Q What is CSS?

Ans: CSS stands for Cascading Style Sheets. CSS is used to define styles for web pages. It controls the look and feel of the website. it helps to build responsive websites.

Q What is mean by cascade in CSS?

Ans: Cascade means in which order the CSS is being apply. There is one Cascade rule in CSS that whichever style will come below that will going apply. Below styles will override above styles if they conflict each other.

Q what is specificity in CSS?

Ans: Specificity is like priority which is given to the selectors. Whichever selector preference is higher they will apply first. Inline CSS has the highest priority and overrides all other selectors. Then id, class, element.

Q why do we use CSS?

Ans:
CSS saves time**:** You can write CSS once and reuse the same sheet on multiple HTML pages.
Easy Maintenance**:** To make a global change simply change the style, and all elements in all the webpages will be updated automatically.
Search Engines**:** CSS is considered a clean coding technique, which means search engines won't have to struggle to "read" its content.
Offline Browsing**:** CSS can store web applications locally with the help of an offline cache. Using of this we can view offline websites.

Q what are the advantage of CSS?

Ans:
CSS plays an important role, by using CSS you simply got to specify a repeated style for an element once & use it multiple times because CSS will automatically apply the required styles.
It simplifies maintenance as a change of one line of code affects the whole website and maintenance time.
CSS changes are device friendly. With people employing a batch of various range of smart devices to access websites over the web, there's a requirement for responsive web design.

Q lists of CSS frameworks.

Ans: tailwind, Bootstrap, Material UI, Materialize.

Q What is the syntax for CSS?

Ans: A CSS style rule consists of a selector, property, and its value. The selector points to the HTML element where CSS style is to be applied. The CSS property is separated by semicolons.

Syntax**:**

```
selector {
   Property: value;
}
```

Q What does the 'a' in rgba mean?

Ans: RGBA contains A (Alpha) which specifies the transparency of elements.

Q What are CSS HSL Colors?

HSL: HSL stands for Hue, Saturation, and Lightness respectively.

- **Hue:** Hue is the degree of the color wheel. Its value lies between 0 to 360 where 0 represents red, 120 represents green and 240 represents a blue color.
- **Saturation:** It takes a percentage value, where 100% represents completely saturated, while 0% represents completely unsaturated (gray).
- **Lightness:** It takes a percentage value, where 100% represents white, while 0% represents black.

Q What are CSS backgrounds, list the properties?

Ans:

1. **background-color:** This property specifies the background color of an element.
2. **background-image:** This property specifies an image to use as the background of an element. By default, the image is repeated so it covers the entire element.
3. **background-repeat:** By default, the background image property repeats the image both horizontally and vertically.
4. **background-attachment:** This property is used to fix the background ground image. The image will not scroll with the page.
5. **background-position:** This property is used to set the image to a particular position.

Q What is the difference between margin and padding?

Ans:

- Margin is used to create space around elements and padding is used to create space around elements inside the border.
- We can set the margin property to auto but we cannot set the padding property to auto.
- In Margin property we can allow negative or float number but in padding we cannot allow negative values.
- Margin and padding target all the 4 sides of the element. Margin and padding will work without the border property also.

Q What is CSS Box Model?

Ans: Every element in html is represented in the form of the box. The CSS box model is a container that contains multiple properties including borders, margin, padding, and the content itself. It is used to create the design and layout of web pages.

Q What is the difference between CSS border and outline?

Ans:

- **CSS border** properties allow us to set the style, color, and width of the border.
- **CSS outline** property allows us to draw a line around the element, outside the border.
- Unlike borders, outlines don't allow us to set each edge to a different width, or set different colors and styles for each edge. An outline is the same on all sides.
- Outlines cannot be circular.
- Outlines do not take up space, because they are always placed on top of the box of the element.

Q What is inheritance?
Ans: If we give style to the parent then its child will inherit all the styles. For ex. If we give font family to the parent element then child element inherits all the styles.

Q difference between em and rem?

Ans: em means refer to the parent and rem means refer to the root element which is html. By default, font size of html is 16px.

Q What are the different CSS link states?

Ans: A link is a connection from one web page to another web page. CSS property can be used to style the links in various different ways.
States of Link: Before discussing CSS properties, it is important to know the states of a link. Links can exist in different states and they can be styled using pseudo-classes.
There are four states of links given below:

- **a:link:** This is a normal, unvisited link.
- **a:visited:** This is a link visited by a user at least once
- **a:hover:** This is a link when the mouse hovers over it
- **a:active:** This is a link that is just clicked.

Q Can we add an image as a list item marker?
Ans: To add an image as the list-item marker in a list, we use the list-style-image property in CSS.
Syntax:

```
list-style-image: none | url | initial | inherit;
```

Q what is the difference between display: none and visibility: hidden?

Ans:

- **visibility: hidden** simply hides the element but it will occupy space and affect the layout of the document.
- **display: none** removes the element from the normal layout flow (causes DOM reflow). It will not affect the layout of the document nor occupy space.

Q What are the various positioning properties in CSS?

Ans:
1. Fixed: It will be positioned relative to the viewport. An element with fixed positioning allows it to remain at the same position even as we scroll the page. We can set the position of the element using the top, right, bottom, and left.
2. Static: This method of positioning is set by default. If we don't mention the method of positioning for any element, the element has the **position: static** method by default.
3. Relative: It will move the element with respective to its static position. The values that you specify will try to bring changes according to its static position.
4. Absolute: It will be positioned with respect to its parent. If the parent is static then it will come out of the document flow.
5. Sticky: Element with **position: sticky** and **top: 0** played a role between **fixed & relative** based on the position where it is placed. If the element is placed in the middle of the document, then when the user scrolls the document, the sticky element starts scrolling until it touches the top. When it touches the top, it will be fixed at that place in spite of further scrolling. We can stick the element at the bottom, with the **bottom** property.

Q What is CSS overflow?

Ans: The CSS overflow controls the big content. It tells whether to clip content or to add scroll bars. The overflow contains the following property:

- visible
- hidden
- scroll
- auto

1. Visible: The content is not clipped and is visible outside the element box.
2. Hidden: The overflow is clipped and the rest of the content is invisible.
3. Scroll: The overflow is clipped but a scrollbar is added to see the rest of the content. The scrollbar can be horizontal or vertical.
4. Auto: It automatically adds a scrollbar whenever it is required.
Overflow-x and Overflow-y: This property specifies how to change the overflow of elements. x deals with horizontal edges and y deals with vertical edges.

Q What is the purpose of the box-sizing property?

Ans: The box-sizing CSS property sets how the total width and height of an element is calculated.

- **content-box**: Content box contains the content width and height of box. Border and padding are excluded in the content box.
- **padding-box**: Width and height values apply to the element's content and its padding. The border is added to the outside of the box. Currently, only Firefox supports the padding-box value.
- **border-box**: Border box contains Width and height of the content, padding, and border.
- The box-sizing: border-box property ensures that the padding and border are included within the element's specified width and height, preventing unexpected element sizing and layout issues.
- **inherit**: inherits the box sizing of the parent element.

Q What is CSS flexbox?

Ans: The Flexible Box Layout Module, makes it easier to design flexible responsive layout structure without using float or positioning. Flexbox makes it simple to align items vertically and horizontally using rows and columns.
Flex Properties:

- flex-direction
- flex-wrap
- flex-flow
- justify-content
- align-items
- align-content

Q What is CSS grid?

Ans: It is a CSS property that offers a grid-based layout system, with rows and columns, making it easier to design web pages without floats and positioning.

Q Difference between grid and flexbox.

Ans:

- CSS Grid Layout is a **two-dimensional** system, meaning it can handle both columns and rows, unlike flexbox which is largely a **one-dimensional** system (either in a column or a row).
- A core difference between CSS Grid and Flexbox is that — CSS Grid's approach is **layout-first** while Flexbox' approach is **content-first**. If you are well aware of your content before making layout, then blindly opt for Flexbox and if not, opt for CSS Grid.
- Flexbox layout is most appropriate to the components of an application (as most of them are fundamentally linear), and **small-scale** layouts, while the Grid layout is intended for **larger-scale** layouts which aren't linear in their design.
- If you only need to define a layout as a row or a column, then you probably need flexbox. If you want to define a grid and fit content into it in two dimensions — you need the grid.

Q What is the difference between Pseudo-classes and pseudo-elements?
Ans: pseudo-classes in CSS are used to apply styles to on a specific state of an element.

a) Dynamic pseudo-classes:

- :link
- :visited
- :hover
- :active
- :focus

b) UI element states pseudo-classes:

- :enabled
- :disabled
- :checked

c) Structural pseudo-classes:

- :first-child
- :nth-child(n)
- :nth-last-child(n)
- :nth-of-type(n)
- :nth-last-of-type(n)
- :last-child
- :first-of-type
- :last-of-type
- :only-child
- :only-of-type
- :root
- :empty

d) Other pseudo-classes:

:not(x) :target :lang(language)

Pseudo-elements:

Pseudo-elements in CSS are used to apply styles on a specific parts of an element. They create virtual elements that are not present in the document tree but can be styled as if they were.

- ::before
- ::after
- ::first-letter
- ::first-line
- ::selection