1.Semantic elements in HTML? = akseman

Semantic elements in HTML will provide meaning to the content . These elements clearly describe their purpose and the type of content they contain, both to the browser and to developers. Examples of semantic elements include <header>, <footer>, <article>, <section>, <nav>, and <aside>.

Advantages of Using Semantic Elements
Semantic elements improve the accessibility of web pages by providing context to screen readers. This allows users with disabilities to navigate and understand the content more easily.
Search engines better understand the structure and content of a webpage when semantic elements are used. This improved better search engine freindly
Semantic elements make HTML code more readable and understandable. Developers can quickly find the structure and purpose of the different parts of a webpage, which makes maintaining and updating the code easier.

2.difference between Local storage and session storage and cookie? = aklsc

Local Storage
in Local Storage the Data is stored per domain and it is accessible by any page from that domain.
here Data persists until explicitly deleted by the user. It remains even after the browser is closed and reopened.
in Local Storage the we can store the data upto 5MB.
it is useful for storing long-term data that does not need to be sent to the server with every request, like user preferences, themes, or other settings.

Session Storage
in Session Storage the Data is stored per window or tab and is only accessible within that specific window or tab.
we will lost the data if we close the browser window or Tab.
in Session Storage we can store the data upto 5MB to 10MB.
it is useful for storing temporary data that only needs to persist during a single session, like form data being filled out, temporary states, or data that should not be available in different tabs.

Cookies
in cookie Data is stored per domain and can be accessed by any page from that domain. Cookies can also have a specific path to limit their scope within the domain.
Lifetime Can be set to expire at a specific date or after a certain period. Cookies can also be session-based (cleared when the browser is closed).
in cookie Typically we can store the data upto 4KB.
Mainly used for sending small pieces of data to the server with each request, such as session identifiers, authentication tokens, and user tracking. Cookies are the only option when we need the data to be sent to the server automatically with HTTP requests.

3.web workers? = akwebwork

Web Workers allows us for running JavaScript scripts in the background, independently of the user interface. This enables web applications to perform tasks asynchronously, without interfering with the user interface.

1.Heavy Computation:

Example: Image processing, data analysis, complex mathematical calculations.
Benefit: Offloading these tasks to a Web Worker prevents the main thread from freezing, ensuring a responsive UI.

Background Data Fetching:

Example: Fetching large datasets or making multiple network requests simultaneously.
Benefit: Improves performance by handling data fetching and processing in the background.

4.WebSocket API? = aksoc

WebSocket API make possible to open a two-way interactive communication session between the user's browser and a server. With this API, we can send messages to a server and receive event-driven responses without having to poll the server for a reply.

Use Cases for WebSockets

Example: Chat applications, live notifications, real-time collaboration tools.
Benefit: Provides instant updates and allows us to communicate in real time without refreshing the page.

Live Data Feeds:

Example: Financial market data, live sports scores, social media feeds.
Benefit: Keeps the data updated in real time, providing us with the most current information without delay.

5.Canvas and SVG? = aksvg

SVG:

SVG stands for Scalable Vector Graphics
SVG is used to define vector-based graphics for the Web
SVG defines graphics in XML format
Each element and attribute in SVG files can be animated
SVG has several methods for drawing paths, rectangles, circles, polygons, text.

Canvas:
<canvas> element is used to draw graphics on a web page.
The <canvas> element is only a container for graphics. we must use JavaScript to actually draw the graphics.
Canvas has several methods for drawing paths, boxes, circles, text, and adding images.
Canvas is supported by all major browsers.
canvas is a rectangular area on an HTML page. By default, a canvas has no border and no content.

Differences Between SVG and Canvas

SVG is a language for describing 2D graphics in XML, while Canvas draws 2D graphics, on the fly (with JavaScript).
SVG is XML based, which means that every element is available within the SVG DOM. we can attach JavaScript event handlers to SVG graphics.
In SVG, each drawn shape is remembered as an object. If attributes of an SVG object are changed, the browser can automatically re-render the shape.
Canvas is rendered pixel by pixel. In canvas, once the graphic is drawn, it is forgotten by the browser. If its position should be changed, the entire scene needs to be redrawn, including any objects that might have been covered by the graphic.
svg is Resolution independent where as canvas Resolution dependent.

6.HTML entities? = akent

HTML entities are special codes used in HTML to define characters which have a special meaning in HTML or are not easily typed or displayed. They are used to display symbols, special characters, and reserved characters in HTML documents.

Examples:
© represents © (copyright symbol)
  non-breaking space

7.difference between class selector and ID selector? = akid

Classes can applied to multiple elements, while IDs should be unique and only applied to one element.
Classes are denoted by a dot (.) followed by the class name, while IDs are denoted by a hash (#) followed by the ID name.
IDs have a higher specificity than classes, meaning styles applied through IDs take precedence over styles applied through classes.
Classes are used for styling groups of elements or applying reusable styles, while IDs are used to uniquely identify and target specific elements.

8.what is DOCKTYPE in html?if we forgot in html what happen explain?
DOCTYPE declaration in HTML (Document Type Declaration) is a element it specifies the version of HTML being used in a document. It tells web browsers how to interpret and render the HTML content. If we forget to include the DOCTYPE declaration in an HTML document, it can lead to several issues.

With DOCTYPE: Browsers render the document in standards mode, which ensures consistent rendering and layout based on the specified HTML version.
Without DOCTYPE: Browsers may render the document in compatibility mode, which attempts to give the rendering behavior of older browsers. This could be inconsistencies in layout and styling, especially when dealing with CSS.

With DOCTYPE: HTML validators and parsing tools accurately check the document for syntax errors and compliance with HTML standards.
Without DOCTYPE: Validation tools may not work correctly, making it challenging to identify and resolve HTML errors.

9.Block-Level Elements and Inline level Elements?

Block-Level Elements

Block-level elements take up the full width available (their parent container), no matte of their content.
Each block-level element starts on a new line, means the content before and after a block-level element will be on separate lines.
They can contain other block-level elements and inline elements.
Block-level elements can have width and height. By default, their width is 100% of their parent container.
Block-level elements have all margin and padding properties.

Examples: div, p, heading tags like h1,h2,h3,h4,h5,h6,ol,ul.

Inline-Level Elements

Inline elements take up only as much width as necessary for their content.
Inline elements do not break lines before or after the element.
They cannot contain block-level elements.
Inline-level elements only take as much width as their content requires, and height is determined by the line-height property.
Inline-level elements only respect horizontal (left and right) margins and padding, not vertical (top and

bottom).

Examples: , anchor tag, img tag

10.Meta tags?

meta tag defines metadata about an HTML document. Metadata is data (information) about data.
meta tags always go inside the head element, and are used to specify character set, page description, keywords, author of the document, and viewport settings.
Metadata will not be displayed on the page, but is machine parsable.
Metadata is used by browsers like how to display content or reload page, search engines (keywords), and other web services.
view port allows us to take control over the viewport (the user's visible area of a web page), through the <meta> tag (See "Setting The Viewport" example below).

11.HTML Attributes

HTML attributes provide additional information about HTML elements.
All HTML elements can have attributes
Attributes provide additional information about elements
Attributes are always specified in the start tag
Attributes usually come in name/value pairs like: name="value"

12.what are empty and void tags in html?

Empty and void tags in HTML are elements that do not have any content between their opening and closing tags. These tags are self-closing, meaning they do not require an end tag.

Examples:elements like line breaks, images, or meta-information. like
,<hr>,<meta>

13.What is Accessibility?

Accessibility refers to the design and implementation of products, devices, services, or environments to be usable by people with disabilities. accessibility ensures that websites and web applications are usable by as many people as possible, including those with visual, auditory, motor, or cognitive disabilities.

Using semantic HTML elements ensures that the content structure is understandable to screen readers.
ARIA roles & attributes help us accessibility for dynamic content.
Ensure that all interactive elements are accessible via keyboard.