**CSS INTERVIEWBIT QUESTION**

1. **What is CSS?**

CSS stands for Cascading Style Sheet. It’s a style sheet language that determines how the elements/contents in the html page are looked or display on UI .

CSS is used to give style for your web pages, including the design colors, animation, layout and variation in display for different devices and screen sizes so that website look more attractive **.**

CSS was developed and is maintained by the World Wide Web Consortium (W3C). It was first released on December 17, 1996.

1. **What is the Box model in CSS? Which CSS properties are a part of it?**

All the Html element can be considered as a boxes . A box is a rectangle area wch is wrapped around every HTML element is box model.

The box model is used to determine the height and width of the rectangular box.

The CSS Box consists of Content, padding, borders, margin.

* **Content:**  Actual Content of the box where the text or image is placed.
* **Padding:** Area surrounding the content ( and the Space between the border and content is also a padding ).
* **Border:** the Area surrounding the padding.
* **Margin:** the Area surrounding the border

**3). What are the advantages of using CSS?**

The main advantages of CSS are given below:

**Separation of content from presentation** - CSS provides a way to present the same content in multiple presentation formats in mobile or desktop or laptop.­­

**Easy to maintain** and **Reusability**: By centralizing styles in external CSS files, you can easily update and maintain the design of your website. you can define styles once and apply them to multiple elements throughout your website. This provide easy to maintain design and layout, and reduces code duplication, so Reusability also happened. it make easier to update styles globally by modifying a single CSS rule.­

**Bandwidth** - CSS, style sheets will be stored in the browser cache and they can be used on multiple pages, without having to download again. It reducing the amount of data that needs to be downloaded for subsequent page loads. This improves website performance and load times.

4) **What is the difference between block, inline, inline-block, ?**

Block Element: The block elements always start on a new line. They will take space row or width of container or web page. List of block elements are <div>, <p>. <h1> to <h6>

Inline Elements: Inline elements don't start on a new line, they appear on the same line as the content and tags beside them. Some examples of inline elements are <a>, <span> , <strong>, and <img> tags.

Inline Block Elements: Inline-block elements are similar to inline elements, except they can have padding and margins and set height and width values also.

**5) What is VH/VW (viewport height/ viewport width) in CSS?**

In CSS, vh (viewport height) and vw (viewport width) are relative units of measurement. it I used to measure the height and width in percentage with respect to the viewport of the web page(viewport: visible area of web page) .

The measure VH is equal to 1/100 of the height of the viewport. If the height of the browser is 1000px, 1vh is equal to 10px. Similarly, if the width OF browser is 1000px, then 1 vw is equal to 10px.

6) **What are the different types of Selectors in CSS?**

CSS provides various types of selectors that allow you to target specific HTML elements or groups of elements. Here are some commonly used CSS selectors:

**Element Selector**: Selects elements based on their HTML tag name.

Example:

p {

color: blue;

}

<p> hello world </p>

**Class Selector**: Selects elements based on the value of their class attribute name.(class name)

By class attribute you can select multiple element .

Example:

.box {

padding: 10px;

margin: 10px;

color : orange

width: 240px;

}

<div class="box"> this is a div container</div>

**ID Selector**: Selects a single element based on its unique id attribute value.

Example:

#container {

width: 960px;

margin: 0 auto;

}

<div id="container"></div>

**Attribute Selector:** Selects elements based on the presence or value of their attributes.

Example: input[type="text"] { border: 1px solid gray; }

**Descendant Selector**: The descendant selector select all elements that are descendants (jo uske under hai or vansh hai) of a specified element.

Example:

div p {

background-color: yellow;

}

<div>

<p>Paragraph 1 in the div.</p>

<p>Paragraph 2 in the div.</p>

<section><p>Paragraph 3 in the div.</p></section>

</div>

<p>Paragraph 4. Not in a div.</p>

<p>Paragraph 5. Not in a div.</p>

output

Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3 in the div.

Paragraph 4. Not in a div.

Paragraph 5. Not in a div.

**Notes: its select paragraph 1,2,3 , talking about 3 is in section tag but its under also div tag dts y its also selected. but not 4,5 coz 4,5 div tag k under nhi hai .**

**Child Selector**: >The child selector (>) selects all elements that are the children of a specified element .

Ex:

div > p {

background-color: red;

}

<div>

<p>Paragraph 1 in the div.</p>

<p>Paragraph 2 in the div.</p>

<section>

<!-- not Child but Descendant -->

<p>Paragraph 3 in the div (inside a section element).</p>

</section>

<p>Paragraph 4 in the div.</p>

</div>

<p>Paragraph 5. Not in a div.</p>

<p>Paragraph 6. Not in a div.</p>

**Output**

Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3 in the div (inside a section element).

Paragraph 4 in the div.

Paragraph 5. Not in a div.

Paragraph 6. Not in a div.

**Notes: here paragraph 1,2,4 are select and apply red color coz they are child of div coz paragraph 3 is under section and 5,6 is outside of div tag .**

**Adjacent Sibling Selector**: The adjacent sibling selector is used to select an element that is directly after another specific element (specified element is div) and rest is skip and so on .

Sibling elements must have the same parent element, and "adjacent" means "immediately following".

Example:

div + p {

background-color: yellow; }

<div>

<p>Paragraph 1 in the div.</p>

<p>Paragraph 2 in the div.</p>

</div>

<p>Paragraph 3. After a div.</p>

<p>Paragraph 4. After a div.</p>

<div>

<p>Paragraph 5 in the div.</p>

<p>Paragraph 6 in the div.</p>

</div>

<p>Paragraph 7. After a div.</p>

<p>Paragraph 8. After a div.</p>

Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3. After a div.

Paragraph 4. After a div.

Paragraph 5 in the div.

Paragraph 6 in the div.

Paragraph 7. After a div.

Paragraph 8. After a div.

Note: har ek div k finish k baad first paragraph element select. dts y paragraph 3,7 is selected .

General Sibling Selector: Selects elements that are siblings of another element.

Example: h2 ~ p { color: gray; }

Pseudo-classes: Selects elements based on a specific state or condition.

Example: a:hover { color: red; }

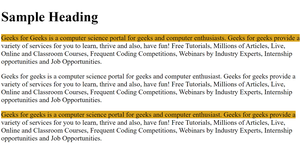
**Pseudo-elements selector** : Selects specific parts of an element.

It Style the first letter, or line, of an element

Ex-p::first-line{

background-color: yellow;

font-size: large; }



7) **What are Pseudo elements and Pseudo classes?**

Pseudo-elements and pseudo-classes are special selectors in CSS that allow you to target specific parts of elements or elements in specific states or conditions.

Pseudo-elements: Pseudo-elements select and style specific parts of an element, creating virtual elements within the actual element. They are denoted by a double colon :: notation.

Some common example of pseudo element are –

::before: Inserts content before the selected element.

::after: Inserts content after the selected element.

::first-line: Selects and styles the first line of text within the selected element.

::first-letter: Selects and styles the first letter of the text within the selected element.

::selection: Selects and styles the portion of text selected by the user.

**Pseudo-classes**: Pseudo-classes select elements based on specific states or conditions, such as user interaction or element hierarchy. They are denoted by a single colon : notation. Some commonly used pseudo-classes are:

:hover: Selects an element when the user hovers over it.

:active: Selects an element when it is being activated or clicked by the user.

:focus: Selects an element when it has keyboard focus.

:first-child: Selects the first child element of its parent.

:nth-child(): Selects elements based on their position within their parent.

:not(): Selects elements that do not match the specified selector.

**8) What property is used for changing the font face?**

We can use the font-family property for achieving this. The **font-family** property is used for specifying what font needs to be applied on the targetted DOM element.

p {

font-family: "Times New Roman", Times, serif;

}

The opacity property sets the opacity level for an element. The opacity-level describes the transparency-level of element .

In CSS3, opacity is specified using the opacity property. The opacity property accepts a value between 0 and 1, where 0 represents complete transparency (the element is invisible) and 1 represents complete opacity (the element is fully visible).

Example :

div {

opacity: 0.6;

}



9) **What are the properties of flexbox?**

flexbox is a CSS layout module that provides a flexible way to arrange and align elements within a container and it can be shift at anywhere on screen. It allows you to create responsive and dynamic layouts by distributing space among the flex items.

Flexbox stands for flexible box and it was introduced around 2017 in CSS

The properties of Flexbox (Flexible Box Layout) in CSS are –

**flex-direction**: The flex-direction property defines in which direction the container wants to stack the flex items.

Some flex direction value :

row: by this property Stacks items horizontally from left to right in the flex container.

column: Stacks items vertically from top to bottom in the flex container.

row-reverse: Stacks items horizontally from right to left in the flex container.

column-reverse: Stacks items vertically from bottom to top in the flex container.

Note: check w3 school flexbox property for figure !

**justify-content :** justify-content property is used to align the flex items

some justify-content value are –

center: The center value aligns the flex items at the center of the container:

flex-start: it means all the items are aligned at the start of the container. This is the default value of this property

flex-end: This value ensures all the items are aligned or shifted at the end of the container.

space-around:: The space-around value displays the flex items with space before,

and after the lines: This value displays the items with equal space between, before, and end of the item. around the items.

space-between: The space-between value placed the flex item at an extreme end position and have equal space between them.

Space evenly: the space evenly value display the flex item with equal space at Start, in-between, and end of the container.

**flex-wrap**: The flex-wrap property specifies whether the flex items should wrap or not. Let’s suppose we have nine div element. So when you small your website screen, then those div wch is not adjust in horizontally wch comes vertically , so benefit of flex wrap is item of div will not hide in this case.

Some example are -

wrap: The flex items would be wrapped if needed.

nowrap: This is the default value that says the items won’t be wrapped.

wrap-reverse: This specifies that the items will be wrapped if needed but in reverse order.

**align-items** : align-items property is used to align the flex items in vertical axis .

align-items: center - The center value aligns the flex items in the middle of the container in y or vertical axis:

align-items: flex-start - The flex-start value aligns the flex items at the top of the container:

align-items: flex-end - The flex-end value aligns the flex items at the bottom of the container:

align-items: stretch - The stretch value stretches the flex items to fill the container, it take full space of container (this is default):

align-items: baseline - The "align-items: baseline;" aligns the flex items such as their baselines aligns .

### 10) Can you name the four types of @media properties?

The four types of @media properties are:

1. All → It’s the default property. Used for all media-type devices.
2. Screen → Used for computer screen, mobile screen.
3. Print → Used for printers.
4. Speech → Used for screen readers.

**11) What are the different ways to hide the element using CSS?**

* Using display property(**display: none**). the display: none; value is used to hide an element from the web page, effectively removing it from the layout and and element will be invisible in that case. The element will not exist in the DOM and the space it would occupy is collapsed. It’s not available for screen readers.
* Using (**visibility: hidden**), property will value is also hide an element from the web page. The element will actually be present in the DOM, but not shown on the screen but it takes up the space of the element. It will be available to screen reader users.

**12) Different Box Sizing Property?**

The box-sizing property defines how the width and height of an element are calculated: should they include padding and borders, or not.

**content-box** : This is the default value for box-sizing.

With content-box property the width and height of an element only include the content area and do not consider the padding or border.

Ex-

.element {

box-sizing: content-box;

}

**border-box:** With border-box, the width and height of an element include the content area, padding, and border. The specified width and height values are applied to the entire box, including the content, padding, and border.

Ex **-**

**.element {**

**box-sizing: border-box;**

**}**

**13) Explain CSS position property?**

The CSS position property is used to specify how an element should be positioned on a webpage. It controls the positioning of elements

It allows you to specify how an element should be positioned relative to its normal position by use of position property.

**Static**: Elements with position: static value are positioned according to the normal flow of the document. It is the default value. and the element positioned are not affected by the top, right, bottom, or left properties value. It's not positioned in any way.

ex- CSS static position.html file check

**Relative**: Elements with position: relative value will be positioned relative to its original position . You can use the top, right, bottom, and left properties to adjust the element from its original position.

Ex- CS Relative.html

**Absolute:** Elements with position: absolute are positioned relative to their nearest positioned ancestor. Its shifts inside of nearest relative position div container . If there is no positioned ancestor, they are positioned relative to the initial containing block (usually the viewport or browser window). The top, right, bottom, and left properties are used to set the position of the element in relative position div contain.

Ex- Css Absolute position.html

**Fixed**: In this, the element is positioned relative to the browser window or viewport. It is fixed or stay in that position even though the window is been scrolled. You can use the top, right, bottom, and left properties to set the final fixed position of the element.

Ex- Css Fixed position.html

**Sticky**: Elements with position: sticky are positioned based on the user's scroll position. They behave like relative elements until they reach a specified threshold and then they become fixed in Top place of web page like sticky. The top, right, bottom, and left properties are used to set or adjust the sticky position.

Ex- Css Sticky position.html

**14) What is the grid system?**

CSS Grid Layout is a two-dimensional grid-based layout system, with rows and columns, making it easier to design web pages without having to use floats and positioning.

An HTML element becomes a grid container when its display property is set to grid or inline-grid. We can use grid template -row/column property to defined no of row and column and its size also in px, fr unit etc in grid also

The spaces between each column/row are called gaps in grid. You can increase or decreboxase this gap according to you.

**15)What are the different types of CSS frameworks are used by the developers?**

Tailwind CSS

BootStrap

Foundation

Bulma

UI Kit

Skeleton

Pure

.

**16)Explain gradient with example**

CSS Gradient basically represents a transition between two or more colors. It is a very important aspect to beautify and add them to your content**.**

There are 3 types of Gradient:

**Linear Gradient -** In linear Gradient, the color flows from left to right, top-to-bottom, or at any angle you chose in a single direction.

**Radial Gradient -** Radial Gradient starts at a single point and is defined from the center position.

**Conic Gradient -** In Conic Gradient transition is rotated around a center point. To create a conic gradient at least 2 colors need to be defined.

KEEP IN MIND NOTES

1) transform: translate(50px,100px);

Notes:

<p>The translate() method moves an element from its current position:</p>

This div element is moved 50 pixels to the right, and 100 pixels down from its current position.

**2) margin: 25px 50px;**

This div element has a top and bottom margin of 25px, and a right and left margin of 50px.

3)p {  
  margin: 25px 50px 75px 100px;  
}

**margin: 25px 50px 75px 100px;**

* + top margin is 25px
  + right margin is 50px
  + bottom margin is 75px
  + left margin is 100px

1. margin: 25px;

This div element has a top, bottom, left, and right margin of 25px.

1. The indexOf() method in JavaScript is used to find the index of the first occurrence of a specified value within a string or an array. It returns the index of the value if it is found, and -1 if the value is not present.