

CSS Interview Contents

CSS Interview Questions For Freshers

1. What is the Box model in CSS? Which CSS properties are a part of it?
2. What are the advantages of using CSS?
3. What are the limitations of CSS?
4. How to include CSS in the webpage?
5. What are the different types of Selectors in CSS?
6. What is a CSS Preprocessor? What are Sass, Less, and Stylus? Why do people use them?
7. What is VH/VW (viewport height/ viewport width) in CSS?
8. Difference between reset vs normalize CSS?. How do they differ?
9. What is the difference between inline, inline-block, and block?
10. How do you test the webpage in different browsers?
11. What is a Pseudo element? What is pseudo-class?
12. How do you specify units in the CSS?. What are the different ways to do it?
13. Does margin-top or margin-bottom have an effect on inline elements?

Advanced CSS Interview Questions

14. Explain CSS position property?
15. What does DOM reflow occur?
16. Different Box Sizing Property?
17. How to center align a div inside another div?
18. Can you name the four types of @media properties

Advanced CSS Interview Questions

19. What is the grid system?
20. What are the different ways to hide the element using CSS?
21. What does the :root pseudo-class refer to?
22. What does Accessibility (a11y) mean?
23. How do I restore the default value of a property?
24. Difference between CSS grid vs flexbox?
25. How does Calc work?
26. What do CSS Custom properties variables mean?
27. What is the difference between CSS variables and preprocessor(SASS, LESS, Stylus) variables?
28. What does * { box-sizing: border-box; } do? What are its advantages?
29. What does important mean in CSS?
30. What is specificity? How to calculate specificity?
31. What is progressive rendering? How do you implement progressive rendering in the website?. What are the advantages of it?
32. What are the advantages of using translate() instead of absolute position?
33. Does style1.css have to be downloaded and parsed before style2.css can be fetched?
34. How to determine if the browser supports a certain feature?

Let's get Started

What Is CSS?

CSS stands for Cascading Style Sheet. It's a style sheet language that determines how the elements/contents in the page are looked/shown. CSS is used to develop a consistent look and feel for all the pages.

CSS was developed and is maintained by the World Wide Web Consortium (W3C). It was first released on December 17, 1996. The CSS Working group currently working with different browser vendors to add/enforce the new feature/ specifications in all the browsers.

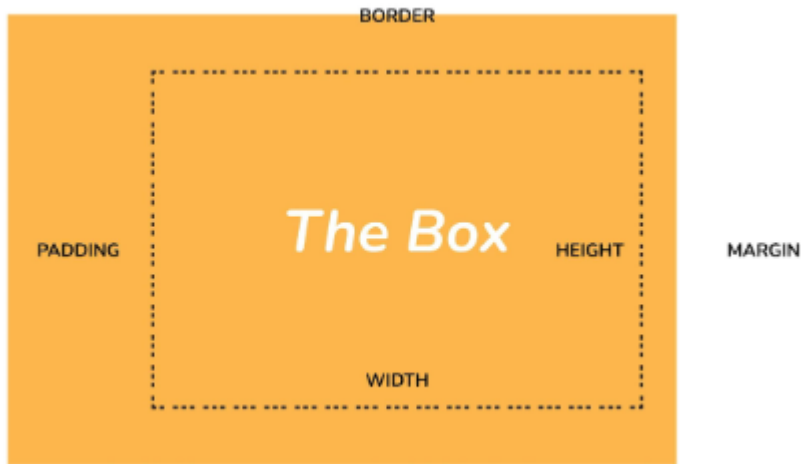
CSS enables the separation of the content from the presentation. This separation provides a lot of flexibility and control over how the website has to look like. This is the main advantage of using CSS.

This article will walk you through the question that you could expect in a CSS interview. The questions range from basic, intermediate, to advanced questions.

CSS Interview Questions For Freshers

1. 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 (or in the absence of that, default values and the content inside), padding, borders, margin.



Box Model In CSS

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

2. 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** - CSS, built effectively can be used to change the look and feel completely 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.

3. What are the limitations of CSS?

Disadvantages of CSS are given below:

- **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.

4. How to include CSS in the webpage?

There are different ways to include a CSS in a webpage,

- 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" />
```

- 2 - Embed CSS with a style tag: A set of CSS styles included within your HTML page.

```
<style type="text/css">
/*Add style rules here*/
</style>
```

Add your CSS rules between the opening and closing style tags and write your CSS exactly the same way as you do in stand-alone stylesheet files.

- 3 - 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>
```

- 4 - Import a stylesheet file (An external file imported into another CSS file): Another way to add CSS is by using the @import rule. This is to add a new CSS file within CSS itself.

```
@import "path/to/style.css";
```

5. What are the different types of Selectors in CSS?

A CSS selector is the part of a CSS ruleset that actually selects the content you want to style. Different types of selectors are listed below.

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";  
  font-size: 20px;  
  line-height:  
  25px;  
}
```

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 {  
  
#container {  
  width: 960px;  
  margin: 0  
  auto;  
}  
  
<div id="container"></div>
```

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.

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>
```

Descendant Combinator: The descendant selector or, more accurately, the descendant combinator lets you combine two or more selectors so you can be more specific in your selection method.

```
#container .box {  
  float: left;  
  padding-bottom: 15px;  
}  
  
<div id="container">  
  <div class="box"></div>  
  
  <div class="box-2"></div>
```

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;
}

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

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

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.

General Sibling Combinator: A selector that uses a general sibling combinator to match elements based on sibling relationships. The selected elements are beside each other in the HTML.

```
h2 ~ p {
  margin-bottom: 20px;
}

<h2>Title</h2>
<p>Paragraph example.</p>
<p>Paragraph example.</p>
<p>Paragraph example.</p>
<div class="box">
  <p>Paragraph example.</p>
</div>
```

In this example, all paragraph elements (`<p>`) will be styled with the specified rules, but only if they are siblings of `<h2>` elements. There could be other elements in between the `<h2>` and `<p>` , and the styles would still apply.

Adjacent Sibling Combinator: A selector that uses the adjacent sibling combinator uses the plus symbol (+), and is almost the same as the general sibling selector. The difference is that the targeted element must be an

```
p + p {  
  text-indent: 1.5em;  
  margin-bottom: 0;  
}  
  
<h2>Title</h2>  
<p>Paragraph example.</p>  
<p>Paragraph example.</p>  
<p>Paragraph example.</p>  
  
<div class="box">  
  <p>Paragraph example.</p>  
  <p>Paragraph example.</p>  
</div>
```

immediate sibling, not just a general sibling.

The above example will apply the specified styles only to paragraph elements that immediately follow other paragraph elements. This means the first paragraph element on a page would not receive these styles. Also, if another element appeared between two paragraphs, the second paragraph of the two wouldn't have the styles applied.

Attribute Selector: The attribute selector targets elements based on the presence and/or value of HTML attributes, and is declared using square brackets.

```
input [type="text"] {  
  background-color:  
  #444; width: 200px;  
}  
  
<input type="text">
```


6. What is a CSS Preprocessor? What are Sass, Less, and Stylus? Why do people use them?

A CSS Preprocessor is a tool used to extend the basic functionality of default vanilla CSS through its own scripting language. It helps us to use complex logical syntax like – variables, functions, mixins, code nesting, and inheritance to name a few, supercharging your vanilla CSS.

SASS: Sass is the acronym for “Syntactically Awesome Style Sheets”. SASS can be written in two different syntaxes using SASS or SCSS

SASS vs SCSS

- SASS is based on indentation and SCSS(Sassy CSS) is not.
- SASS uses .sass extension while SCSS uses .scss extension.
- SASS doesn't use curly brackets or semicolons. SCSS uses it, just like the CSS.

SASS Syntax

```
$font-color: #fff
$bg-color: #00f

#box
  color: $font-color
  background: $bg-color
```

SCSS Syntax

```
$font-color: #fff;
$bg-color: #00f;

#box{
  color: $font-color;
  background: $bg-color;
}
```

LESS: LESS is an acronym for “Leaner Stylesheets”. LESS is easy to add to any javascript projects by using NPM or less.js file. It uses the extension .less.

LESS syntax is the same as the SCSS with some exceptions. LESS uses @ to define the variables.

```
@font-color: #fff;
@bg-color: #00f

#box{
  color: @font-color;
  background: @bg-color;
}
```

Stylus: Stylus offers a great deal of flexibility in writing syntax, supports native CSS as well as allows omission of brackets, colons, and semicolons. It doesn't use @ or \$ for defining variables.

```
/* STYLUS SYNTAX WRITTEN LIKE NATIVE CSS */
font-color=
#fff; bg-color =
#00f;

#box {
  color: font-color;
  background: bg-color;
}

/* OR */

/* STYLUS SYNTAX WITHOUT CURLY BRACES */
font-color=
#fff; bg-color =
#00f;

#box
  color: font-color;
  background: bg-color;
```

7. 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.

8. Difference between reset vs normalize CSS?. How do they differ?

Reset CSS: CSS resets aim to remove all built-in browser styling. For example margins, paddings, font-sizes of all elements are reset to be the same.

Normalize CSS: Normalize CSS aims to make built-in browser styling consistent across browsers. It also corrects bugs for common browser dependencies.

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

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>`.

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>`, ``, ``, and `` tags.

Inline Block Elements: Inline-block elements are similar to inline elements, except they can have padding and margins added on all four sides.

10. How do you test the webpage in different browsers?

It's most important to test a website in different browsers when you're first designing it, or when making major changes. However, it's also important to repeat these tests periodically, since browsers go through a lot of updates and changes.

11. What is a Pseudo element? What is 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: #FF00FF;  
}
```

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: #ff0000;  
    font-variant: small-caps;  
}
```

12. How do you 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. The common practice is to set default body font-size to 62.5% (equal to 10px).

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.

13. Does margin-top or margin-bottom have an effect on inline elements?

No, it doesn't affect the inline elements. Inline elements flow with the contents of the page.

Advanced CSS Interview Questions

14. 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 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

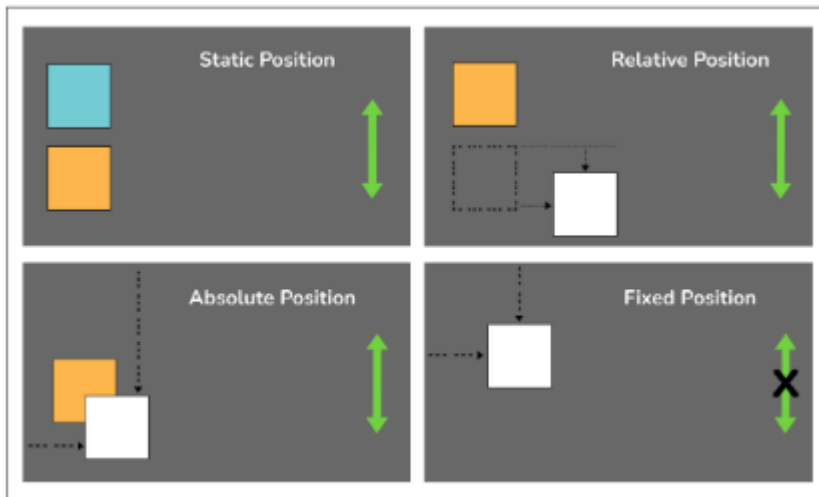
The element is positioned relative to the viewport or the browser window itself. viewport doesn't change if you scroll and hence the fixed element will stay right in the same position.

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.



CSS Position Property

15. What does DOM reflow occur?

Reflow is the name of the web browser process for re-calculating the positions and geometries of elements in the document, for the purpose of re-rendering part or all of the document.

Reflow occurs when:

- Insert, remove or update an element in the DOM.
- Modify content on the page, e.g. the text in an input box.
- Move a DOM element.
- Animate a DOM element.
- Take measurements of an element such as `offsetHeight` or `getComputedStyle`.
- Change a CSS style.

16. 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.

17. How to center align a div inside another div?

Centering with table

HTML:

```
<div class="cn"><div class="inner">your content</div></div>
```

CSS:

```
.cn {  
  display:  
  table-cell; width:  
  500px; height:  
  500px;  
  vertical-align: middle;  
  text-align: center;  
}  
  
.inner {  
  display: inline-block;  
  width: 200px; height: 200px;  
}
```

Centering with

transform HTML:

```
<div class="cn"><div class="inner">your content</div></div>
```

CSS:

```
.cn {
  position: relative;
  width: 500px;
  height: 500px;
}

.inner {
  position: absolute;
  top: 50%; left: 50%;
  transform: translate(-50%, -50%);
  width: 200px;
  height: 200px;
}
```

Centering with flexbox

HTML:

```
<div class="cn"><div class="inner">your content</div></div>
```

CSS:

```
.cn {
  display: flex;
  justify-content: center;
  align-items: center;
}
```

Centering with grid

HTML:

```
<div class="wrap_grid">
  <div id="container">vertical aligned text<br />some more text here
</div>
</div>
```

CSS:

```
.wrap-grid {
  display: grid;
  place-content: center;
}
```

18. Can you name the four types of @media properties?

The four types of @media properties are:

- All → It's the default property. Used for all media-type

devices. • Screen → Used for computer screen, mobile screen.

• Print → Used for printers.

• Speech → Used for screen readers.

19. What is the grid system?

CSS Grid Layout is the most powerful layout system available in CSS. It is said to be a 2-dimensional system, meaning it can handle both columns and rows, unlike flexbox which is largely a 1-dimensional system.

20. 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.

21. What does the :root pseudo-class refer to?

The `:root` selector allows you to target the highest-level "parent" element in the DOM, or document tree. It is defined in the CSS Selectors Level 3 specification.

22. What does Accessibility (a11y) mean?

Accessibility refers to how software or hardware combinations are designed to make a system accessible to persons with disabilities, such as visual impairment, hearing

For example, a website developed with accessibility in mind might have text-to-speech capabilities. In the USA public websites have to have accessible compliance. It's defined in 508 compliance. It gives the guidelines and best practices for all website users that should be met with key areas of accessibility.

23. How do I restore the default value of a property?

The keyword initial can be used to reset it to its default value.

24. 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.

25. 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.

```
.foo {  
  width: calc(100px + 50px)  
}
```

26. What do CSS Custom properties variables mean?

Custom properties (sometimes referred to as CSS variables or cascading variables) are defined by users that contain specific values to be reused throughout a document. The value is set using -- notion. And the values are accessed using the var() function.

```

:root {
  --main-bg-color: brown
}

.one {
  color: white;
  background-color: var(--main-bg-color);
  margin:
  10px, width:
  50px,
  height:
  50px;
  display: inline-block;
}

```

27. What is the difference between CSS variables and preprocessor(SASS, LESS, Stylus) variables?

CSS variables can be used without the need of the preprocessor. Currently, all the major browsers support the CSS variables.

CSS variable cascade. But the preprocessor variables don't cascade. CSS variable can be accessed and manipulated javascript.

28. What does * { box-sizing: border-box; } do? What are its advantages?

It makes every element in the document include the padding and border in the element's inner dimension for the height and width computation. In box-sizing: border-box, 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.

29. What does important mean in CSS?

The style is having the important will have the highest precedence and it overrides the cascaded property.

```
p {  
  color: red !important;  
}  
#thing {  
  color: green;  
}  
  
<p id="thing">Will be RED.</p>
```

30. What is specificity? How to calculate specificity?

A process of determining which CSS rule will be applied to an element. It actually determines which rules will take precedence. Inline style usually wins then ID then the class value (or pseudo-class or attribute selector), the universal selector (*) has no specificity. ID selectors have a higher specificity than attribute selectors.

31. What is progressive rendering? How do you implement progressive rendering in the website?. What are the advantages of it?

Progressive rendering is the name given to techniques used to improve the performance of a webpage (in particular, improve perceived load time) to render content for display as quickly as possible.

We can implement the progressive rendering of the page by loading the lazy loading of the images. We can use Intersection Observer API to lazy load the image. The API makes it simple to detect when an element enters the viewport and take an action when it does. Once the image enters the viewport, we will start loading the images.

A sample snippet is given below.

```



document.addEventListener("DOMContentLoaded", function() {
    var lazyImages = [].slice.call(document.querySelectorAll("img.lazy"));

    if ("IntersectionObserver" in window) {
        let lazyImageObserver = new IntersectionObserver(function(entries, observer) {
            entries.forEach(function(entry) {
                if (entry.isIntersecting) {
                    let lazyImage =
                        entry.target;
                    lazyImage.src = lazyImage.dataset.src;
                    lazyImage.srcset = lazyImage.dataset.srcset;
                    lazyImage.classList.remove("lazy");
                    lazyImageObserver.unobserve(lazyImage);
                }
            });
        });

        lazyImages.forEach(function(lazyImage) {
            lazyImageObserver.observe(lazyImage);
        });
    } else {
        // Possibly fall back to event handlers here
    }
});

```

32. What are the advantages of using translate() instead of absolute position?

Translate() does not cause the browser to trigger repaint and layout and instead only acts on the compositor. The absolute position triggers the repaint or DOM reflow. So, translate() gives the better performance.

33. Does style1.css have to be downloaded and parsed before style2.css can be fetched?

```

<head>
  <link h ref=" style1. css" rel=" stylesheet">
  <link href="style2.css" rel="stylesheet">
</head>

```

No, the browsers will download the CSS in the order of its appearance on the HTML page.

34. How to determine if the browser supports a certain feature?

The @support in CSS can be very useful to scan if the current browser has support for a certain feature.

```
@supports (display: grid) {  
  div {  
    display: grid;  
  }  
}
```