---

## Task 4: Create a Daily Routine (Ordered + Nested List)

**Goal:** Mix ordered and unordered lists.

**Instructions:**

1. Create `daily-routine.html`.
2. Create an ordered list for your daily schedule: Morning, Afternoon, Evening.
3. Under each, add a nested unordered list for tasks (e.g., Morning → Wake up, Breakfast).
4. Use indentation properly to keep hierarchy readable.
5. Add a `<h2>` heading "My Daily Routine".
6. output :

1. Morning
  - Wake up
  - Brush teeth
  - Exercise
  - Breakfast
2. Afternoon
  - Work/Study
  - Lunch
  - Short nap
3. Evening
  - Go for a walk
  - Dinner
  - Read a book
  - Sleep

## Task 5: Create a Glossary Page (Description List)

**Goal:** Understand description lists (`<dl>`, `<dt>`, `<dd>`).

**Instructions:**

1. Create `glossary.html`.
2. Use `<dl>` to define at least 5 web terms.

Example:

```
<dt>HTML</dt>
<dd>HyperText Markup Language — structure of web pages.</dd>
```

3. Add `<hr>` after every 2 definitions.
4. Comment explaining what `<dt>` and `<dd>` stand for.
5. output :

## HTML

HyperText Markup Language — structure of web pages.

## CSS

Cascading Style Sheets — used for styling HTML elements.

---

## JavaScript

Programming language for interactivity and logic.

## React

JavaScript library for building user interfaces.

---

## Node.js

Runtime environment for running JS on the server.

---

## Task 6: Create a “Course Curriculum” Page

**Goal:** Practice complex nested lists.

### Instructions:

1. Create `course.html`.
2. Use `<h2>` — “Web Development Curriculum”.
3. Use `<ol>` for modules (HTML, CSS, JavaScript).
4. Under each, nest `<ul>` for subtopics (e.g., HTML → Tags, Lists, Forms).
5. Use consistent indentation for clarity.
6. output :

## 1. HTML

- Tags
- Lists
- Forms

## 2. CSS

- Selectors
- Box Model
- Flexbox & Grid

## 3. JavaScript

- Variables
- Functions
- DOM Manipulation

---

### Task 7: Create a "Recipe Ingredients" Page

**Goal:** Learn logical list grouping.

**Instructions:**

1. Create `recipe.html`.
2. Add `<h2>` — "Chocolate Cake Recipe".
3. Create two main lists: Ingredients (unordered) and Steps (ordered).
4. For "Steps", include 5 cooking steps in sequence.
5. Highlight important steps using `<strong>` or `<em>`.
6. output :

# Ingredients

- Flour
- Sugar
- Eggs
- Butter
- Chocolate powder

# Steps

1. Preheat the oven.
2. Mix all ingredients.
3. **Bake for 30 minutes.**
4. Let it cool.
5. *Decorate and serve!*

---

## Task 8: Create a “Bucket List” Webpage

**Goal:** Experiment with list customization and semantics.

**Instructions:**

1. Create `bucket-list.html`.
2. Add a heading “My Life Goals”.
3. Use `<ol>` for 10 items you want to achieve.
4. Use nested `<ul>` for sub-goals under a few main goals.

5. Add emojis or special characters before list items (use HTML entities or press window + . )
6. output :

1.  Travel the world
  - Europe Tour
  - Visit Japan
2.  Write a Book
3.  Become a Full-Stack Developer
  - Master React
  - Learn Node.js
4.  Build a Dream Home
5.  Learn Guitar

## Task 9: Create a “Tech Stacks” Hierarchical List

**Goal:** Practice multi-level nested lists.

**Instructions:**

1. Create `tech-stack.html`.
2. Add `<h2>` — “Full Stack Developer Skill Map”.
3. Main list: Frontend, Backend, Database, Tools.
4. Under each, add nested `<ul>` listing specific technologies.

Example:

- Frontend → HTML, CSS, JavaScript, React
- Backend → Node.js, Express
- Database → MongoDB

5. Use comments to label each hierarchy level.

6. output :

- Frontend
  - HTML
  - CSS
  - JavaScript
  - React
- Backend
  - Node.js
  - Express
- Database
  - MongoDB
  - MySQL
- Tools
  - Git
  - VS Code
  - Postman

---

### **Task 10: Create a “Comparison Table using Lists”**

**Goal:** Simulate table-style data with lists.

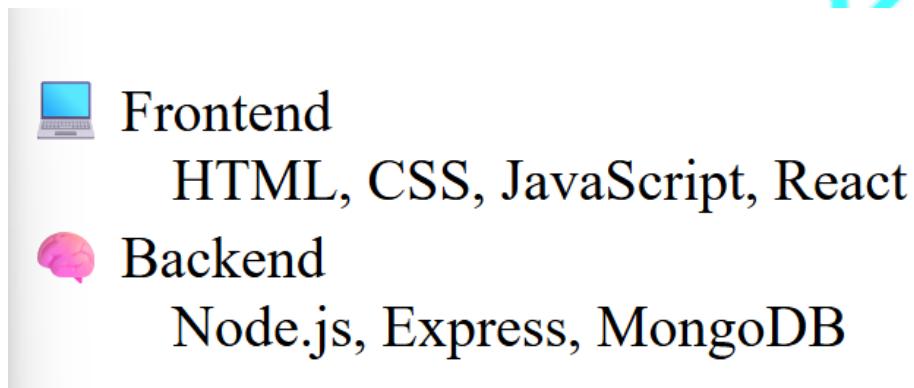
**Instructions:**

1. Create `comparison.html`.
2. Add `<h2>` — “Frontend vs Backend Technologies”.
3. Create a `<dl>` list:

- `<dt>Frontend</dt>` → `<dd>HTML, CSS, JS, React</dd>`
- `<dt>Backend</dt>` → `<dd>Node.js, Express, MongoDB</dd>`

4. Add styling or emojis to differentiate visually.

5. output :



**Extra questions :**

Preceding Text

- I. List Item 1
  - a. Nested Item 1.1
  - b. Nested Item 1.2
- II. List Item 2
  - 1. Nested Item 2.1
  - 2. Nested Item 2.2
    - o Nested Item 2.2.1
    - o Nested Item 2.2.2
      - Nested Item 2.2.2.1
      - Nested Item 2.2.2.2
    - o Nested Item 2.2.3
  - 3. Nested Item 2.3
- III. List Item 3
  - Nested Item 3.1
  - Nested Item 3.1
  - Nested Item 3.1

## Pro Tips for Students

- Use **unordered lists** for non-sequential items and **ordered lists** for steps or rankings.
- Keep lists properly **indented** to make HTML readable.
- Always **close** `</li>` **tags**, especially when nesting.
- **Description lists** are great for glossaries or key-value data.
- Test nested lists in the browser to understand visual hierarchy.
- Use HTML entities like `&bull;`, `&rarr;`, or emojis for creative list styles.