1. HTML Detailed Notes

1.1 DOCTYPE Declaration

- The very first line in an HTML document.
- Tells the browser which HTML version to use.

  For HTML5, always use:

  ```
  <!DOCTYPE html>
  ```

  hierarchy order maintained by Blink Engine

- It's **case-insensitive** but conventionally uppercase.
- Without it, browsers may switch to "quirks mode," causing inconsistent rendering.

1.2 `<html>` Tag

- Root element wrapping the entire HTML page.

  Should include a `lang` attribute specifying the language for accessibility and SEO.

  ```
  <html lang="en">
  ```

- Contains two main child elements: `<head>` and `<body>`.

1.3 `<head>` Tag

- Metadata container, not visible on the page.

  hold info about website but doesn't show anywhere

- Common content inside `<head>`:
  - `<title>`: Defines page title shown in browser tab.
  - `<meta charset="UTF-8">`: Sets character encoding.

    - Links to CSS (`<link>`) or JavaScript (`<script>`) files.
    - Other metadata like viewport settings, SEO meta tags.

  Example:

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

1.4 <body> Tag

- Contains all visible content.
- Text, images, videos, buttons, forms, etc., go inside.
- Everything inside <body> is rendered on the page.

  Example:

```
<body>
  <h1>Welcome!</h1>
  <p>This is a sample paragraph.</p>
</body>
```

1.5 Headings: <h1> to <h6>

- Represent headings with different importance.
- <h1> is the highest-level (main title), <h6> is lowest-level.
- Search engines and screen readers use these for structure and navigation.
- Don't skip heading levels (e.g., from <h1> to <h4>) for accessibility.

  Example:

```
<h1>Main Title</h1>
```

```
<h2>Section Heading</h2>
<h3>Subsection</h3>
```

1.6 Paragraph <p> Tag

- Used to define blocks of text (paragraphs).
- Browsers add vertical spacing (margin) before and after paragraphs.
- Always wrap textual content in paragraphs for semantic clarity.

  Example:

  ```
  <p>This is a paragraph of text describing something
  important.</p>
  ```

1.7 Anchor <a> Tag

- Defines hyperlinks.
- `href` attribute specifies URL.
- `target` attribute controls where the link opens:
  - `_self`: default, opens in same tab.
  - `_blank`: opens in new tab/window.
  - `_parent`, `_top`: for frames (rare).

  Example:

  ```
  <a href="https://openai.com" target="_blank">Visit OpenAI</a>
  ```

- Use descriptive link text (not just “click here”) for accessibility.

1.8 Image <img> Tag

- Embeds images.
- Attributes:
  - `src`: image file URL (required).
  - `alt`: text description for screen readers and when image fails to load (required for accessibility).
  - `width` and `height` can set image size.

  Example:

  ```
  <img src="logo.png" alt="Company Logo" width="200"
  height="100">
  ```

- Always provide meaningful alt text for users with screen readers.

1.9 `<div>` and `<span>` Tags

- **`<div>`**: Block-level container for grouping content.
- **`<span>`**: Inline container for styling parts of text or inline elements.
- Both are non-semantic — they don't convey meaning, just structure.

  Example:

  ```
  <div class="container">
    <p>This paragraph is inside a div.</p>
    <span style="color: red;">Important text</span>
  </div>
  ```

1.10 Block vs Inline Elements

- **Block elements**:
  - Take full width by default.

    - Start on a new line.
    - Examples: `<div>`, `<p>`, `<h1>`, `<ul>`, `<table>`.
- **Inline elements**:
    - Take only necessary width.
    - Flow within text.
    - Examples: `<a>`, `<span>`, `<img>`, `<strong>`.

Note: You can change display behavior using CSS (`display: block/inline/inline-block`).

1.11 Lists

- Two types:

1. **Unordered list (`<ul>`)**
    - Bulleted list.
    - `type` attribute changes bullet style (disc, circle, square). none

Example:

```
 <ul type="circle">
   <li>Apple</li>
   <li>Banana</li>
</ul>
```

2. **Ordered list (`<ol>`)**
    - Numbered list.
    - `type` attribute controls numbering style:

        - 1 (numbers), A (uppercase letters), a (lowercase letters), I (uppercase Roman numerals), i (lowercase Roman numerals).

Example:

```
 <ol type="A">
   <li>First</li>
   <li>Second</li>
</ol>
```

1.12 Tables

- Used to display tabular data.
- Structure:
    - `<table>`: container.
    - `<thead>`: table header section.
    - `<tbody>`: table body section.
    - `<tr>`: table row.
    - `<th>`: table header cell (bold, centered by default).
    - `<td>`: table data cell.
- Attribute `border="1"` adds borders around cells (simple styling).

  Example:

```
<table border="1">
  <thead>
    <tr>
      <th>Name</th>
      <th>Age</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Alice</td>
      <td>25</td>
    </tr>
  </tbody>
</table>
```

1.13 Audio Tag

- Embed sound files.
- Attributes:
  - `controls`: show play/pause UI.
  - `autoplay`: automatically starts playback (use carefully).
  - `loop`: repeats audio indefinitely.
  - `muted`: starts muted.
  - `<source>`: specify audio file and type (helps browser pick correct format).

Example:

```
<audio controls autoplay loop>
  <source src="song.mp3" type="audio/mpeg" />
  Your browser does not support audio.
</audio>
```

1.14 Video Tag

- Embed videos.
- Attributes:
  - `controls`: playback UI.
  - `autoplay`: plays automatically.
  - `loop`: repeat.

    - `poster`: image shown before video plays.
- Multiple `<source>` tags for different video formats.

  Example:

```
<video controls autoplay loop poster="thumbnail.jpg">
  <source src="video.mp4" type="video/mp4" />
  <source src="video.webm" type="video/webm" />
  Your browser does not support the video tag.
</video>
```

1.15 Semantic vs Non-Semantic Tags

- **Semantic tags**: clearly describe their meaning.
    - E.g., `<header>`, `<footer>`, `<article>`, `<section>`, `<nav>`, `<aside>`, `<main>`
    - Help search engines, accessibility tools, and developers understand the page structure.
- **Non-semantic tags**: no inherent meaning.
    - E.g., `<div>`, `<span>`
    - Used mainly for styling and grouping elements when semantic alternatives don’t fit.

1.16 Absolute vs Relative Paths

- **Absolute path**: full URL or full path from root.

  Example:

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

    - Always works regardless of file location.

- **Relative path**: relative to current file location.

  Example:

  ```
  <img src="images/photo.jpg" />
  ```

  - Easier for local projects but depends on folder structure.

---

2. CSS Detailed Notes

2.1 Inline CSS

- Apply CSS directly inside an HTML tag using the `style` attribute.
- Not recommended for large projects (hard to maintain).

  Example:

  ```
  <p style="color: blue; font-size: 18px;">Blue text</p>
  ```

2.2 Color Formats

Named colors: easy to use but limited.

```
color: red;
```

- **Hexadecimal**: 6-digit base-16 codes, start with #.
  - `#ff0000` is red.
  - Shorthand: `#f00` same as `#ff0000`.
- **RGB**: `rgb(red, green, blue)`
  - Values between 0-255.
  - Example: `rgb(255, 0, 0)` is red.

- **RGBA**: RGB with alpha (opacity).
  - Fourth parameter: 0 (transparent) to 1 (opaque).
  - Example: `rgba(255, 0, 0, 0.5)` is semi-transparent red.

2.3 Font Properties

- `font-size`: controls text size.
  - Units: `px`, `em`, `%`, `rem` (best for responsiveness).
- `font-weight`: thickness of text.
  - Values: `normal`, `bold`, or numeric `100-900`.
- `font-style`: normal, italic, oblique.
- `font-family`: specify font type.
  - List fallback fonts separated by commas.

Example:

```
font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;
```

2.4 Text Transformation & Alignment

- `text-transform`: changes letter case.
  - `uppercase`, `lowercase`, `capitalize`
- `text-align`: horizontal alignment.
  - `left`, `right`, `center`, `justify`

2.5 Background Properties

- `background-image`: set an image as background.
- `background-size`: control scaling (`cover`, `contain`, or specific sizes).
- `background-repeat`: repeat behavior (`repeat`, `no-repeat`, `repeat-x`, `repeat-y`).
- `background-position`: position background image (e.g., `center`, `top left`).

  Example:

  ```
  background-image: url('bg.jpg');
  background-size: cover;
  background-repeat: no-repeat;
  background-position: center;
  ```

2.6 Borders & Padding

- `border`: shorthand for width, style, and color.
  - Example: `border: 1px solid black;`
- `padding`: space inside element between content and border.
  - Can specify all sides or individually (`padding-top`, etc.)

2.7 Table Border Collapse

- By default, table borders double up between cells.
- Use `border-collapse: collapse;` to merge borders into one.

  Example:

  ```
  table {
    border-collapse: collapse;
  }
  ```

2.8 Text Shadow

- Adds shadow behind text for visual depth.
- Syntax: `text-shadow: horizontal-offset vertical-offset blur-radius color;`

Example:

```
text-shadow: 2px 2px 5px rgba(0,0,0,0.3);
```

---

get: limited data, directly exposed in url ,saved in url
post: can more data ,privacy,data in browser body