**Q1. What is the Box model in CSS? Which CSS properties are a part of it?**

A rectangle box is wrapped around every HTML element. The box model is used to determine the height and width of the rectangular box. The CSS Box consists of Width and height, padding, borders, margin.

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

**Q2. What are the advantages of using CSS?**

* **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 -** CSS, built effectively can be used to change the look and feel complete by making small changes. To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
* **Bandwidth -** Used effectively, the style sheets will be stored in the browser cache and they can be used on multiple pages, without having to download again.

**Q3. What are the limitations of CSS?**

* **Browser Compatibility:** Some style selectors are supported and some are not. We have to determine which style is supported or not using the @support selector.
* **Cross Browser issue:** Some selectors behave differently in a different browser.
* **There is no parent selector:** Currently, Using CSS, you can’t select a parent tag.

**Q4. How to include CSS in a web page?**

There are 3 different ways to include CSS in web page

1. **External Style Sheet**: An external file linked to your HTML document: Using link tag, we can link the style sheet to the HTML page.

<link rel="stylesheet" type="text/css" href="mystyles.css" />

1. **Internal Style Sheet**: A set of CSS styles included within your HTML page.

<style type="text/css">

/\*Add style rules here\*/

</style>

1. **Add inline styles** to HTML elements(CSS rules applied directly within an HTML tag.): Style can be added directly to the HTML element using a style tag.

<h2 style="color:red;background:black">Inline Style</h2>

### **Q5. What are the different types of Selectors in CSS?**

1. **Universal Selector**: The universal selector works like a wildcard character, selecting all elements on a page. In the given example, the provided styles will get applied to all the elements on the page.

\* {

color: "green";

}

1. **Element Type Selector**: This selector matches one or more HTML elements of the same name. In the given example, the provided styles will get applied to all the ul elements on the page.

ul {

border: solid 1px #ccc;

}

1. **ID Selector**: This selector matches any HTML element that has an ID attribute with the same value as that of the selector. In the given example, the provided styles will get applied to all the elements having ID as a container on the page.

#container {

width: 960px;

}

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

1. **Class Selector**: The class selector also matches all elements on the page that have their class attribute set to the same value as the class. In the given example, the provided styles will get applied to all the elements having ID as the box on the page.

.box {

padding: 10px;

margin: 10px;

width: 240px;

}

<div class="box"></div>

**Q6. What is new in CSS3?**

* CSS3 Selectors - ^, $ and \* are introduced. These selectors allow you to select elements having a value that starts with, ends with or has anywhere in it the value provided, respectively.
* Pseudo-classes - CSS3 introduces new pseudo-classes.
* CSS3 allows us to define colors in new ways. With older versions of CSS, we used to declare colours using the hexadecimal format (#FFF, or #FFFFFF for white). It was also possible to declare colours using the rgb() notation, providing either integers (0–255) or percentages. Now, there are 147 new colors “names” added to the list of keywords for colors and along with rgb, other options are added that allow us to define semi-transparent colors.
* The new border-radius property lets you create rounded corners without the need for images or additional markup.
* CSS3 provides the ability to add drop shadows to elements using the box-shadow property. CSS also allows us to add shadows to the text. This can be done using text-shadow CSS property. It adds shadows to individual characters in text.
* W3C added the syntax for generating linear gradients with CSS3.
* In CSS3, we can have multiple background images. There’s no need to include an element for every background image; it provides us with the ability to add **multiple background images** to any HTML element.

**Q7. How does absolute positioning work in CSS?**

In absolute positioning, the element is removed from the normal document flow, and no space is created for the element in the page layout. It is positioned relative to its closest positioned ancestor if any; otherwise, it is placed relative to the initial containing block. Its final position is determined by the values of top, right, bottom, and left.

**Q8. How does fixed positioning work in CSS?**

Fixed positioning is similar to absolute positioning, with the exception that the element's containing block is the initial containing block established by the viewport, unless any ancestor has transform, perspective, or filter property set to something other than none, which then causes that ancestor to take the place of the elements containing block. This can be used to create a "floating" element that stays in the same position regardless of scrolling.

**Q9. How to define a pseudo class in CSS?**

We can define a pseudo class by listing the selector followed by a colon and finally the pseudo class element. Following is an example.

div:hover {

color: red;

}

Pseudo classes can be used to give elements special states—the most common example being a:hover, which is used to change the color of a link when a mouse hovers over it. Other uses include using distinct styling for visited and unvisited links and styling an element differently when focused.

**Q10. What is responsive web design?**

* Responsive Web design is the approach that suggests that design and development should respond to the user’s behavior and environment based on screen size, platform and orientation.
* The practice consists of a mix of flexible grids and layouts, images and an intelligent use of CSS media queries. As the user switches from their laptop to phone, the website should automatically switch to accommodate for resolution, image size and scripting abilities. In other words, the website should have the technology to automatically respond to the user’s preferences. This would eliminate the need for a different design and development phase for each new gadget on the market.

**Q11. What is the exclamation(!) symbol used in CSS?**

* The exclamation symbol (!) is generally used to denote a style as important. The styles that are marked as !important override the styles that are not. For example, if there are two CSS classes applied on the same element. Class A adds a color red to the text but class B adds a green color. Now, generally, the style that is applied later will persist. But if either of the classes has the !important declaration will never be overridden by other styles.

Eg p {

color: red !important;

}

**Q12. How to style HTML elements through attributes?**

* There are a few attributes that apply to all tags, like class and id, but a lot of tags have their own ones. For example, input tags can have a type (text, number, radio, etc) and a tag can have href.
* You can target elements with particular attributes by using square brackets: [attribute="value"]. For example, to target all input fields that are of type radio like so:

input[type="radio"] {

background-color: #eee;

}

**Q13. What is the use of CSS Opacity?**

The CSS opacity property is used to specify the transparency of an element. In simple words, you can say that it specifies the clarity of the image. In technical terms, Opacity is defined as the degree to which light is allowed to travel through an object. For example:

<style>

p {

opacity: 0.4;

filter: alpha(opacity=40); /\* For IE8 and earlier \*/

}

</style>

**Q14. What is Descendant combinator and Child combinator?**

**Descendant combinator**

#container .box {

float: left;

padding-bottom: 15px;

}

This declaration block will apply to all elements that have a class of box that is inside an element with an ID of the container. It’s worth noting that the .box element doesn’t have to be an immediate child: there could be another element wrapping .box, and the styles would still apply.

**Child combinator**

A selector that uses the child combinator is similar to a selector that uses a descendant combinator, except it only targets immediate child elements.

#container> .box {

float: left;

padding-bottom: 15px;

}

The selector will match all elements that have a class of box and that are immediate children of the #container element. That means, unlike the descendant combinator, there can’t be another element wrapping .box it has to be a direct child element.

**Q15. How to control image position in the background?**

The background-position property is used to define the initial position of the background image. By default, the background image is placed on the top-left of the webpage.

We can set the following positions:

1. center
2. top
3. bottom
4. left
5. right

**Q16.** **What is VH/VW (viewport height/ viewport width) in CSS?**

* It’s a CSS unit used to measure the height and width in percentage with respect to the viewport. It is used mainly in responsive design techniques. 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 is 1000px, then 1 vw is equal to 10px.

**Q17. Enlist the various fonts’ attributes?**

* Font-style
* Font-variant
* Font-weight
* Font-size/line-height
* Font-family
* Caption
* Icon

**Q18. How does Z index function?**

Overlapping may occur while using CSS for positioning HTML elements. Z index helps in specifying the overlapping element. It is a number which can be positive or negative, the default value being zero.

**Q19. What is the difference between inline, inline-block, and block?**

1. **Block Element**: The block elements always start on a new line. They will also take space for an entire row or width. List of block elements are <div>, <p>.
2. **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.
3. **Inline Block Elements**: Inline-block elements are similar to inline elements, except they can have padding and margins added on all four sides.

**Q20. What is a Pseudo-class?**

Pseudo-classes select regular elements but under certain conditions like when the user is hovering over the link.

* :link
* :visited
* :hover
* :active
* :focus

Example of the pseudo-class, In the below example, the color applies to the anchor tag when it’s hovered.

/\* mouse over link \*/

a:hover {

color: #FFOOFF;

}

**Q21. What is a Pseudo element?**

A pseudo-element however allows us to create items that do not normally exist in the document tree, for example ::after.

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

In the below example, the color will appear only on the first line of the paragraph.

p: :first-line {

color: #ffOOOO;

font-variant: small-caps;

}

### **Q22. How to specify units in the CSS? What are the different ways to do it?**

There are different ways to specify units in CSS like px, em, pt, percentage (%).

* px(Pixel) gives fine-grained control and maintains alignment because 1 px or multiple of 1 px is guaranteed to look sharp. px is not cascade.
* em maintains relative size. you can have responsive fonts. Em, will cascade 1em is equal to the current font-size of the element or the browser default. If u sent font-size to 16px then 1em = 16px.
* pt(point) are traditionally used in print. 1pt = 1/72 inch and it is a fixed-size unit.
* %(percentage) sets font-size relative to the font size of the body. Hence, you have to set the font-size of the body to a reasonable size.

**Q23. Explain CSS position property?**

* **Absolute**

To place an element exactly where you want to place it. absolute position is actually set relative to the element's parent. if no parent is available then the relative place to the page itself (it will default all the way back up to the element).

* **Relative**

"Relative to itself". Setting position: relative; on an element and no other positioning attributes, it will have no effect on its positioning. It allows the use of z-index on the element and it limits the scope of absolutely positioned child elements. Any child element will be absolutely positioned within that block.

* **Fixed**

The element is positioned relative to the viewport or the browser window itself. viewport doesn't change if you scroll and hence the fixed element will stay right in the same position.

* **Static**

Static default for every single page element. The only reason you would ever set an element to position: static is to forcefully-remove some positioning that got applied to an element outside of your control.

* **Sticky**

Sticky positioning is a hybrid of relative and fixed positioning. The element is treated as relative positioned until it crosses a specified threshold, at which point it is treated as fixed positioned.

### **Q24. Different Box Sizing Property?**

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

* **Content-box:** The default width and height values apply to the element's content only. The padding and border are added to the outside of the 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:** Width and height values apply to the content, padding, and border.

**Q25. What are the different ways to hide the element using CSS?**

* **Using display property(display: none)**. It’s not available for screen readers. The element will not exist in the DOM if display: none is used.
* **Using visibility property(visibility: hidden)**, will take up the space of the element. It will be available to screen reader users. The element will actually be present in the DOM, but not shown on the screen.
* **Using position property (position: absolute)**. Make it available outside the screen.

**Q26. What is Flexbox in CSS?**

* The flexbox layout officially called CSS flexible box layout module is a new layout module in CSS3. It is made to improve the items alignment, directions and order in the container even when they are dynamic, or even unknown size. The prime characteristic of the flex container is the ability to modify the width or height of its children to fill the available space in the best possible way on different screen sizes.
* Many designers and developers find this flexbox layout easier to use, as the positioning of the elements is simpler thus more complex layouts can be achieved with less code, leading to a simpler development process. Flexbox layout algorithm is direction based unlike the block or inline layout which are vertically and horizontally based. This flexbox layout should be used for small application components, while the new CSS Grid Layout Module is emerging to handle the large scale layouts.

**Q27. Difference between CSS grid vs flexbox?**

* CSS Grid Layout is a two-dimensional system, meaning it can handle both columns and rows. Grid layout is intended for larger-scale layouts which aren’t linear in design.
* Flexbox is largely a one-dimensional system (either in a column or a row). Flexbox layout is most appropriate to the components of an application.

**Q28. How does Calc work?**

The CSS3 calc() function allows us to perform mathematical operations on property values. Instead of declaring, for example, static pixel values for an element's width, we can use calc() to specify that the width is the result of the addition of two or more numeric values.

.sample {

**width**: calc(100px + 50px) //or **width:** calc( 50% - 16px);

}

**Q29. What is {box-sizing: border-box;} ?**

* By default, elements have box-sizing: content-box applied, and only the content size is being accounted for.
* box-sizing: border-box changes how the width and height of elements are being calculated, border and padding are also being included in the calculation.
* The height of an element is now calculated by the content’s height + vertical padding + vertical border width.
* The width of an element is now calculated by the content’s width + horizontal padding + horizontal border width.
* Taking into account paddings and borders as part of our box model resonates better with how designers actually imagine content in grids.

**Q30. What is the difference between padding and margin?**

In CSS, the margin is the property by which we can create space around elements. We can even create space to the exterior defined borders.

In CSS, we have margin property as follows:

* margin-top
* margin-right
* margin-bottom
* Margin-left

In CSS, padding is the property by which we can generate space around an element’s content as well as inside any known border.

**CSS padding also has properties like,**

1. Padding-top
2. Padding-right
3. Padding-bottom
4. Padding-left