Que1 :What happens in the background when you type URL on the browser.

1. Enter a URL into a web browser.
2. The browser looks up the IP address for the domain name via DNS.
3. The browser sends a HTTP *request* to the server.
4. The server sends back a HTTP *respons.*
5. The browser begins rendering the HTML.
6. The browser sends requests for additional objects embedded in HTML (images, css, JavaScript) and repeats steps 3-5.
7. Once the page is loaded, the browser sends further a sync requests as needed.

Que 2:Why HTML5 is used?

Hypertext Markup Language revision 5 (HTML5) is markup language for the structure and presentation of World Wide Web contents. HTML5 supports the traditional HTML and XHTML-style syntax and other new features in its.

HTML 5 allows the web developer and web designer to use neater code and remove div tag and replace all the div tags with new HTML 5 elements. HTML5 also offers offline browsing, means that visitors can load certain elements on a webpage without an active internet connection.

Web languages need regular upgrades in order to stay current and solve new problems faced by web developers. HTML5 is the latest version of HTML.

Que 3:Why HTML is created?

* First developed by [Tim Berners-Lee](https://www.computerhope.com/people/tim_berners-lee.htm) in [1990](https://www.computerhope.com/history/1990.htm).
* HTML is short for Hypertext Markup Language.
* HTML is used to create electronic documents or pages that are displayed on the [World Wide Web](https://www.computerhope.com/jargon/w/www.htm). Each page contains a series of connections to other pages called [hyperlinks](https://www.computerhope.com/jargon/h/hyperlink.htm).
* HTML code ensures the proper formatting of text and images so that your [Internet browser](https://www.computerhope.com/jargon/b/browser.htm) may display them as they are intended to look. Without HTML, a browser would not know how to display text as [elements](https://www.computerhope.com/jargon/h/html-element.htm) or load images or other elements.

Que 4: About semantics**.**

In HTML, for example, the [<h1>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/h1) element is a semantic element, which gives the text it wraps around the role (or meaning) of "a top level heading on your page."

* Semantic naming mirrors proper custom element/component naming.
* Search engines will consider its contents as important keywords to influence the page's search rankings.
* Finding blocks of meaningful code is significantly easier than searching though endless div’s with or without semantic or name spaced classes.
* Screen readers can use it as a signpost to help visually impaired users navigate a page.
* Suggests to the developer the type of data that will be populated.

Semantic elements.

<article>

<aside>

<figure>

<footer>

**Syntax**

* Syntax is the proper order of words in a phrase or sentence.
* Syntax is a tool used in writing proper grammatical sentences.
* When referring to a **programming** language, **syntax** is a set of rules for grammar and spelling. In other words, it means using character structures that a computer can interpret.
* For **example**, if a user tries to execute a command without proper **syntax**, it generates **syntax** error, usually causing the **program** to fail.

Que 5: What are all the CSS Grid methods.

**Base method.**

The basic way to create a grid is to declare **grid-template-column.** The Grid items will automatically populate the grid container from top left to bottom right, based on the HTML source order, and will add rows as necessary.

# List Method.

Ordered & unordered. Using the “**base method**” and the **repeat() function,** we can easily achieve a list sequence of items.

# Dynamic List Method.

CSS Grid, the **repeat() function**and the **minmax() function.** This is similar to the “**List Method**”, except that in the “**Dynamic List Method**”, the number of items in each row is dynamic, and may change according to the width of the view-port.

# Positioning Method.

In every grid we create, we also create grid lines automatically. Using these grid lines, we can position the grid items in any square we want.

When positioning a grid item, we define 4 grid lines, grid-column-start, grid-column-end, grid-row-start and grid-row-end.

# Areas Method.

**Areas Method is the most important grid method of all.** This utilization of CSS grid is basically the main reason CSS gird was invented. It’s the easiest way to create complex grid systems.

# Carousel Method.

This method works like a trick. In CSS grid, the default flow of the items is to create a new row when there isn’t enough space in the current row.

**CSS Box model:**

The full CSS box model applies to block boxes, inline boxes only use some of the behavior defined in the box model. The model defines how the different parts of a box -margin, border, padding, and content - work together to create a box that you can see on the page.

* **Content box**: The area where your content is displayed, which can be sized using properties like [width](https://developer.mozilla.org/en-US/docs/Web/CSS/width) and [height](https://developer.mozilla.org/en-US/docs/Web/CSS/height).
* **Padding box**: The padding sits around the content as white space; its size can be controlled using [padding](https://developer.mozilla.org/en-US/docs/Web/CSS/padding) and related properties.
* **Border box**: The border box wraps the content and any padding. Its size and style can be controlled using [border](https://developer.mozilla.org/en-US/docs/Web/CSS/border) and related properties.
* **Margin box**: The margin is the outermost layer, wrapping the content, padding and border as whitespace between this box and other elements. Its size can be controlled using [margin](https://developer.mozilla.org/en-US/docs/Web/CSS/margin) and related properties.

**HTML Dom**

The **HTML DOM** is an **Document Object Model** for **HTML**. It defines:

* HTML elements as **objects.**
* **Properties** for all HTML elements.
* **Methods** for all HTML elements.
* **Events** for all HTML elements.