HTML (HyperText Markup Language): HTML is a markup language used to create the structure of web pages. It consists of elements represented by tags, which enclose content and determine how it is displayed in the browser. These tags define headings, paragraphs, links, images, forms, and other web elements. HTML is the foundation of web content and helps browsers understand what to display

CSS (Cascading Style Sheets): **CSS is used to control the presentation and layout of HTML elements. It allows you to style and format the structure defined in HTML, controlling aspects like colors, fonts, spacing, and positioning. CSS styles can be applied inline, in the `<head>` of an HTML document, or in external stylesheets.**

HTML Heading Elements:

HTML heading elements are used to define headings in a document, ranging from `<h1>` to `<h6>`, with `<h1>` being the most important and `<h6>` the least. These elements help structure the content and make it easier for search engines and users to understand the document hierarchy.

---

2. HTML Paragraph Elements:

The `<p>` element is used to define paragraphs of text. It’s a block-level element that automatically provides spacing around the text and is used to group related sentences or content together in a logical manner.

---

3. Self-Closing Tags:

Self-closing tags are HTML elements that do not have closing tags because they don’t wrap content. These elements are typically used for things like line breaks, images, and meta information, and end with a slash (e.g., `<br />`, `<img />`).

---

4. HTML List Elements:

HTML list elements are used to group related items in a list format. There are two main types of lists: ordered lists (`<ol>`) which create numbered lists, and unordered lists (`<ul>`) which create bulleted lists. Each item in a list is contained within a `<li>` (list item) element.

---

5. Nesting and Indentation:

Nesting refers to placing HTML elements inside other elements, creating a hierarchical structure. Proper indentation is used to improve readability by visually representing the relationships between elements (e.g., child elements indented under parent elements).

---

6. Anchor Elements:

The `<a>` element is used to create hyperlinks, allowing users to navigate from one page to another or to different sections within the same page. The `href` attribute specifies the destination of the link, which can be a URL, an email address, or an anchor within the same page.

---

7. Image Elements:

The `<img>` element is used to embed images in an HTML document. It’s a self-closing tag, and its primary attributes include `src` (source of the image) and `alt` (alternative text, describing the image for accessibility and SEO purposes).

Webpages:

A webpage is a document on the World Wide Web that is displayed in a web browser. It is typically made using HTML, CSS, and JavaScript and can contain text, images, links, multimedia content, and interactive elements. A collection of linked webpages forms a website.

---

2. HTML Boilerplate:

An HTML boilerplate refers to the basic structure of an HTML document that includes essential tags like `<!DOCTYPE html>`, `<html>`, `<head>`, and `<body>`. It serves as a starting template for web development, ensuring that necessary meta tags and document setup are in place.

---

3. How to Add CSS:

CSS can be added to HTML in three ways:

- **Inline CSS**: Using the `style` attribute within an HTML element.
- **Internal CSS**: Adding styles within a `<style>` tag inside the HTML `<head>`.
- **External CSS**: Linking to an external stylesheet using a `<link>` tag in the HTML `<head>`.

4. CSS Selectors:

CSS selectors define which HTML elements to style. Common types include:

- **Element Selector**: Targets elements by their tag name (e.g., `div`, `p`).
- **Class Selector**: Targets elements with a specific class (e.g., `.classname`).
- **ID Selector**: Targets a single element with a specific ID (e.g., `#idname`).
- **Attribute Selector**: Targets elements with a specific attribute (e.g., `[type="text"]`).

5. CSS Colours:

CSS colors can be specified in various formats:

- **Named Colors**: Predefined color names (e.g., `red`, `blue`).
- **Hexadecimal**: A 6-digit hexadecimal code (e.g., `#FF5733`).
- **RGB/RGBA**: Specifies colors using Red, Green, Blue (e.g., `rgb(255,87,51)`, with alpha for transparency `rgba(255,87,51,0.5)`).
- **HSL/HSLA**: Uses Hue, Saturation, and Lightness (e.g., `hsl(0, 100%, 50%)`).

6. Font Properties:

CSS font properties allow you to style text. Key properties include:

- **font-family**: Specifies the typeface (e.g., Arial, Times New Roman).
- **font-size**: Controls the size of the text.
- **font-weight**: Specifies the thickness of the font (e.g., bold, normal).
- **font-style**: Controls whether the text is italic, normal, etc.
- **line-height**: Defines the space between lines of text.

7. Inspecting CSS:

Web browsers provide developer tools (right-click → "Inspect") that allow you to view and modify HTML and CSS in real-time. You can see which styles are applied, check for errors, and experiment with new styles directly in the browser.

8. CSS Box Model:

The CSS box model describes the rectangular boxes generated for elements. It consists of:

- **Content**: The actual content of the element (e.g., text or image).
- **Padding**: The space between the content and the border.
- **Border**: A line that wraps around the padding and content.
- **Margin**: The space outside the border that separates the element from others.

---

9. Cascade:

The "cascade" in CSS refers to how styles are applied when multiple rules target the same element. The browser prioritizes styles based on specificity, importance (`!important`), and the order in which the rules are written.

---

10. Combining CSS Selectors:

Multiple CSS selectors can be combined to target elements more precisely:

- **Descendant Selector**: Targets elements inside other elements (e.g., `div p`).
- **Group Selector**: Combines multiple selectors (e.g., `h1, h2, h3`).
- **Child Selector**: Targets direct children of an element (e.g., `div > p`).

---

11. CSS Positioning:

CSS positioning controls the layout of elements on the page. Types include:

- **Static**: Default positioning, follows the normal document flow.
- **Relative**: Positioned relative to its normal position.
- **Absolute**: Positioned relative to its nearest positioned ancestor.
- **Fixed**: Positioned relative to the browser window.
- **Sticky**: Toggles between relative and fixed based on scroll position.

---

12. CSS Display:

The `display` property determines how an element is displayed on the page. Common values include:

- **block**: Element takes up the full width available and starts on a new line.
- **inline**: Element flows with the text and doesn’t break the line.
- **inline-block**: Behaves like `inline` but can have width and height.
- **none**: Element is not displayed.

---

13. CSS Float:

The `float` property allows an element to float to the left or right of its container, allowing text and inline elements to wrap around it. Common float values are `left`, `right`, and `none`.

---

14. Media Queries:

Media queries are used to apply CSS rules based on the device's characteristics (e.g., screen size, orientation). They enable responsive design by specifying different styles for different screen sizes or conditions.

---

15. Display Flex:

`display: flex;` turns an element into a flex container, allowing its children to be arranged in a flexible, responsive layout. It provides control over how child elements grow, shrink, and are spaced.

---

16. Flex Direction:

The `flex-direction` property defines the direction in which the flex container’s children are laid out. Possible values:

- **row**: Aligns items horizontally (default).
- **column**: Aligns items vertically.
- **row-reverse**: Aligns items horizontally in reverse.
- **column-reverse**: Aligns items vertically in reverse.

---

17. Flex Layout:

Flex layout refers to the flexible arrangement of items inside a container using `display: flex`. It adjusts the size and position of items automatically to fit the available space, making it ideal for responsive design.

---

18. Flex Sizing:

Flex sizing is controlled using the `flex-grow`, `flex-shrink`, and `flex-basis` properties:

- **flex-grow**: Defines how much an item should grow relative to others.
- **flex-shrink**: Defines how much an item should shrink relative to others.
- **flex-basis**: Specifies the initial size of the item before growing or shrinking.

---

19. Grid:

CSS Grid is a powerful layout system that divides a page into rows and columns. It enables precise control over layout and positioning of elements in a grid structure, ideal for complex layouts.

---

20. Grid Sizing:

Grid sizing defines how grid cells (rows and columns) are sized. You can use units like `fr` (fractional units), percentages, or fixed sizes (e.g., `px`, `em`) to control the grid layout.

---

21. Grid Placement:

Grid placement determines where an element is placed within a grid container using properties like `grid-row`, `grid-column`, `grid-area`, and `grid-template`. This allows precise control over which row and column an element occupies.