**1. What is CSS ?**

The full form of CSS is Cascading Style Sheets. It is a styling language which is simple enough for [HTML](https://career.guru99.com/top-50-html-interview-questions/) elements. It is popular in web designing, and its application is common in XHTML also.

**2. What is the origin of CSS ?**

Standard Generalized Markup Language marked the beginning of style sheets in 1980s.

**3. What are the different variations of CSS ?**  
The variations for CSS are:

* CSS 1
* CSS 2
* CSS 2.1
* CSS 3
* CSS 4

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

Limitations are:

* Ascending by selectors is not possible
* Limitations of vertical control
* No expressions
* No column declaration
* Pseudo-class not controlled by dynamic behavior
* Rules, styles, targeting specific text not possible

**5. What are the advantages of CSS ?**

Advantages are:

* Bandwidth
* Site-wide consistency
* Page reformatting
* Accessibility
* Content separated from presentation

[](https://career.guru99.com/wp-content/uploads/2013/10/CSS.jpg)CSS

**6. What are CSS frameworks?**

It is a pre-planned libraries, which allows easier and more standards-compliant webpage styling, using CSS language.

**7. How block elements can be centered with CSS1?**

Block level elements can be centered by:

The margin-left and margin-right properties can be set to some explicit value:



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | BODY {    width: 40em;    background: fluorescent;    }    P {    width: 30em;    margin-right: auto;    margin-left: auto    } |

In this case, the left and right margins will be each, five ems wide since they split up the ten ems left over from (40em-30em). It was unnecessary for setting up an explicit width for the BODY element; it was done here for simplicity.

**8. Who maintains the CSS specifications?**

World Wide Web Consortium maintains the CSS specifications.

**9. In how many ways can a CSS be integrated as a web page?**

CSS can be integrated in three ways:

* Inline: Style attribute can be used to have CSS applied HTML elements.
* Embedded: The Head element can have a Style element within which the code can be placed.
* Linked/ Imported: CSS can be placed in an external file and linked via link element.

**10. What benefits and demerits do External Style Sheets have?**  
Benefits:

* One file can be used to control multiple documents having different styles.
* Multiple HTML elements can have many documents, which can have classes.
* To group styles in composite situations, methods as selector and grouping are used.

Demerits:

* Extra download is needed to import documents having style information.
* To render the document, the external style sheet should be loaded.
* Not practical for small style definitions.

**11. Discuss the merits and demerits of Embedded Style Sheets?**  
Merits of Embedded Style Sheets:

* Multiple tag types can be created in a single document.
* Styles, in complex situations, can be applied by using Selector and Grouping methods.
* Extra download is unnecessary.

Demerits of Embedded Style Sheets:

* Multiple documents cannot be controlled.

**12. What does CSS selector mean?**

A string equivalent of HTML elements by which declarations or a set of it, is declared and is a link that can be referred for linking HTML and Style sheet is CSS selector.

**13. Enlist the media types CSS allows?**

The design and customization of documents are rendered by media. By applying media control over the external style sheets, they can be retrieved and used by loading it from the network.

**14. Differentiate logical tags from physical tags?**

* While physical tags are also referred to as presentational mark-up, logical tags are useless for appearances.
* Physical tags are newer versions while logical tags are old and concentrate on content.

**15. Differentiate Style Sheet concept from HTML?**

While HTML provides easy structure method, it lacks styling, unlike Style sheets. Moreover, style sheets have better browser capabilities and formatting options.

**16. Describe ‘ruleset’?**

Ruleset : Selectors can be attached to other selectors to be identified by ruleset.

It has two parts:

* Selector, e.g. R and
* declaration {text-indent: 11pt}

**17. Comment on the Case-sensitivity of CSS ?**

Although, there are no case-sensitivity of CSS, nevertheless font families, URL’s of images, etc is. Only when [XML](https://career.guru99.com/xml-interview-questions/) declarations along with XHTML DOCTYPE are being used on the page, CSS is case -sensitive.

**18. Define Declaration block?**

A catalog of directions within braces consisting of property, colon and value is called declaration block.  
e.g.: [property 1: value 3]

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

They are:

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

**20. Why is it easy to insert a file by importing it?**

Importing enables combining external sheets to be inserted in many sheets. Different files and sheets can be used to have different functions. Syntax:

@import notation, used with <Style> tag.

**21. What is the usage of Class selector?**

Selectors that are unique to a specific style, are called CLASS selectors. Declaration of style and association with HTML can be made through this. Syntax:

Classname  
it can be A-Z, a-z or digits.  
.top {font: 14em ;}, class selector  
<Body class= “top”> this class is associated with element </body>

**22. Differentiate Class selector from ID selector?**

While an overall block is given to class selector, ID selector prefers only a single element differing from other elements. In other words, ID are uniques while classes are not. Its possible that an element has both class and ID.

**23. Can more than one declaration be added in CSS?**

Yes, it can be achieved by using a semicolon.

**24. What is Pseudo-elements ?**

Pseudo-elements are used to add special effects to some selectors.  CSS in used to apply styles in HTML mark-up. In some cases when extra mark-up or styling is not possible for the document, then there is a feature available in CSS known as pseudo-elements. It will allow extra mark-up to the document without disturbing the actual document.

**25. How to overrule underlining Hyperlinks?**

Control statements and external style sheets are used to overrule underlining Hyperlinks.

E.g.:



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | B {    text-decoration: none;    }    <B href="career.html" style="text-decoration: none">link text</B> |

**26. What happens if 100% width is used along with floats all across the page?**

While making the float declaration, 1 pixel is added every time it is used in the form of the border, and   even more float is allowed thereafter.

**27. Can default property value be restored through CSS? If yes, how?**

In CSS, you cannot revert back to old values due to lack of default values. The property can be re- declared to get the default property.

**28. Enlist the various Media types used?**

Different media has different properties as they are case insensitive.

They are:

* Aural – for sound synthesizers and speech
* Print – gives a preview of the content when printed
* Projection- projects the CSS on projectors.
* Handheld- uses handheld devices.
* Screen- computers and laptop screens.

**29. What is CSS Box Model and what are its elements?**

This box defines design and layout of elements of CSS. The elements are:

**Margin**: the top most layer, the overall structure is shown  
**Border**: the padding and content option with a border around it is shown.  Background color affects the border.  
**Padding**: Space is shown. Background colour affects the border.  
**Content**: Actual content is shown.

**30. What is contextual selector?**

Selector used to select special occurrences of an element is called contextual selector. A space separates the individual selectors. Only the last element of the pattern is addressed in this kind of selector. For e.g.: TD P TEXT {color: blue}

**31. Compare RGB values with Hexadecimal color codes ?**

A color can be specified in two ways:

* A color is represented by 6 characters i.e. hexadecimal color coding. It is a combination of numbers and letters and is preceded by #.       e.g.: g {color: #00cjfi}
* A color is represented by a mixture of red, green and blue. The value of a color can also be specified. e.g.: rgb(r,g,b): In this type the values can be in between the integers 0 and 255. rgb(r%,g%,b%):  red, green and blue percentage is shown.

**32. Define Image sprites with context to CSS ?**

When a set of images is collaborated into one image, it is known as ‘Image Sprites’. As the loading every image on a webpage consumes time, using image sprites lessens the time taken and gives information quickly.

CSS coding:



|  |  |
| --- | --- |
| 1 | img.add { width: 60px; height: 55px; background: url (image.god) 0 0; } |

In this case, only the part needed is used. The user can save substantial margin and time through this.

**33. Compare Grouping and Nesting in CSS ?**

Grouping:  Selectors can be grouped having the same values of property and the code be reduced.  
E.g. :



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | h1 {    color: blue;    }    h2 {    color: blue;    }    p {    color: blue;    } |

It can be seen from the code that every element shares the same property. Rewriting can be avoided by writing each selector separated by a comma.

Nesting: Specifying a selector within a selector is called nesting.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | P    {    color: red;    text-align: left;    }    .marked    {    background-color: blue;    }    .marked p    {    color: green;    } |

**34. How can the dimension be defined of an element ?**

Dimension properties can be defined by:

* Height
* Max-height
* Max-width
* Min-height
* Min-width
* Width

**35. Define float property of CSS?**

By float property, the image can be moved to the right or the left along with the text to be wrapped around it. Elements before this property is applied do not change their properties.

**36. 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.

**37. What is graceful degradation?**

In case the component fails, it will continue to work properly in the presence of a graceful degradation. The latest browser application is used when a webpage is designed. As it is not available to everyone, there is a basic functionality, which enables its use to a wider audience. In case the image is unavailable for viewing, text is shown with the alt tag.

**38. What is progressive enhancement?**

It’s an alternative to graceful degradation, which concentrates on the matter of the web. The functionality is same, but it provides an extra edge to users having the latest bandwidth. It has been into prominent use recently with mobile internet connections expanding their base.

**39. How can backward compatibility be designed in CSS?**

HTML sheet methods is collaborated with CSS and used accordingly.

**40. How can the gap under the image be removed?**

As images being inline elements are treated same as texts, so there is a gap left, which can be   removed by:

CSS



|  |  |
| --- | --- |
| 1 | img { display: block ; } |

**41. Why is @import only at the top?**

@import is preferred only at the top, to avoid any overriding rules. Generally, ranking order is followed in most programming languages such as Java, Modula, etc. In C, the # is a prominent example of a @import being at the top.

**42. Which among the following is more precedent: CSS properties or HTML procedures?**

CSS is more precedent over HTML procedures. Browsers, which do not have CSS support, display HTML attributes.

**43. What is Inline style?**

The Inline style in a CSS is used to add up styling to individual HTML elements.

**44.  How comments can be added in CSS?**

The comments in CSS can be added with /\* and \*/.

**45. Define Attribute Selector ?**

It is defined by a set of elements, value and its parts.

**46. Define property?**

A style, that helps in influencing CSS. E.g. FONT. They have corresponding values or properties within them, like FONT has different style like bold, italic etc.

**47.  What is Alternate Style Sheet?**

Alternate Style Sheets allows the user to select the style in which the page is displayed using the view>page style menu. Through Alternate Style Sheet, user can see a multiple version of the page on their needs and preferences.

**48. Are quotes mandatory in URL’s?**

Quotes are optional in URL’s, and it can be single or double.

**49. What is at-rule?**

Rule, which is applicable in the entire sheet and not partly, is known as at-rule. It is preceded by @ followed by A-Z, a-z or 0-9.

**50. How can CSS be cascaded to mix with user’s personal sheet?**

Properties can be a set in recommended places and the document modified for CSS to mix with user’s   personal sheet.

**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.

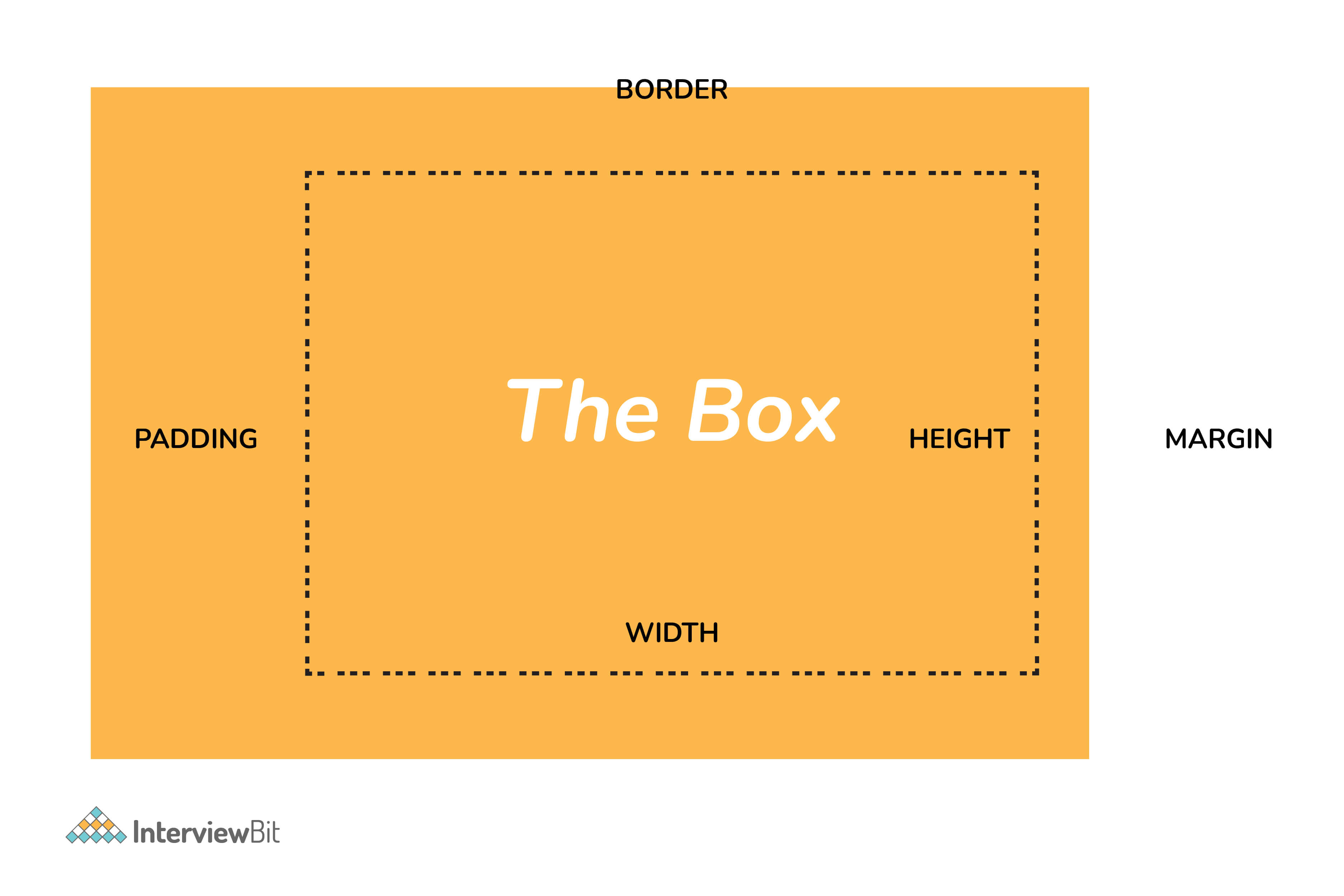
CSS enables the separation of the content from the presentation. This separation provides a lot of flexibility and control over how the website has to look like. This is the main advantage of using CSS.

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 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.

**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** {

line-style: none;

**border**: solid 1px #ccc;

}

**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;

**margin**: 0 auto;

}

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

**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**>

</**div**>

<**div** class=”box”></**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 immediate sibling, not just a general sibling.

**p** + **p** {

**text-indent**: 1.Sem;

**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**>

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>, <span> , <strong>, and <img> 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**: #FFOOFF;

}

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;

}

**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**

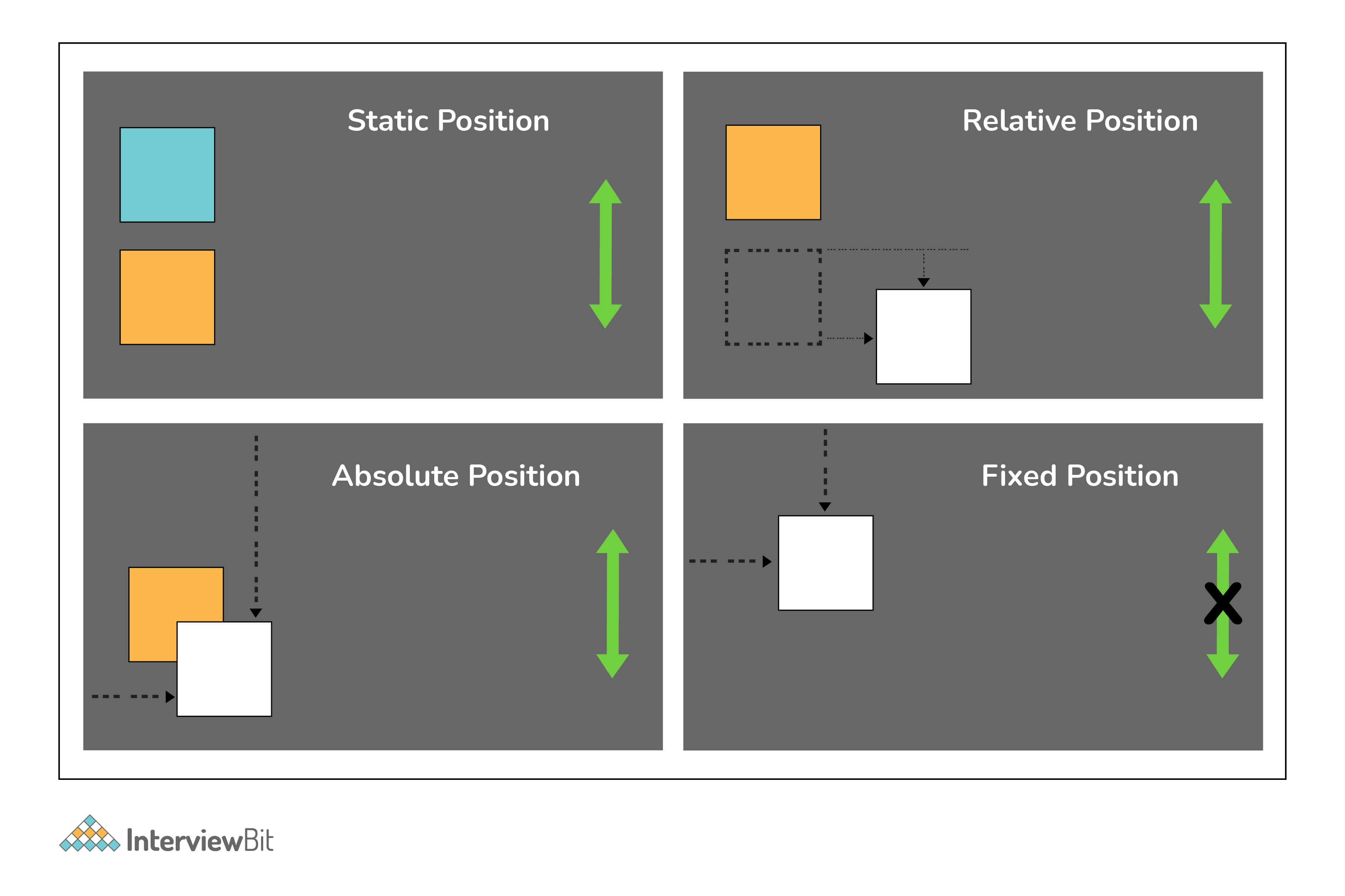
The element is positioned relative to the viewport or the browser window itself. viewport doesn't change if you scroll and hence the fixed element will stay right in the same position.

**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 loss, or limited dexterity.

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**: l0px,

width: 50px,

height: 5Opx;

**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.

<**img** class="lazy"

**src**="placeholder-image.jpg"

data-**src**="image-**to**-lazy-load-1x.jpg"

data-srcset="image-**to**-lazy-load-2x.jpg 2x, image-**to**-lazy-load-1x.jpg 1x"

alt="**I**'m an image!">

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=" stylel. 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;

}

}

1) What is CSS?

CSS stands for Cascading Style Sheet. It is a popular styling language which is used with HTML to design websites. It can also be used with any XML documents including plain XML, SVG, and XUL.[More details...](https://www.javatpoint.com/what-is-css)

2) What is the origin of CSS?

SGML (Standard Generalized Markup Language) is the origin of CSS. It is a language that defines markup languages.

3) What are the different variations of CSS?

Following are the different variations of CSS:

* CSS1
* CSS2
* CSS2.1
* CSS3
* CSS4

4) How can you integrate CSS on a web page?

There are three methods to integrate CSS on web pages.

1. Inline method - It is used to insert style sheets in HTML document
2. Embedded/Internal method - It is used to add a unique style to a single document
3. Linked/Imported/External method - It is used when you want to make changes on multiple pages.

[More details...](https://www.javatpoint.com/how-to-add-css)

5) What are the advantages of CSS?

* Bandwidth
* Site-wide consistency
* Page reformatting
* Accessibility
* Content separated from presentation

6) What are the limitations of CSS?

* Ascending by selectors is not possible
* Limitations of vertical control
* No expressions
* No column declaration
* Pseudo-class not controlled by dynamic behavior
* Rules, styles, targeting specific text not possible

7) What are the CSS frameworks?

CSS frameworks are the preplanned libraries which make easy and more standard compliant web page styling. The frequently used CSS frameworks are: -

* Bootstrap
* Foundation
* Semantic UI
* Gumby
* Ulkit

8) Why background and color are the separate properties if they should always be set together?

There are two reasons behind this:

* It enhances the legibility of style sheets. The background property is a complex property in CSS, and if it is combined with color, the complexity will further increase.
* Color is an inherited property while the background is not. So this can make confusion further.

9) What is Embedded Style Sheet?

An Embedded style sheet is a CSS style specification method used with HTML. You can embed the entire stylesheet in an HTML document by using the STYLE element. [More details...](https://www.javatpoint.com/internal-css)

1. **<style>**
2. body {
3. background-color: linen;
4. }
5. h1 {
6. color: red;
7. margin-left: 80px;
8. }
9. **</style>**

10) What are the advantages of Embedded Style Sheets?

* You can create classes for use on multiple tag types in the document.
* You can use selector and grouping methods to apply styles in complex situations.
* No extra download is required to import the information.

11) What is a CSS selector?

It is a string that identifies the elements to which a particular declaration apply. It is also referred as a link between the HTML document and the style sheet. It is equivalent of HTML elements. There are several different types of selectors in CSS: -

* CSS Element Selector
* CSS Id Selector
* CSS Class Selector
* CSS Universal Selector
* CSS Group Selector

[More details...](https://www.javatpoint.com/css-selector)

12) Name some CSS style components.

Some CSS Style components are:

* Selector
* Property
* Value

13) What is the use of CSS Opacity?

The CSS opacity property is used to specify the transparency of an element. In simple word, 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:

1. **<style>**
2. img.trans {
3. opacity: 0.4;
4. filter: alpha(opacity=40); /\* For IE8 and earlier \*/
5. }
6. **</style>**

14) Explain universal selector.

The universal selector matches the name of any of the element type instead of selecting elements of a specific type.

1. **<style>**
2. \* {
3. color: green;
4. font-size: 20px;
5. }
6. **</style>**

15) Which command is used for the selection of all the elements of a paragraph?

The p[lang] command is used for selecting all the elements of a paragraph.

16) What is the use of % unit?

It is used for defining percentage values.

17) Name the property used to specify the background color of an element.

The background-color property is used to specify the background color of the element. For example:

1. **<style>**
2. h2,p{
3. background-color: #b0d4de;
4. }
5. **</style>**

18) Name the property for controlling the image repetition of the background.

The background-repeat property repeats the background image horizontally and vertically. Some images are repeated only horizontally or vertically.

1. **<style>**
2. body {
3. background-image: url("paper1.gif");
4. margin-left:100px;
5. }
6. **</style>**

19) Name the property for controlling the 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.

You can set the following positions:

1. center
2. top
3. bottom
4. left
5. right
6. background: white url('good-morning.jpg');
7. background-repeat: no-repeat;
8. background-attachment: fixed;
9. background-position: center;

20) Name the property for controlling the image scroll in the background.

The background-attachment property is used to specify if the background image is fixed or scroll with the rest of the page in the browser window. If you set fixed the background image, then the image not move during scrolling in the browser. Let's take an example with the fixed background image.

1. background: white url('bbb.gif');
2. background-repeat: no-repeat;
3. background-attachment: fixed;

21) What is the use of ruleset?

The ruleset is used to identify that selectors can be attached with other selectors. It has two parts:

* Selector - Selector indicates the HTML element you want to style.
* Declaration Block - The declaration block can contain one or more declarations separated by a semicolon.

22) What is the difference between class selectors and id selectors?

An overall block is given to class selector while id selectors take only a single element differing from other elements.

CSS Class Selector

1. **<style>**
2. .center {
3. text-align: center;
4. color: blue;
5. }
6. **</style>**

CSS Id Selector

1. **<style>**
2. #para1 {
3. text-align: center;
4. color: blue;
5. }
6. **</style>**

[More details...](https://www.javatpoint.com/css-selector)

23) What are the advantages of External Style Sheets?

* You can create classes for reusing it in many documents.
* By using it, you can control the styles of multiple documents from one file.
* In complex situations, you can use selectors and grouping methods to apply styles.

[More details...](https://www.javatpoint.com/external-css)

24) What is the difference between inline, embedded and external style sheets?

**Inline**: Inline Style Sheet is used to style only a small piece of code.

Syntax

1. **<htmltag** style="cssproperty1:value; cssproperty2:value;"**>** **</htmltag>**

**Embedded**: Embedded style sheets are put between the <head>...</head> tags.

Syntax

1. **<style>**
2. body {
3. background-color: linen;
4. }
5. h1 {
6. color: red;
7. margin-left: 80px;
8. }
9. **</style>**

**External**: This is used to apply the style to all the pages within your website by changing just one style sheet.

Syntax

1. **<head>**
2. **<link** rel="stylesheet" type="text/css" href="mystyle.css"**>**
3. **</head>**

25) What is RWD?

RWD stands for Responsive Web Design. This technique is used to display the designed page perfectly on every screen size and device, for example, mobile, tablet, desktop and laptop. You don't need to create a different page for each device.

26) What are the benefits of CSS sprites?

If a web page has a large number of images that take a longer time to load because each image separately sends out an HTTP request. The concept of CSS sprites is used to reduce the loading time for a web page because it combines the various small images into one image. It reduces the number of HTTP requests and hence the loading time.

27) What is the difference between logical tags and physical tags?

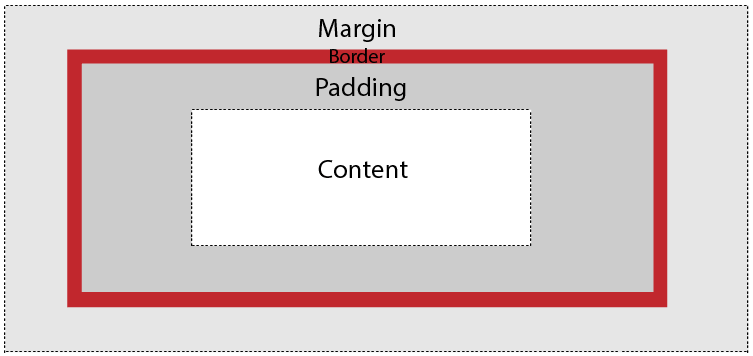
* Physical tags are referred to as presentational markup while logical tags are useless for appearances.
* Physical tags are newer versions, on the other hand, logical tags are old and concentrate on content.

28) What is the CSS Box model and what are its elements?

The CSS box model is used to define the design and layout of elements of CSS.

The elements are:

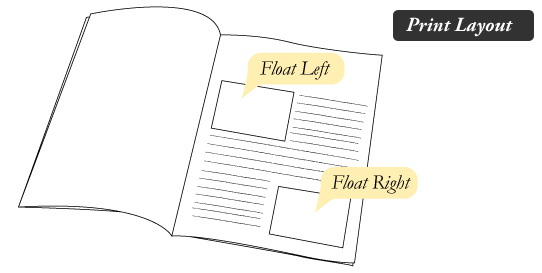
* Margin - It removes the area around the border. It is transparent.
* Border - It represents the area around the padding
* Padding - It removes the area around the content. It is transparent.
* Content - It represents the content like text, images, etc.



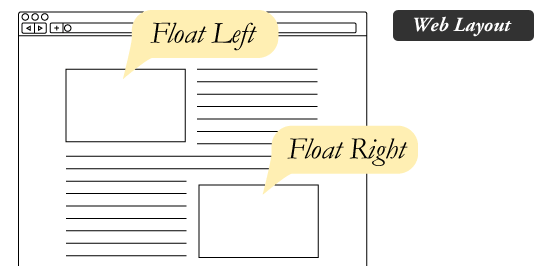
29) What is the float property of CSS?

The CSS float property is used to move the image to the right or left along with the texts to be wrapped around it. It doesn't change the property of the elements used before it.

To understand its purpose and origin, let's take a look at its print display. In the print display, an image is set into the page such that text wraps around it as needed.



Its web layout is also just similar to print layout.

  
[More details...](https://www.javatpoint.com/css-float)

30) How to restore the default property value using CSS?

In short, there is no easy way to restore to default values to whatever a browser uses.

The closest option is to use the 'initial' property value, which restores the default CSS values, rather than the browser's default styles.

31) What is the purpose of the z-index and how is it used?

The z-index helps to specify the stack order of positioned elements that may overlap one another. The z-index default value is zero and can take on either a positive or negative number.

An element with a higher z-index is always stacked above than a lower index.

Z-Index can take the following values:

* **Auto:** Sets the stack order equal to its parents.
* **Number:** Orders the stack order.
* **Initial:** Sets this property to its default value (0).
* **Inherit:** Inherits this property from its parent element.

32) Explain the difference between visibility: hidden and display: none?

**visibility: hidden** hides the element, but it occupies space and affects the layout of the document.

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. h1.vis {
6. visibility: visible;
7. }
9. h1.hid {
10. visibility: hidden;
11. }
12. **</style>**
13. **</head>**
14. **<body>**
15. **<h1** class="vis"**>**It is visible**</h1>**
16. **<h1** class="hid"**>**It is hidden**</h1>**
18. **<p>**Note - Second heading is hidden, but it still occupy space.**</p>**
19. **</body>**
20. **</html>**

**display: none** also hides the element but not occupy space. It will not affect the layout of the document.

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. h1.vis {
6. display: block;
7. }
9. h1.hid {
10. display: none;
11. }
12. **</style>**
13. **</head>**
14. **<body>**
15. **<h1** class="vis"**>**It is visible**</h1>**
16. **<h1** class="hid"**>**It is hidden**</h1>**
18. **<p>**Note - Second heading is hidden and not occupy space.**</p>**
19. **</body>**
20. **</html>**

33) What do you understand by W3C?

W3C stands for World Wide Web Consortium. Its purpose is to deliver the information of the World Wide Web. It also develops rules and guidelines for the Web.

34) What is tweening?

It is the process of generating intermediate frames between two images.

It gives the impression that the first image has smoothly evolved into the second one.

It is an important method used in all types of animations.

In CSS3, Transforms (matrix, translate, rotate, scale) module can be used to achieve tweening.

35) What is the difference between CSS2 and CSS3?

The main difference between CSS2 and CSS3 is that CSS3 is divided into different sections which are also known as modules. Unlike CSS2, CSS3 modules are supported by many browsers.

Apart from that, CSS3 contains new General Sibling Combinators which is responsible for matching the sibling elements with the given elements.

**Q1. What is the full form of CSS?**

CSS stands for [Cascading Style Sheets](https://www.edureka.co/blog/what-is-css/). It is a technology developed by the World Wide Web Consortium or W3C. It was developed to streamline the styling of webpages into a separate technology.

**Q2. Why was CSS developed?**

CSS was first developed in 1997 as a way for web developers to define the visual appearance of the web pages that they were creating. It was intended to allow developers to separate the content and structure of a website’s code from the visual design, something that had not been possible prior to this time.

The separation of structure and style allows [HTML](https://www.edureka.co/blog/what-is-html/) to perform more of the function that it was originally based on — the markup of content, without having to worry about the design and layout of the page itself, something commonly known as the *“look and feel”* of the page.

**Q3. What are the major versions of CSS?**

The following are the major versions of CSS

1. CSS 1
2. CSS 2
3. CSS 2.1
4. CSS 3
5. CSS 4

**Q4. What are the different ways you could integrate CSS into your HTML page?**

There are three ways that you could integrate a certain CSS style:

1. You can integrate your style using the style-tags in the head section of your HTML page.
2. You can integrate your style using inline-styling.
3. You can write your CSS in a separate file and add it to your HTML page using the link tag.

**Q5. What is the meaning of cascading? How do style sheets cascade?**

CSS brought about a revolution in web-development and how people perceive the process of building a website. Prior to the existence of CSS, elements had to be styled in an in-line fashion or the style were implemented in the head section of an HTML page. This was changed due to the cascading nature of CSS. Here are the three major ways CSS cascades:

1. **Elements** –  The same CSS style can be applied to multiple elements to achieve the same style.
2. **Multiple Style One Element** – Multiple styles can be applied to a particular HTML element to achieve a unique style.
3. **Same style, Multiple Pages** – The same stylesheet can be applied to different HTML pages altogether to achieve a template styling very quickly.

**Q6. What are the advantages of using CSS?**

Following are the advantages of using CSS:

* The style of several documents can be controlled from a single site by using them.
* Multiple HTML elements can have many documents, where classes can be created.
* To group styles in complex situations, selector and grouping methods are used.

**Q7. What are the disadvantages of using CSS?**

Following are the disadvantages of using CSS:

* Ascending by selectors is not possible
* Limitations of vertical control
* No expressions
* No column declaration
* Pseudo-class not controlled by dynamic behaviour
* Rules, styles, targeting specific text not possible

**Q8. Name a few prominent CSS frameworks.**

Below are the prominent CSS frameworks in the web development industry today:

* **Bootstrap**

****

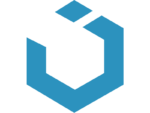
Bootstrap is the most popular **CSS framework** for developing responsive and mobile-first websites. **Bootstrap 4** is the newest version of Bootstrap

* **Foundation**

****

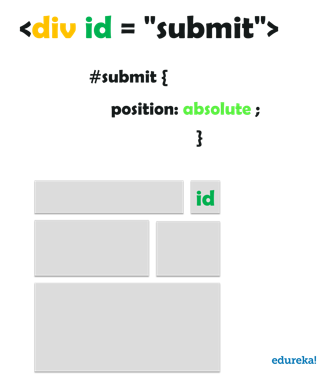
Foundation is a responsive front-end framework. Foundation provides a responsive grid and HTML and CSS UI components, templates, and code snippets, including typography, forms, buttons, navigation and other interface elements, as well as optional functionality provided by JavaScript extensions.

* **Semantic UI**Semantic UI is a modern front-end development framework, powered by LESS(CSS-preprocessor) and [jQuery](https://www.edureka.co/blog/jquery-tutorial/). It has a sleek, subtle, and flat design look that provides a lightweight user experience.
* **Ulkit**

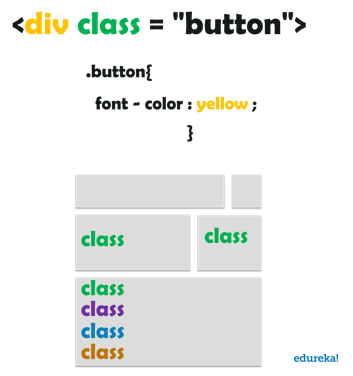
UIkit is a lightweight and modular front-end framework for developing fast and powerful web interfaces.

**Q9. What is the difference between the usage of an ID and a Class?**

**ID** – An ID is *unique*. A particular ID can be only assigned to a single element. IDs are used when specific styling is being tried to be achieved over a single element. Below is a pictorial example of how to use an ID.

****

**Class** – Just like the word suggests, a *class is a collective way of targetting HTML elements for styling.*Classes are not unique and multiple elements can have the same class. In fact, multiple classes can also be added to the same element to achieve the desired style and look. Below is an example of the usage of classes.

**Q10. What is the RGB stream?**  
RGB is a system of representing a certain colour in CSS. There are three streams in this nomenclature of representing a colour, namely the Red, Green and Blue stream. The intensity of the three colours is represented in numbers ranging from 0 to 256. This allows CSS to have a wide range of colours spreading across the entire spectrum of visible colours.

These conclude the easy section. Here is [getting started documentation for CSS.](https://devdocs.io/css/) Things are going to get a much more particular now. It’s time for intermediate CSS interview questions.

**Intermediate CSS Interview Questions**

**Q11. What are the ways to assign a certain colour to an element in CSS?**

CSS can assign a wide range of colours to elements using different notations. There are three notations as of now that are used that are explained below:

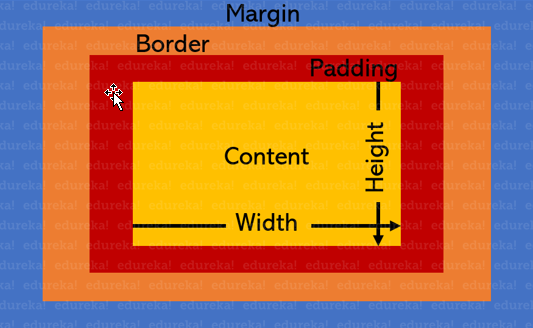
* **Hexadecimal notation**  
  A colour in hexadecimal string notation always begins with the character **“#”**. After that, the hexadecimal digits of the colour code is written. The string is case-insensitive.
* **RGB functional notation**RGB (Red/Green/Blue) functional notation, like hexadecimal string notation, represents colours using their red, green, and blue components (as well as, optionally, an alpha channel component for opacity). However, instead of using a string, the colour is defined using the CSS function**RGB()**. This function accepts as its input parameters the values of the red, green, and blue components and an optional fourth parameter, the value for the alpha channel.
* **HSL functional notation**Designers and artists often prefer to work using the HSL (Hue/Saturation/Luminosity) colour method. On the web, HSL colours are represented using HSL functional notation. The HSL() CSS function is very similar to the RGB() function in usage otherwise

**Q12. Explain the CSS Box Model and its different elements.**

The CSS box model describes the rectangular boxes that are generated for elements in the document tree and laid out according to the visual formatting model. Each box has a content area (e.g. text, an image, etc.) and an optional surrounding**padding**,**border**, and**margin**areas.

The CSS box model is responsible for calculating:

* How much space a block element takes up.
* Whether or not borders and/or margins overlap, or collapse.
* A box’s dimensions.



The box model has the following rules:

* The dimensions of a block element are calculated by width, height, padding, borders, and margin.
* If no height is specified, a block element will be as high as the content it contains, plus padding.
* If no width is specified, a non-floated block element will expand to fit the width of its parent minus padding.
* The height of an element is calculated by the content’s height.
* The width of an element is calculated by the content’s width.
* By default, padding and border are not part of the width and height of an element.

**Q13. What is the z-index in CSS?**

The z-index helps specify the stack order of positioned elements that may overlap one another. The z-index default value is zero and can take on either a positive or negative number.

An element with a higher z-index is always stacked above than a lower index.

Z-Index can take the following values:

* **Auto:** Sets the stack order equal to its parents.
* **Number:** Orders the stack order.
* **Initial:** Sets this property to its default value (0).
* **Inherit:** Inherits this property from its parent element.

**Q14. What are CSS Sprites?**

CSS sprites combine multiple images into one single larger image. It is a commonly-used technique for icons (Gmail uses it). This is how you could implement it:

1. Use a sprite generator that packs multiple images into one and generates the appropriate CSS for it.
2. Each image would have a corresponding CSS class with background-image, background-position and background-size properties defined.
3. To use that image, add the corresponding class to your element.

**CSS Interview Questions**

**Q15. Mention a few benefits of using CSS Sprites.**

CSS sprites come with their own advantages. Here are a few of them –

* Reduce the number of HTTP requests for multiple images (only one single request is required per sprite sheet). But with HTTP2, loading multiple images is no longer much of an issue.
* Advance downloading of assets that won’t be downloaded until needed, such as images that only appear upon :hover pseudo-states. Blinking wouldn’t be seen.

**Q16. What are pseudo-elements in CSS?**

A CSS pseudo-element is a keyword added to a selector that lets you style a specific part of the selected element(s). They can be used for decoration (:first-line, :first-letter) or adding elements to the markup (combined with content: ...) without having to modify the markup (:before, :after).

* :first-line and :first-letter can be used to decorate text.
* Triangular arrows in tooltips use :before and :after. This encourages separation of concerns because the triangle is considered a part of styling and not really the DOM. It’s not really possible to draw a triangle with just CSS styles without using an additional HTML element.

**Q17. How will you target an h2 and h3 with the same styling?**

You can target multiple elements by separating the separators with a comma (,)

h2, h3 {

color: blue;

}

**Q18. What is the float property used for in CSS?**

The float CSS property places an element on the left or right side of its container, allowing text and inline elements to wrap around it. The element is removed from the normal flow of the page, though it still remains a part of the flow (in contrast to absolute positioning). Below is the usage of float

float: none;

float: left;

float: right;

**Q19.  What are the different modules used in the current version of CSS?**

There are several modules in CSS. Below are a few of them:

* Selectors
* Box Model
* Backgrounds and Borders
* Text Effects
* 2D/3D Transformations
* Animations
* Multiple Column Layout
* User Interface

**Q20. What are the different media types allowed by CSS?**

There are four types of @media properties (including *screen*):

* **all** – for all media type devices
* **print** – for printers
* **speech** – for screenreaders that “reads” the page out loud
* **screen** – for computer screens, tablets, smart-phones etc.

Here is an example of print-media type’s usage:

@media print {

h1 {

background-color: yellow;

 }

}

**Q22. What are the different units used in CSS?**

CSS has two types of lengths. Relative length and absolute length. Different units are used for them.

**Relative Length**

|  |  |
| --- | --- |
| **UNIT** | **DESCRIPTION** |
| em | Relative to the font-size of the element (2em means 2 times the size of the current font) |
| ex | Relative to the x-height of the current font (rarely used) |
| ch | Relative to the width of the “0” (zero) |
| rem | Relative to font-size of the root element |
| vw | Relative to 1% of the width of the viewport\* |
| vh | Relative to 1% of the height of the viewport\* |
| vmin | Relative to 1% of viewport’s\* smaller dimension |
| vmax | Relative to 1% of viewport’s\* larger dimension |
| % | Relative to the parent element |

**Absolute Length**

|  |  |
| --- | --- |
| **UNIT** | **DESCRIPTION** |
| CM | centimetres |
| MM | millimetres |
| IN | inches (1in = 96px = 2.54cm) |
| PX | pixels (1px = 1/96th of 1in) |
| PT | points (1pt = 1/72 of 1in) |
| PC | picas (1pc = 12 pt) |

**Q23. How do you control image repetition using CSS?**

You can use the **background-repeat** property to control image.

[[](https://www.edureka.co/full-stack-developer-training)](https://www.edureka.co/full-stack-developer-training" \t "_blank)

**[Full Stack Web Developer Masters Program](https://www.edureka.co/full-stack-developer-training" \t "_blank)**

[Explore Curriculum](https://www.edureka.co/full-stack-developer-training" \t "_blank)

h1 {

background-repeat: none;

 }

**Q24. What is the general nomenclature of writing CSS?**

If you look at the above image, you will notice that the styling commands are written in a *property & value fashion*. The property is, *font-colour*while the value is *yellow.*The CSS syntax also incorporates a statement terminator in the form of a semi-colon **‘;’.**The entire style in then wrapped around curly braces and then attached to a selector(*.boxes* here). This creates a style that can be added to a style sheet and then applied to an HTML page. This is how CSS is written everywhere.

**Q25. What will this piece of CSS code do to an element? .container { margin: 0 auto; }**

When you have specified a width on the object that you have applied margin: 0 auto to, the object will sit centrally within its parent container. Specifying auto as the second parameter basically tells the browser to automatically determine the left and right margins itself, which it does by setting them equally. It guarantees that the left and right margins will be set to the same size. The first parameter 0 indicates that the top and bottom margins will both be set to 0.

margin-top:0; margin-bottom:0; margin-left:auto; margin-right:auto;

Therefore, to give you an example, if the parent is 100px and the child is 50px, then the auto property will determine that there’s 50px of free space to share between margin-left and margin-right:

var freeSpace = 100 - 50;  
var equalShare = freeSpace / 2;

Which would give:

margin-left:25;

margin-right:25;

**Q26. What is the overflow property in CSS used for?**

The **overflow** property specifies what should happen if content **overflows** an element’s box. This property specifies whether to clip content or to add scrollbars when an element’s content is too big to fit in a specified area. Below are the overflow options available in CSS –

overflow: auto;

overflow: none;

overflow: scroll;

overflow: visible;

**Q27. What is the property that is used for controlling *image-scroll*?**

The background-attachment property sets whether a background image scrolls with the rest of the page, or is fixed. Here is an example of a background-image that will not scroll with the page (fixed):

body {  
  background-image: url("img\_tree.gif");  
  background-repeat: no-repeat;  
  background-attachment: fixed;  
}

**Q28. What is responsive web design?**

**Responsive design** is an approach to web page creation that makes use of flexible layouts, flexible images and cascading style sheet media queries. The goal of **responsive design** is to build web pages that detect the visitor’s screen size and orientation and change the layout accordingly.

**Q29. What is the difference between {visibility: hidden} and {display: none}?**

display:none means that the tag in question will not appear on the page at all (although you can still interact with it through the DOM). There will be no space allocated for it between the other tags.

visibility:hidden means that unlike display:none, the tag is not visible, but space is allocated for it on the page. The tag is rendered, it just isn’t seen on the page.

For example:

test | <span style="[style-tag-value]">Appropriate style in this tag</span> | test

Replacing [style-tag-value] with display:none results in:

test | | test

Replacing [style-tag-value] with visibility:hidden results in:

test |                        | test

**Q30. Explain the concept of specificity in CSS.**

**Specificity** is the means by which browsers decide which CSS property values are the most relevant to an element and, therefore, will be applied. Specificity is based on the matching rules which are composed of different sorts of CSS selectors.

Specificity is a weight that is applied to a given CSS declaration, determined by the number of each selector type in the matching selector. When multiple declarations have equal specificity, the last declaration found in the CSS is applied to the element. Specificity only applies when the same element is targeted by multiple declarations. As per CSS rules, directly targeted elements will always take precedence over rules which an element inherits from its ancestor.

**CSS Interview Questions**

**Q31. What are the various font-related attributes in CSS?**

Below are the different font-related attributes available in CSS:

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

**Q32. What is the use of box-shadow in CSS?**

The box-shadow CSS property adds shadow effects around an element’s frame. You can set multiple effects separated by commas. A box-shadow is described by X and Y offsets relative to the element, color, blur and spread radii. Below are a few implementations of box-shadow

box-shadow: 10px 5px 5px red;

box-shadow: 60px -16px teal;

box-shadow: 12px 12px 2px 1px rgba(0, 0, 255, .2);

box-shadow: inset 5em 1em gold;

**Q34. What are contextual selectors?**

Contextual selectors in CSS allow you to specify different styles for different parts of your document. You can assign styles directly to specific HTML tags, or, you could create independent classes and assign them to tags in the HTML. Either approach lets you mix and match styles.

**Q35. How would you style an image or element to have rounded corners?**

Use the border-radius property to add rounded corners to an image. 50% will make the image circular.

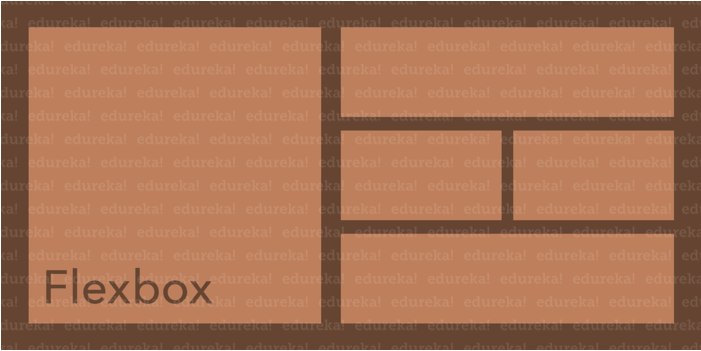
border-radius: 50%;

Now let’s discuss some of the more advanced CSS interview questions.

**Advanced CSS Interview Questions**

**Q36. What is CSS flexbox?**

The flexbox layout officially called CSS**flexible box layout module** is a new layout module in CSS3. It is made to improve the items align, directions and order in the container even when they are with 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.

**Q37. How does a browser determine what elements match a CSS selector?**

Browsers match selectors from rightmost (key selector) to left. Browsers filter out elements in the DOM according to the key selector and traverse up its parent elements to determine matches. The shorter the length of the selector chain, the faster the browser can determine if that element matches the selector.

For example with this selector p span, browsers firstly find all the **<span>** elements and traverse up its parent all the way up to the root to find the**<p>** element. For a particular **<span>**, as soon as it finds a **<p>**, it knows that the <span> matches and can stop its matching.

**Q38. Explain the scenario you would use translate() instead of absolute positioning?**

Translate is a value of CSS transform. Changing transform or opacity does not trigger browser reflow or repaint but does trigger compositions; whereas changing the absolute positioning triggers reflow. Transform causes the browser to create a GPU layer for the element but changing absolute positioning properties uses the CPU. Hence translate() is more efficient and will result in shorter paint times for smoother animations.

When using translate(), the element still occupies its original space (sort of like position: relative), unlike in changing the absolute positioning.

**Q39. Explain the difference in approach when designing a responsive website over a mobile-first strategy?**

These two approaches are not exclusive. Making a website responsive means some elements will respond by adapting its size or other functionality according to the device’s screen size, typically the viewport width, through CSS media queries.

For example, making the font size smaller on smaller devices.

@media (min-width: 601px) {

.my-class {

font-size: 24px;

}

**Question: What is CSS?**

**Answer:** CSS or Cascading Style Sheets is a styling language that is simple enough for HTML elements that find usage in web designing. Besides, the application of CSS is also found in XHTML.

**Question: What are the different variations of CSS?**

**Answer:**The different variations of CSS include:

* CSS 4
* CSS 3
* CSS 2.1
* CSS 2
* CSS 1

**Question: What are the benefits of using CSS?**

**Answer:** There are multiple benefits of using CSS, such as:

* Accessibility
* Page reformatting
* Site-wide consistency
* Bandwidth
* Separation of the bandwidth of presentation

**Question: What are the disadvantages of CSS?**

**Answer:**The disadvantages of using CSS are:

* Target specific texts, styling, and rules are not allowed.
* There are no expressions.
* There is no dynamic behavior to control pseudo-class.
* It is not possible to ascend by selectors.
* Vertical control is limited.
* The column declaration is not there.

**Question: What is meant by CSS frameworks?**

**Answer:**CSS frameworks are pre-planned libraries, allowing a more convenient web page styling which is also compliant with standards.

**Question: Give a few examples of modules found in the present version of CSS.**

**Answer:**Some of the commonly used modules in CSS include:

* Box Model
* Selectors
* Text effects
* Backgrounds and Borders
* Animations
* 2D/3D Transformations
* User Interface.
* Multiple Column Layout

**Question: What is the difference between CSS2 and CSS3?**

**Answer:**

* In CSS2, everything is present in a single document while CSS3 is categorized into multiple sections called modules.
* Graphics-related features such as Box-shadow or Border-radius, flexbox, etc. can be found which are absent incase of CSS2.
* Properties such as background-position, background-image, and background-repeat styles can be helpful for using multiple background images in a single webpage.
* CSS3 versions support many new browsers.
* CSS3 introduces many new selectors in the form of pseudo-elements and pseudo-classes.

**Question: Name a few CSS style components**

**Answer:**A few CSS style components include Property, Value, and Selector

**Question: What do you understand by CSS opacity?**

**Answer:**The technical definition of opacity is the degree to which light is allowed to pass through an object.

CSS opacity is the property used to describe the transparency of an element. Or, in other words, it explains how clear the image is.

**Question: How can the background color of an element be changed?**

**Answer:**The background color of an image can be changed using the background-color property.

**body**   
{  
background-color: coral;  
}

**Question: How can image repetition of the backup be controlled?**

**Answer:**Background-repetition property controls the repetition of images in the background. Use no-repeat if the image is to be displayed once in the background.

<**html**>  
<**head**>  
<**style**>  
**body** {  
background-image: url("/css/images/css.jpg");  
background-repeat: repeat;  
}  
</**style**>  
</**head**>  
<**body**>  
<**p**>CSS Background Image: Hackr.io</**p**>  
</**body**>  
</**html**>

**Question: What is the use of the background-position property?**

**Answer:**It can be used to define the initial position of a background image. The default position is the top left of the page. The positions that can be set include top, bottom, left, right, and center.

<**html**>  
<**head**>  
<**style**>  
body {  
background-image: url('hackr.gif');  
background-repeat: no-repeat;  
background-attachment: fixed;  
background-position: center;  
}  
<**p**>CSS Background Position: Hackr.io</**p**>  
</**body**>  
</**html**>

**Question: Which property controls the image scroll in the background?**

**Answer:**Image scroll in the background can be controlled using the background-scroll property.

**Question: Why should background and color be used as separate properties?**

**Answer:**This is done with mainly two purposes:

1. It makes the style sheets more legible. Background property, which is complex in itself, becomes all the more complex with color.
2. Color is an in-built property while the background is not one. And this can lead to a lot of confusion.

**Question: How to center block elements using CSS1?**

**Answer:**In order to center the block-level elements, we need to set the margin-right and margin-left properties to explicit values.

**Question: How to maintain the CSS specifications?**

**Answer:**It is maintained using the World Wide Web Consortium.

**Question: What are the ways to integrate CSS as a web page?**

There are three methods to integrate CSS in the form of a web page:

* **Embedded:**There can be a style element inside the head element inside which we can place the code.
* **Inline:** CSS applied HTML elements can be found using style attributes.
* **Imported or Linked:**In the case of linked or imported CSS, the CSS is placed in an external file and a link element is used to link it.

**Question: What are the external style sheets?**

**Answer:**External style sheets are sheets used externally which can be linked to the HTML pages.

**Question: What is embedded style sheets?**

**Answer:**These are the sheets where style sets for the entire HTML document are defined in a single place. For doing this, the style sheet information under the style tags should be embedded into an HTML document.

**Question: What are the advantages and disadvantages of using external style sheets?**

**Answer:**

**Advantages**

* There can be many documents for multiple HTML elements, along with many classes.
* Multiple documents with various styles can be controlled using different styles.
* Selector and grouping methods can be used for grouping styles in composite situations.

**Disadvantages**

* For rendering the document, external style sheets have to be loaded.
* It is not suitable for small style definitions.
* For importing documents with style information, an additional download is required.

**Question: What are the advantages and disadvantages of embedded style sheets?**

**Answer:**

**Advantages**

* You can create different tag types inside a document.
* The additional download is not required, unlike external style sheets.
* In complicated situations, you can use Grouping and Selector methods to apply styles.

**Disadvantage**

* You cannot control multiple documents.

Advance CSS Interview Questions

**Question: What is the meaning of the CSS selector?**

**Answer:**Just like there is the string in HTML, there is a selector in CSS which is used for the purpose of linking HTML and style sheet elements.

**Question: What are the media types allowed by CSS?**

**Answer:**CSS allows different media types such as speech, audio, visual and tactile media, paged or continuous media, bitmap or grid media, and even interactive media.

**Question: What is the ruleset?**

**Answer:**Rulesets can be used to identify the selectors attached to one another.

It consists of two different parts- selector and declaration.

**Question: How case-sensitive is CSS?**

**Answer:**CSS is not case-sensitive but the URLs of images and font families are case-sensitive. Only in case of usage of XML declarations with XHTML DOCTYPE on the page, we find CSS to be case-sensitive.

**Question: What is a declaration block?**

**Answer:**It is basically a catalog of directions comprising the property, followed by a colon, and finally the value enclosed within braces.

**Question: What are the different font attributes available?**

**Answer:**The various font attributes available include font-variant, font-weight, font-style, font-family, line-height of font size, caption, icon, and font-family.

**Question: How does importing a file make it easy to insert?**

**Answer:**Importing of files helps in integrating external sheets that can be inserted in multiple sheets. There can be different sheets and styles for different functions.

**Question: What is the difference between physical and logical tags?**

**Answer:**Logical tags are older as compared to the physical ones and mainly focus on the content. They hardly find any usage in terms of presentation. Logical tags do not find any application in terms of aesthetics while the physical ones find their application in presentation too.

**Question: How does the style sheet concept differ as compared to HTML?**

**Answer:**Style sheets associate a styling factor with them. HTML, on the other hand, offers an easy structure method. Style sheets also feature better formatting options and browsing capabilities.

**Question: Is it possible to add multiple declarations in CSS?**

**Answer:**Yes, we can do this with the help of semicolons.

**Question: What are pseudo-elements?**

**Answer:**These elements are used to provide special effects to certain selectors. CSS finds usage in applying styles in the HTML markups. If by any chance, additional styling or markup is not feasible for the document, this feature of pseudo-elements helps by allowing extra markup without hampering the original document.

**Question: What is Tweening?**

**Answer:**It is the process by which intermediate frames between two pictures are created in order to find the appearance of the first image developing into the second.

**Question: What are CSS counters?**

**Answer:**These are variables that can be increased by using CSS that inspect and find the number of times of usage of variables.

**Question: What is meant by a universal selector?**

**Answer:**It is a selector which can match the name of any element type, rather than selecting the elements of a specific category.

**Question: How to select all the elements of a paragraph?**

**Answer:**The elements in a paragraph can be selected using the p[lang] command.

**Question: How are percentage values decoded in CSS?**

**Answer:**The percentage values are decoded in CSS using a percentage symbol.

**Question: What is RWD?**

**Answer:**RWD is the abbreviated form of Responsive Web Design. It is used to display the designed page suitably on any kind of screen size depending on the device under consideration.

**Question: What is the float property of CSS?**

**Answer:**It is used to position an image to the left or right as required including the text wrapping around it. The property of the elements used before it remains unchanged.

**Question: What is the difference between visibility: hidden and display:none?**

**Answer:**The difference between visibility:hidden and display:none properties is that in the case of the former, the elements are hidden but not deleted. No space is consumed.

In case of the latter, the element is hidden and the layout is affected, that is, some space is taken up.

<!DOCTYPE html>  
<**html**>  
<**head**>  
<**style**>  
**h3**   
{  
display: none;  
}  
</**style**>  
</**head**>  
<**body**>  
<**h2**>This heading is visible</**h2**>  
<**h3**>This is a hidden heading</**h3**>  
<**p**>The hidden heading does not take up space even after hiding it since we have used display: none;.</**p**>  
</**body**>  
</**html**>  
visibility:hidden  
<!DOCTYPE html>  
<**html**>  
<**head**>  
<**style**>  
**h3** {  
visibility:hidden;  
}  
</**style**>  
</**head**>  
<**body**>  
<**h2**>This heading is visible</**h2**>   
<**h3**>This is a hidden heading</**h3**>   
<**p**>The hidden heading takes up space even after hiding it.</**p**>  
</**body**>  
</**html**>

**Question: What is the difference between the class selector and ID selector?**

**Answer:**A class selector takes an overall block while an ID selector takes a single element that differs from others.

**Use of CSS class selector**

<**style**>  
.center {  
text-align: center;  
color:red;  
}  
</**style**>

**Use of CSS ID Selector**

<**style**>  
#para1  
{  
text-align: centre;  
color:red;  
}  
</**style**>

**Q #1) What is CSS?**

**Answer:**CSS outlines the style of an HTML webpage. It is a language by which we can set the behavior of an HTML webpage. It describes how the HTML content will be shown on screen.

CSS controls the layout of several HTML web pages. CSS is referred to as the Cascading Style Sheet.

**Q #2) Name all the modules which are used in the current version of CSS.**

**Answer: There are several modules in CSS as stated below:**

* Selectors
* Box Model
* Backgrounds and Borders
* Text Effects
* 2D/3D Transformations
* Animations
* Multiple Column Layout
* User Interface.

**Q #3) Distinguish between CSS2 and CSS3.**

**Answer: The differences between CSS2 and CSS3 are as follows:**

* CSS3 is divided into two various sections which are called a module. Whereas in CSS2 everything accedes into a single document with all the information in it.
* CSS3 modules are supported almost on every browser and on the other hand modules of CSS and CSS2 are not supported in every browser.
* In CSS3, we will find that many graphics related characteristics have been introduced like Border-radius or box-shadow, flexbox.
* In CSS3, a user can precise multiple background images on a webpage by using properties like background-image, background-position, and background-repeat styles.

**Q #4) Cite different types of CSS.**

**Answer: There are three types of CSS as mentioned below:**

* **External:**These are written in separate files.
* **Internal:**These are cited at the top of the web page code document.
* **Inline:** These are written right next to the text.

**Q #5) Why is the external style sheet useful?**

**Answer:**External style sheet is very useful as we write all the styling codes in a single file and it can be used anywhere by just referring to the link of that external style sheet file.

So, if we do any changes in that external file, then the changes can also be observed on the webpage. Thus we can say that it is very useful and it makes your work easy while working on larger files.

**Q #6) What are the uses of an embedded style sheet**?

**Answer:**Embedded style sheet gives us the privilege to define styles in one place in an HTML document.

We can generate multiple classes using an embedded style sheet to use on multiple tag types of a web page and also there is no extra downloading required for importing the information.

**Example:**

<!DOCTYPE html>

<html>

<head>

<style type="text/css">

p {

  font-family: georgia, serif;

  font-size: x-large;

  color:#ff9900;

  }

a:hover {

  color: LimeGreen;

  text-decoration: none;

  }

</style>

</head>

<body>

<p>Embedded style sheet gives us the privilege to define styles at one place in a HTML document.

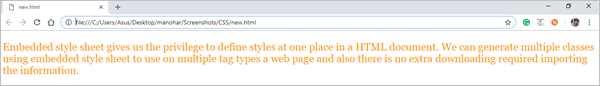
We can generate multiple classes using embedded style sheet to use on multiple tag types a web page

and also there is no extra downloading required importing the information.

</p>

</body>

</html>

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2018/10/embedded.png)

**Q #7) How to use CSS selector?**

**Answer:**By using the CSS selector, we can choose the content which we want to style so that we can say that it is a bridge between the style sheet and the HTML files.

The syntax for CSS selector is “select” HTML elements created on their id, class, type, etc.

**Q #8) Explain the concept of Tweening.**

**Answer:**Tweening is the process in which we create intermediate frames between two images to get the appearance of the first image which develops into the second image.

It is mainly used for creating animation.

**Q #9) Define CSS image scripts.**

**Answer:**CSS image scripts are a group of images that are placed into one image. It reduces the load time and request number to the server while projecting multiple images into a single web page.

**Q #10) Explain the term Responsive web design.**

**Answer:**It is a method in which we design and develop a web page according to the user activities and conditions which are based on various components like the size of the screen, portability of the web page on the different devices, etc. It is done by using different flexible layouts and grids.

**Q #11) What are CSS counters?**

**Answer:**CSS counters are variables that can be incremented by rules of CSS that inspector track how many times the variable has been used.

**Q #12) What is CSS specificity?**

**Answer:**CSS specificity is a score or rank that decides which style declaration has to be used to an element. (\*) this universal selector has low specificity while ID selectors have high specificity.

**There are four categories in CSS which authorize the specificity level of the selector.**

* Inline style
* IDs
* Classes, Attributes, and pseudo-classes.
* Elements and pseudo-elements.

**Q #13) How can we calculate specificity?**

**Answer:**To calculate specificity we will start with 0, then we have to add 1000 for each ID and we have to add 10 to the attributes, classes or pseudo-classes with each element name or pseudo-element and later we have to add 1 to them.

**Example:**

h2

             #content h2

           <div id=”content”>

              <h2 style=”color:#FF0000”>heading</h2>

            </div>

**Q #14) How do we make a rounded corner by using CSS?**

**Answer:**We can make a rounded corner by using the property “border-radius”. We can apply this property to any element.

**Example:**

<html>

<head>

<style>

#rcorners1 {

    border-radius: 25px;

    background: #715751;

    padding: 20px;

    width: 200px;

    height: 150px;

}

</style>

</head>

<body>

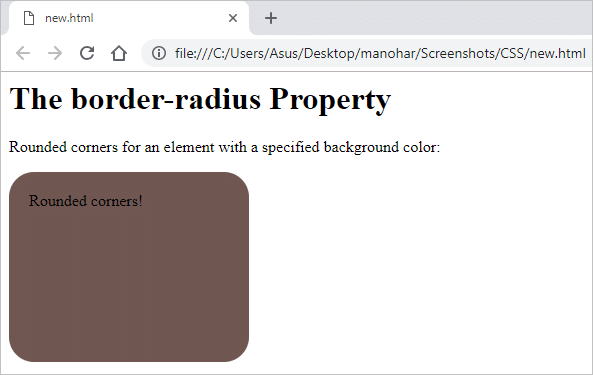
<h1>The border-radius Property</h1>

<p>Rounded corners for an element with a specified background color:</p>

<p id="rcorners1">Rounded corners!</p>

</body>

</html>

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2018/10/border-radius.png)

**Q #15) How will you add border images to an HTML element?**

**Answer:**We can set the image to be used as the border-image alongside an element by using the property of CSS “border-image”.

**Example:**

#borderimg {

    border: 15px solid transparent;

    padding: 20px;

    border-image: url(border.png) 30 round;

}

**Q #16) What are gradients in CSS?**

**Answer:**It is a property of CSS which allows you to display a smooth transformation between two or more than two specified colors.

There are two types of gradients that are present in CSS. **They are:**

* Linear Gradient
* Radial Gradient

**Q #17) What is CSS flexbox?**

**Answer:**It allows you to design a flexible responsive layout structure without using any float or positioning property of CSS. To use CSS flexbox you need to define a flex container initially.

**Example:**

<!DOCTYPE html>

<html>

<head>

<style>

.flex-container {

  display: flex;

  background-color: #f4b042;

}

.flex-container > div {

  background-color: #d60a33;

  margin: 10px;

  padding: 20px;

  font-size: 30px;

}

</style>

</head>

<body>

<div class="flex-container">

  <div>1</div>

  <div>2</div>

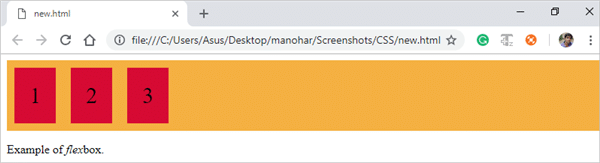
  <div>3</div>

</div>

<p> Example of  <em>flex</em>box.</p>

</body>

</html>

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2018/10/flexbox.png)

**Q #18) Write all the properties of the flexbox.**

**Answer:** There are several properties of the flexbox that are used in the HTML webpage.

**They are:**

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

**Q #19) How to align image vertically in a division that spans vertically on the whole webpage?**

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2018/10/align-image-vertically-in-a-division.png)

**Answer:**It can be done by using the syntax verticle-align: middle in the <div1> element and even we can bind the two text spans around with another span and after this, we have to use verticle-align: middle in the content #icon.

**Q #20) What is the difference between padding and margin?**

**Answer:**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

**Margin property has some defined values as shown below.**

* **Auto –** Using this property browser calculates the margin.
* **Length –** It sets the margin values in px,pt,cm etc.
* **% –** It sets the width % of the element.
* **Inherit –** By this property we can inherit the margin property from the parent element.

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

Negative values are not allowed in padding.

div {

padding-top: 60px;

padding-right: 40px;

padding-bottom: 50px;

padding-left: 70px;

}

**Q #21) What is the use of the Box Model in CSS?**

**Answer:**In CSS, the box model is a box that binds all the HTML elements and it includes features like margins, border, padding, and the actual content.

By using a box model we will get the authority to add the borders all around the elements and we can also define the space between the elements.

**Q #22) How can we add icons to the web page?**

**Answer:**We can add icons to the HTML webpage by using an icon library like font-awesome.

We have to add the name of the given icon class to any inline HTML element. (<i> or <span>) . Icons in the icon libraries are scalable vectors that can be customized with CSS.

**Q #23) What is a CSS pseudo-class?**

**Answer:**It is a class that is used to define a special state of an HTML element.

This class can be used by styling an element when a user snooped over it and also it can style an HTML element when it gets the focus.

selector:pseudo-class {

property:value;

}

**Q #24) Explain the concept of pseudo-elements in CSS.**

**Answer:**It is a feature of CSS which is used to style the given parts of an element.

**For Example**, we can style the first letter or line of an HTML element.

selector::pseudo-element {

property:value;

}

**Q #25) What is CSS opacity?**

**Answer:**It is the property that elaborates on the transparency of an element.

By this property, we can transparent the image that can take the values from 0.0-1.0. If the value is lower, then the image is more transparent. IE8 and earlier versions of the browser can take the values from 0-100.

img {

opacity: 0.6;

filter: alpha(opacity=60); /\* For IE8 and earlier \*/}

**Q #26) Write all the position states used in CSS.**

**Answer: In CSS, there are four position states as stated below:**

* Static(default)
* Relative
* Fixed
* Absolute

**Q #27) What are navigation bars in CSS?**

**Answer:**By using navigation bars we can make an ordinary HTML page into a user-specific and more dynamic web page. Basically, it is a list of links, hence use of <ul> and <li> elements makes the perfect sense.

ul {

list-style-type: none;

margin: 0;

padding: 0;

}

**Q #28) What are the differences between relative and absolute in CSS?**

**Answer:**The main difference between relative and absolute is that “relative” is used for the same tag in CSS and it means that if we write the left:10px then the padding will shift to 10px in the left while absolute is totally relative to the non-static parent.

It means, if we write left:10px then the result will be 10px far from the left edge of the parent element.

**Q #29) Define ‘important’ declarations used in CSS.**

**Answer:**Important declarations are defined as that declaration which is having more importance than the normal declaration.

While executing, these declarations override the declaration which is having less importance.

**For example,** if there are two users having an important declaration then one of the declarations will override the declaration of another user.

**For Example:**  
**Body {background: #FF00FF !important; color: blue}**

In this body, background has more weight than the color.

**Q #30) Define different cascading methods that can be used inside the cascading order.**

**Answer: Cascading order is itself a sorting method that allows many other different sorting methods:**

**a) Sort by origin:** There are some rules which can provide an alternate way defined as:

* The normal weight of the style sheet of a particular provider will be overridden by the increased weight of the user’s style sheet.
* Stylesheet rules of a particular user will be overridden by the normal width of the provider’s style sheet.

**b) Sort by selector’s specificity:** Less specific selector is been overridden by the more specific selector.

**For example**, A contextual selector is less specific in comparison to the ID selector which is a more specific one and with that contextual selector is been overridden by the ID selector.

**c) Sort by order specified:** This comes in the scenario when the two selectors are of same weight and the other properties than the specification which will be seen for overriding.

**Example:**

All other styles will be seen overridden if the style attribute is used for inline style.

And also, if the link element is used for external style, then it will override the imported style.

**Q #31) Differentiate between inline and block element.**

**Answer:**Inline element does not have an element to set width and height and also it does not have the line break.

**Example:** em, strong, etc.

**Block element specification:**

* They do have the line break.
* They define the width by setting a container and also allow setting height.
* It can also contain an element that occurs in the inline element.

**Example:**

width and height  
max-width and max-height  
min-width and min-height  
hi (i=1-6)- heading element  
p- Paragraph element.

**Q #32) How is the concept of inheritance applied in CSS?**

**Answer:**Inheritance is a concept in which the child class will inherit the properties of its parent class. It is a concept which is been used in many languages and is the easy way of defining the same property again.

It is used in CSS to define the hierarchy from the top level to the bottom level. Inherited properties can be overridden by the children’s class if the child uses the same name.

**Example:**

Body {font-size: 15pt;}

**And another definition is being defined in the child class**

Body {font-size: 15pt;}  
H1 {font-size: 18pt;}

All the paragraph text will be displayed using the property and will be defined in the body except for the H1 style which will show the text in font 18 only.

**Q #33) Differentiate between the ID and class.**

**Answer:**Both ID and class is been used in HTML and assigns the value from CSS.

**Please find below the differences:**

* The ID is a kind of element which uniquely assigns a name to a particular element whereas class has an element with a certain set of properties that can be used for the complete block.
* The id can be used as an element because it can uniquely identify it whereas class is also defined to block the elements and applies too many tags wherever it is used.
* ID provides the restriction to use its properties to one specific element whereas in class the inheritance is applied to a specific block or group of the element.

**[What is new in CSS3?](https://www.zeolearn.com/interview-questions/css3" \l "collapse-beginner-121)**

CSS3 extends CSS2.2 (or rather CSS 2) and it is the latest upgrade to the Cascading Style Sheets language. It brings a lot of new features and additions. Some of these features include rounded corners on HTML elements, shadows on divs and texts, gradient-fills, animations, as well as new layouts like multi-columns, flexible box or grid layouts.

Let’s have a look at everything 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.   
  One is the :root selector, which allows designers to point to the root element of a document. In HTML, it would be <html>.  
  Similar to the :first-child selector, the :last-child was added. With it one can select the last element named of a parent element.  
  A new user interaction pseudo-class selector was added, the :target selector. It can be used to highlight a span or div when it is referred to within the page itself.  
  The negation pseudo-class selector,:not can be coupled with almost any other selector that has been implemented.
* 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. We can specify the color, height, width, blur, and offset of one or multiple inner and/or outer drop shadows on your elements.
* 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.
* Radial gradients have been added in CSS3. Radial gradients are circular or elliptical gradients. Rather than proceeding along a straight axis, colors blend out from a starting point in all directions.
* 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.

**Explain what elements will match each of the following CSS selectors:**

1. div, p
2. div p
3. div > p
4. div + p
5. div ~ p

Hide answer

1. div, p - Selects all <div> elements and all <p> elements
2. div p - Selects all <p> elements that are anywhere inside a <div> element
3. div > p - Selects all <p> elements where the immediate parent is a <div> element
4. div + p - Selects all <p> elements that are placed immediately after a <div> element
5. div ~ p - Selects all <p> elements that are anywhere preceded by a <div> element

**Explain the meaning of each of these CSS units for expressing length:**

* **cm**
* **em**
* **in**
* **mm**
* **pc**
* **pt**
* **px**

Hide answer

* cm - centimeters
* em - elements (i.e., relative to the font-size of the element; e.g., 2 em means 2 times the current font size)
* in - inches
* mm - millimeters
* pc - picas (1 pc = 12 pt = 1/6th of an inch)
* pt - points (1 pt = 1/72nd of an inch)
* px - pixels (1 px = 1/96th of an inch)

**Apply to Join Toptal's Development Network**

and enjoy reliable, steady, remote Freelance CSS Developer Jobs.

[**Apply as a Freelancer**](https://www.toptal.com/talent/apply)

**In CSS3, how would you select:**

* **Every**<a>**element whose href attribute value begins with “https”.**
* **Every**<a>**element whose href attribute value ends with “.pdf”.**
* **Every**<a>**element whose href attribute value contains the substring “css”.**

Hide answer

Select every <a> element whose href attribute value begins with “https”:

a[href^="https"]

Select every <a> element whose href attribute value ends with “.pdf”:

a[href$=".pdf"]

Select every <a> element whose href attribute value contains the substring “css”:

a[href\*="css"]

**Given the following HTML:**

<div id="page">

<h1>Heading Title</h1>

<h2>Subheading Title</h2>

<h2>Subheading Title</h2>

<h1>Heading Title</h1>

<h2>Subheading Title</h2>

<h1>Heading Title</h1>

</div>

**How could you use CSS to achieve the following automatic numbering:**

**1) Heading Title  
1.1) Subheading Title  
1.2) Subheading Title  
  
2) Heading Title  
2.1) Subheading Title  
  
3) Heading Title**

Hide answer

The following CSS will achieve this type of automatic numbering:

#page {

counter-reset: heading;

}

h1:before {

content: counter(heading)") ";

counter-increment: heading;

}

h1 {

counter-reset: subheading;

}

h2:before {

content: counter(heading)"." counter(subheading)") ";

counter-increment: subheading;

}

**Explain the CSS “box model” and the layout components that it consists of.**

**Provide some usage examples.**

Hide answer

The CSS box model is a rectangular layout paradigm for HTML elements that consists of the following:

* **Content** - The content of the box, where text and images appear
* **Padding** - A transparent area surrounding the content (i.e., the amount of space between the border and the content)
* **Border** - A border surrounding the padding (if any) and content
* **Margin** - A transparent area surrounding the border (i.e., the amount of space between the border and any neighboring elements)

Each of these properties can be specified independently for each side of the element (i.e., top, right, bottom, left) or fewer values can be specified to apply to multiple sides. For example:

/\* top right bottom left \*/

padding: 25px 50px 75px 100px;

/\* same padding on all 4 sides: \*/

padding: 25px;

/\* top/bottom padding 25px; right/left padding 50px \*/

padding: 25px 50px;

/\* top padding 25px; right/left padding 50px; bottom padding 75px \*/

padding: 25px 50px 75px;

*Q1*:

Describe floats and how they work

**Junior**

**[CSS](https://www.fullstack.cafe/interview-questions/css" \o "CSS Interview Questions)**[50](https://www.fullstack.cafe/interview-questions/css" \o "CSS Interview Questions)

Answer

*Float* is a CSS positioning property. Floated elements remain a part of the flow of the web page. This is distinctly different than page elements that use absolute positioning. Absolutely positioned page elements are removed from the flow of the webpage.

#sidebar {

float: right; // left right none inherit

}

The CSS clear property can be used to be positioned below left/right/both floated elements.

*Having Tech or Coding Interview?* Check 👉 [50 CSS Interview Questions](https://www.fullstack.cafe/interview-questions/css)

*Source:* [css-tricks.com](https://css-tricks.com/all-about-floats/)

*Q2*:

Have you played around with the new CSS Flexbox or Grid specs?

**Junior**

**[CSS](https://www.fullstack.cafe/interview-questions/css" \o "CSS Interview Questions)**[50](https://www.fullstack.cafe/interview-questions/css" \o "CSS Interview Questions)

Answer

Yes. Flexbox is mainly meant for 1-dimensional layouts while Grid is meant for 2-dimensional layouts.

Flexbox solves many common problems in CSS, such as vertical centering of elements within a container, sticky footer, etc. Bootstrap and Bulma are based on Flexbox, and it is probably the recommended way to create layouts these days. Have tried Flexbox before but ran into some browser incompatibility issues (Safari) in using flex-grow, and I had to rewrite my code using inline-blocks and math to calculate the widths in percentages, it wasn't a nice experience.

Grid is by far the most intuitive approach for creating grid-based layouts (it better be!) but browser support is not wide at the moment.

*Having Tech or Coding Interview?* Check 👉 [50 CSS Interview Questions](https://www.fullstack.cafe/interview-questions/css)

*Source:* [codeburst.io](https://codeburst.io/clearing-your-front-end-job-interview-css-95bdd82871f2)

*Q3*:

Describe pseudo-elements and discuss what they are used for.

**Mid**

**[CSS](https://www.fullstack.cafe/interview-questions/css" \o "CSS Interview Questions)**[50](https://www.fullstack.cafe/interview-questions/css" \o "CSS Interview Questions)

Answer

A *CSS pseudo-element* is a keyword added to a selector that lets you style a specific part of the selected element(s). They can be used for decoration (:first-line, :first-letter) or adding elements to the markup (combined with content: ...) without having to modify the markup (:before, :after).

Example of usage:

* :first-line and :first-letter can be used to decorate text.
* Used in the .clearfix hack to add a zero-space element with clear: both.
* Triangular arrows in tooltips use :before and :after. Encourages separation of concerns because the triangle is considered part of styling and not really the DOM, but not really possible to draw a triangle with just CSS styles.

*Having Tech or Coding Interview?* Check 👉 [50 CSS Interview Questions](https://www.fullstack.cafe/interview-questions/css)

*Source:* [codeburst.io](https://codeburst.io/clearing-your-front-end-job-interview-css-95bdd82871f2)

*Q4*:

Explain the usage of "table-layout" property

**Mid**

**[CSS](https://www.fullstack.cafe/interview-questions/css" \o "CSS Interview Questions)**[50](https://www.fullstack.cafe/interview-questions/css" \o "CSS Interview Questions)

Answer

The table-layout property defines the algorithm used to lay out table cells, rows, and columns.

table-layout: auto|fixed|initial|inherit;

* **auto** - Browsers use an automatic table layout algorithm. The column width is set by the widest unbreakable content in the cells. The content will dictate the layout.
* **fixed** - The layout is fixed based on the first row. Set the width of those, and the rest of the table follows. If no widths are present on the first row, the column widths are divided equally across the table, regardless of content inside the cells.
* **initial** - Sets this property to its default value.
* **inherit** - Inherits this property from its parent element.

*Having Tech or Coding Interview?* Check 👉 [50 CSS Interview Questions](https://www.fullstack.cafe/interview-questions/css)

*Source:* [w3schools.com](https://www.w3schools.com/cssref/pr_tab_table-layout.asp)

*Q5*:

How does CSS actually work (under the hood of browser)?

**Mid**

**[CSS](https://www.fullstack.cafe/interview-questions/css" \o "CSS Interview Questions)**[50](https://www.fullstack.cafe/interview-questions/css" \o "CSS Interview Questions)

Answer

When a browser displays a document, it must combine the document's content with its style information. It processes the document in two stages:

* The browser converts *HTML* and *CSS* into the *DOM (Document Object Model)*. The DOM represents the document in the computer's memory. It combines the document's content with its style.
* The browser displays the contents of the DOM.

*Having Tech or Coding Interview?* Check 👉 [50 CSS Interview Questions](https://www.fullstack.cafe/interview-questions/css)

*Source:* [developer.mozilla.org](https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS/How_CSS_works)

[8 Steps to Increase Your Dev Resume Response Rate by 90%](https://www.fullstack.cafe/blog/full-stack-resume-template)

**[Career](https://www.fullstack.cafe/interview-questions/career" \o "Career Interview Questions)**[1](https://www.fullstack.cafe/interview-questions/career" \o "Career Interview Questions)

*Q6*:

How is responsive design different from adaptive design?

**Mid**

**[CSS](https://www.fullstack.cafe/interview-questions/css" \o "CSS Interview Questions)**[50](https://www.fullstack.cafe/interview-questions/css" \o "CSS Interview Questions)

Answer

Both *responsive* and *adaptive* design attempt to optimize the user experience across different devices, adjusting for different viewport sizes, resolutions, usage contexts, control mechanisms, and so on.

**Responsive design** works on the principle of flexibility — a single fluid website that can look good on any device. Responsive websites use *media queries*, *flexible grids*, and *responsive images* to create a user experience that flexes and changes based on a multitude of factors. Like a single ball growing or shrinking to fit through several different hoops.

**Adaptive design** is more like the modern definition of progressive enhancement. Instead of one flexible design, adaptive design detects the device and other features, and then provides the appropriate feature and layout based on a *predefined set of viewport sizes* and other characteristics. The site detects the type of device used, and delivers the pre-set layout for that device. Instead of a single ball going through several different-sized hoops, you’d have several different balls to use depending on the hoop size.

*Having Tech or Coding Interview?* Check 👉 [50 CSS Interview Questions](https://www.fullstack.cafe/interview-questions/css)

*Source:* [codeburst.io](https://codeburst.io/clearing-your-front-end-job-interview-css-95bdd82871f2)

*Q7*:

What does Accessibility (a11y) mean?

**Mid**

**[CSS](https://www.fullstack.cafe/interview-questions/css" \o "CSS Interview Questions)**[50](https://www.fullstack.cafe/interview-questions/css" \o "CSS Interview Questions)

Answer

**Accessibility (a11y)** is a measure of a computer system's accessibility is to all people, including those with disabilities or impairments. It concerns both software and hardware and how they are configured in order to enable a disabled or impaired person to use that computer system successfully.

Accessibility is also known as *assistive technology*.

*Having Tech or Coding Interview?* Check 👉 [50 CSS Interview Questions](https://www.fullstack.cafe/interview-questions/css)

*Source:* [techopedia.com](https://www.techopedia.com/definition/10165/accessibility-a11y)

*Q8*:

What is a Mixin and how to use on?

**Mid**

**[CSS](https://www.fullstack.cafe/interview-questions/css" \o "CSS Interview Questions)**[50](https://www.fullstack.cafe/interview-questions/css" \o "CSS Interview Questions)

Answer

A **Mixin** is a block of code that lets us group CSS declarations we may reuse throughout our site.

To define mixin:

@mixin grid($flex: true /\*default argument\*/) {

@if $flex {

@include flex;

} @else {

display: block;

}

}

To use a Mixin, we simply use @include followed by the name of the Mixin and a semi-colon.

/\*scss\*/

.row {

@include grid(true);

}

/\*css\*/

.row {

display: -webkit-flex;

display: flex;

}

*Having Tech or Coding Interview?* Check 👉 [50 CSS Interview Questions](https://www.fullstack.cafe/interview-questions/css)

*Source:* [scotch.io](https://scotch.io/tutorials/how-to-use-sass-mixins)

 Full-Stack, Web & Mobile

 Coding & Data Structures

 System Design & Architecture

**[Objective-C](https://www.fullstack.cafe/interview-questions/objective-c" \o "Objective-C Interview Questions)**[43](https://www.fullstack.cafe/interview-questions/objective-c" \o "Objective-C Interview Questions)

**[Ionic](https://www.fullstack.cafe/interview-questions/ionic" \o "Ionic Interview Questions)**[29](https://www.fullstack.cafe/interview-questions/ionic" \o "Ionic Interview Questions)

**[CSS](https://www.fullstack.cafe/interview-questions/css" \o "CSS Interview Questions)**[50](https://www.fullstack.cafe/interview-questions/css" \o "CSS Interview Questions)

**[LINQ](https://www.fullstack.cafe/interview-questions/linq" \o "LINQ Interview Questions)**[38](https://www.fullstack.cafe/interview-questions/linq" \o "LINQ Interview Questions)

**[Android](https://www.fullstack.cafe/interview-questions/android" \o "Android Interview Questions)**[113](https://www.fullstack.cafe/interview-questions/android" \o "Android Interview Questions)

**[React](https://www.fullstack.cafe/interview-questions/react" \o "React Interview Questions)**[155](https://www.fullstack.cafe/interview-questions/react" \o "React Interview Questions)

**[Redis](https://www.fullstack.cafe/interview-questions/redis" \o "Redis Interview Questions)**[25](https://www.fullstack.cafe/interview-questions/redis" \o "Redis Interview Questions)

**[OOP](https://www.fullstack.cafe/interview-questions/oop" \o "OOP Interview Questions)**[54](https://www.fullstack.cafe/interview-questions/oop" \o "OOP Interview Questions)

**[ADO.NET](https://www.fullstack.cafe/interview-questions/adonet" \o "ADO.NET Interview Questions)**[33](https://www.fullstack.cafe/interview-questions/adonet" \o "ADO.NET Interview Questions)

**[MySQL](https://www.fullstack.cafe/interview-questions/mysql" \o "MySQL Interview Questions)**[55](https://www.fullstack.cafe/interview-questions/mysql" \o "MySQL Interview Questions)

*Q9*:

What's the difference between SCSS and Sass?

**Mid**

**[CSS](https://www.fullstack.cafe/interview-questions/css" \o "CSS Interview Questions)**[50](https://www.fullstack.cafe/interview-questions/css" \o "CSS Interview Questions)

Answer

There are two syntaxes available for Sass. The first, known as **SCSS (Sassy CSS)** and used throughout this reference, is an extension of the syntax of CSS. This means that every valid CSS stylesheet is a valid SCSS file with the same meaning. This syntax is enhanced with the Sass features described below. Files using this syntax have the .scss extension.

The second and older syntax, known as the indented syntax (or sometimes just **Sass**), provides a more concise way of writing CSS. It uses indentation rather than brackets to indicate nesting of selectors, and newlines rather than semicolons to separate properties. Files using this syntax have the .sass extension.

Consider **example.sass**:

$color: red

=my-border($color)

border: 1px solid $color

body

background: $color

+my-border(green)

Consider **example.scss**:

$color: red;

@mixin my-border($color) {

border: 1px solid $color;

}

body {

background: $color;

@include my-border(green);

}

*Having Tech or Coding Interview?* Check 👉 [50 CSS Interview Questions](https://www.fullstack.cafe/interview-questions/css)

*Source:* [stackoverflow.com](https://stackoverflow.com/questions/5654447/whats-the-difference-between-scss-and-sass)



[10 Unit Testing Interview Questions (CLARIFIED) You'll Be Asked On Next Tech Interview](https://www.fullstack.cafe/blog/unit-testing-interview-questions)

**[Software Testing](https://www.fullstack.cafe/interview-questions/software-testing" \o "Software Testing Interview Questions)**[26](https://www.fullstack.cafe/interview-questions/software-testing" \o "Software Testing Interview Questions)

*Q10*:

What's the difference between a relative, fixed, absolute and statically positioned element?

**Mid**

**[CSS](https://www.fullstack.cafe/interview-questions/css" \o "CSS Interview Questions)**[50](https://www.fullstack.cafe/interview-questions/css" \o "CSS Interview Questions)

Answer

A positioned element is an element whose computed position property is either relative, absolute, fixed or sticky.

* static - The default position; the element will flow into the page as it normally would. The top, right, bottom, left and z-index properties do not apply.
* relative - The element's position is adjusted relative to itself, without changing layout (and thus leaving a gap for the element where it would have been had it not been positioned).
* absolute - The element is removed from the flow of the page and positioned at a specified position relative to its closest positioned ancestor if any, or otherwise relative to the initial containing block. Absolutely positioned boxes can have margins, and they do not collapse with any other margins. These elements do not affect the position of other elements.
* fixed - The element is removed from the flow of the page and positioned at a specified position relative to the viewport and doesn't move when scrolled.
* 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.

1. Explain the three main ways to apply CSS styles to a web page.

You will, of course, need to know how to add CSS to a page, and there are three main ways:

**Using the inline style attribute on an element**

<div>

<p style="color: maroon;"></p>

</div>

**Using a <style> block in the <head> section of your HTML**

<head>

<title>CSS Refresher</title>

<style>

body {

font-family: sans-serif;

font-size: 1.2em;

}

</style>

</head>

**Loading an external CSS file using the <link> tag**

<head>

<title>CSS Refresher</title>

<link rel="stylesheet" href="/css/styles.css" />

</head>

The first two are useful but you’ll almost always be loading external CSS files. It’s more maintainable to keep your styles in separate files, not to mention it’s a nice separation of concerns.

2. Could you describe the different kinds of selectors?

A selector in CSS is the bit before the curly bracket, for example, body or .nav-item, that selects the content you want to style. The different kinds of selectors include tags, classes & IDs:

* A tag references an [HTML](https://www.goskills.com/Course/Intro-HTML/) tag
* A class references the class attribute on an HTML tag
* Likewise, an ID references the id attribute on an HTML tag

<!-- tag -->

<section>

<!-- class -->

<div class="button-cta">

<!-- id -->

<a href="cta-click">Click me!</a>

</div>

</section>

3. Explain the three main ways to target elements.

Now, in order to style certain elements on the page, you need to know how to specify those elements. There are three main ways to target something:

* By tag, e.g. you can target the body or p or span
* By the class you’ve specified in your HTML. For example, if you have three list items with the class nav-item you can target them with .nav-item
* By the ID you’ve specified in your HTML. For example, if you have a link with the ID home-button you can target it with #home-button

/\* tag \*/

body { ... }

/\* class \*/

.nav-item { ... }

/\* id \*/

#home-button { ... }

Keep in mind that IDs should be unique, meaning that there should never be more than one item with a particular ID on a page. It’s also generally better to target tags for very generic, site-wide styles, and classes for more specific styles. You can use a unique class or modifier class instead of an ID if you really need to target a single item.

If you want to target something with two or more classes or IDs, simply concatenate them all together:

.button.nav { ... }

4. How do you target something inside or around another element?

Sometimes you need to target something that’s next to something else or target something only if it’s inside a particular container. Luckily, that’s easy!

If you’re wanting to style occurrences of a certain class inside another class, you can write the following:

.nav .nav-item { ... }

The above targets any .nav-item inside .nav. If you only want those that are immediately inside the .nav as opposed to any level deep, you can add the > character, like so:

.nav > .nav-item { ... }

Want to target a button only if it’s next to another button? The + character has got you covered:

.button + .button { ... }

5. What are pseudo elements and what are they used for?

Pseudo elements are used to style particular parts of an element, rather than the whole thing. For example, you can use it to style the first line or first letter of a paragraph, text you’ve selected, or you can use it to insert text or shapes before or after an element.

They always start with a double colon - although a single colon is still allowed for backwards compatibility - and they look like this:

p::first-line { ... }

span::first-letter { ... }

::selection { ... }

.header::after { ... }

.tooltip::before { ... }

6. What are pseudo classes and what are they used for?

Pseudo classes are similar to pseudo elements, but instead of styling a part of an element, they apply styles when an element is in a certain state. For example, you could style a button differently based on whether the user has their mouse pointer over it, or when they click the button.

Another common use case is to style only certain occurrences of elements in a row. For example, styling the first tab in a series of tabs, or every second tab.

They all start with a single colon and look like this:

.link:hover { ... }

.link:active { ... }

.tab:first-child { ... }

.tab:last-child { ... }

.avatar:nth-child(2n) { ... }

7. What are attributes and how are they used?

You already know about classes, which means you already know about attributes, seeing as a class is just one of the many attributes that HTML tags can have. There are a few 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 tags can have href.

You can target elements with particular attributes by using square brackets: [attribute="value"]. For example, you can target all input fields that are of type radio like so:

input[type="radio"] {

background-color: #eee;

}

Specificity

8. Can you describe the rules around specificity?

If you’ve spent any time wrangling CSS, you’ve likely come across the tricky rules around specificity. What that means is which rules override others simply by being more specific.

The following list displays the [order of specificity](https://developer.mozilla.org/en-US/docs/Web/CSS/Specificity), from low to high:

1. Type selectors (e.g. h1) and pseudo-elements (e.g. ::before)
2. Class selectors (e.g. .nav-item), attributes selectors (e.g. [type="radio"]) and pseudo-classes (e.g. :hover)
3. ID selectors (e.g. #example)

This is the reason it’s unwise to use IDs to target specific elements, as they are of a higher specificity. Using it in too many places can backfire later on, where you end up trying to override those styles, which can be hard to do.

Also, the more selectors you use, the higher it ranks in terms of specificity, e.g. button.primary[target="\_blank"] is more specific than simply button.

/\* low specificity \*/

button { ... }

/\* higher specificity \*/

button.primary[target="\_blank"] { ... }

9. Should you use **!important**?

Ahh, no way of escaping this one. When you’re wrestling with specificity, trying to override that one style, you may be tempted to add **!important** to your style. Think very carefully before you do so! It may seem harmless, but if you do this all over the place, you will soon find yourself in a big mess that you cannot get out of.

What **!important** does, is make that particular style have the highest specificity possible. Nothing else can override it, apart from another **!important**. You might be able to see where this is going, but once you start using them, you’ll almost certainly get into a position where you need to override a rule marked as **!important**, which forces you to use another one. And so the cycle continues.

/\* high specificity \*/

.large .navigation.navigation-large {

font-size: 2em;

}

/\* will override the above, but it's dangerous! \*/

.navigation {

font-size: 3em **!important**;

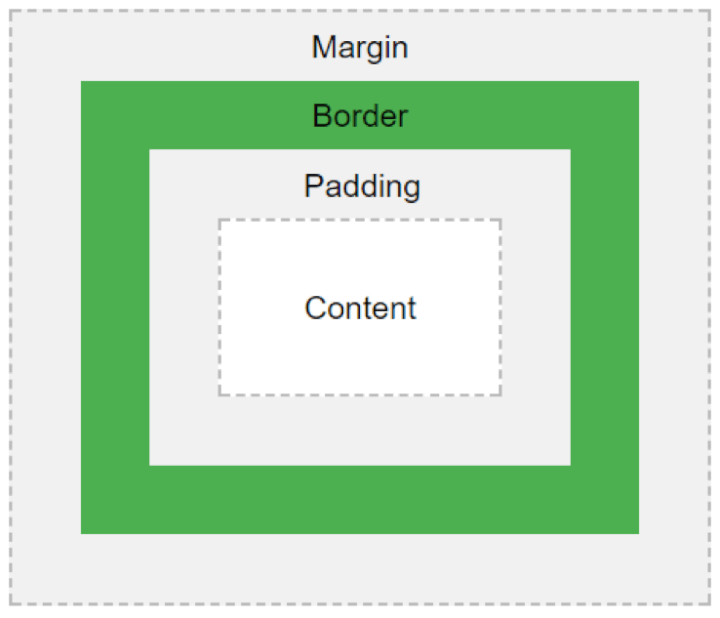
}

The CSS Box Model

10. How do margin, border and padding fit together in the box model?

While the border is pretty self-explanatory, it can be somewhat confusing to see the difference between margin and padding. Surely both simply add space around the element you’re styling?

The easiest way to understand is to look at a visual representation of it. Looking at the below image, the very centre rectangle is the size of your element. Immediately surrounding that is padding. Then comes the border, and only then comes the margin.

CSS Box Model

Padding adds bulk to your element, in between the border and the element. This means that if your element has a background color, that color will also fill the padding.

Margin adds empty space around your element. That means that the aforementioned background color will not fill the margin.

Interested in web development?

Learn all about it in our comprehensive (and free) ebook!

Top of Form

ENTER YOUR EMAIL ADDRESS



Bottom of Form

Fonts

11. What’s your preferred way of sizing fonts?

Something that has almost always come up for me is the way you size your text, mainly focused on the units you use. You can of course use pixels (px), but there’s also em, rem, %, vs and vh, along with a few others. Some people still don’t like using pixels, but browsers have improved and they’re generally handled pretty well.

Defining your font sizes in em allows you to change the size of your text based on the size defined at a higher level. For example, if a container has specified a font-size of 2em, and you specify a font-size of 2em on an element inside that container, that element has an effective font-size of 4em! However, this can be a little confusing as you might not always see the size you expect!

.container {

font-size: 2em;

}

.container > p {

font-size: 2em; /\* this is 2em x 2em = 4em! \*/

}

The rem unit was created to remedy that confusion. It scales well in the browser, just like em and px, but it uses a base size. From that, all further rem values are calculated. For example, if your base rem value is equal to 16px, then 1rem will always be equal to 16px, 2rem will always be equal to 32px, and so on.

Note: While I’ve explained these units using font-sizes, the same rules apply to any dimensions where you use px, em or rem.

12. What are web safe fonts and fallback fonts?

Not all operating systems and browsers have the same fonts installed. Web safe fonts are fonts that are commonly pre-installed on many computer systems, such as Arial and Times New Roman. In case the browser or operating system doesn’t recognize the first font you set (e.g. Ubuntu), you should choose a web safe fallback font to display (e.g. Arial), followed by a generic font family (e.g. sans-serif). If your fallback font doesn’t display either, the browser can pick a generic font in the sans-serif family.

Media queries

13. How would you use media queries in a mobile-first approach?

There’s no way to avoid these nowadays, everyone expects their website to work on mobile devices, even if they don’t specifically ask for it.

The most common approach is the mobile-first one. All styles outside of [media queries](https://www.goskills.com/Course/Intro-CSS/Lesson/1893/Media-Queries) are targeted at mobile devices. Then, through progressively larger media queries, you can style larger screens one step at a time.

/\* mobile styles \*/

body {

font-size: 1em;

}

/\* desktop styles \*/

@media only screen and (min-width: 768px) {

body {

font-size: 1.5em;

}

}

Browsers

14. How do you handle browser differences in your user base?

In the past few years, browsers have come a long way. Modern browsers update automatically and provide pretty good feature coverage. There are still differences between browsers, and while some support certain features (like certain CSS styles), others don’t or display them differently.

Your biggest concern is legacy browsers. You may have seen jokes about Internet Explorer, and while the newer versions aren’t as bad, there are still a lot of people using older versions. These days it’s not uncommon to still support IE9, sometimes even IE8.

This can severely limit the kind of styles you can use, though most can be approximated using fallback styles. The @supports query in CSS can be very useful to scan if the user’s current browser has a certain feature. If they do, you can apply those styles without worry. If not, you can have some fallback styles as a backup.

For more information, have a look at the [Mozilla documentation](https://developer.mozilla.org/en-US/docs/Web/CSS/@supports) about @supports.

15. Do you use any tools for browser support?

One of my favourite online tools is [caniuse.com](https://caniuse.com/). It’s a website that tells you exactly which browsers support which features, including CSS and JavaScript, which can be extremely helpful in finding out what you can and can’t use!

16. Have you used Flexbox & CSS Grid before? What are the differences between them?

Recently we’ve seen the rise of Flexbox and even more recently, CSS Grid. While these aren’t well supported in older browsers, support for them in newer browsers is continuously growing and now is the time to look into them and see if you can start using them. Make sure you check browser support!

**Flexbox** is a very useful layout tool, especially for smaller areas within the site. Its main features are to align items in horizontal or vertical axes, space them out automatically, invert the order in which they’re displayed, along with a few other layout options.

**CSS Grid** is more of a layout tool for the entire page. While Flexbox excels in laying out items along a single axis, Grid is better for layouts with both horizontal and vertical axes, i.e. grids!

CSS preprocessors

17. Do you use any CSS preprocessors, and which do you prefer?

If you’re working on a medium to large project, it’d be a good idea to use a CSS preprocessor. They allow you to write more concise CSS, split it up into multiple files and use a large number of very useful functions and mixins (you can even create your own!), along with variables.

The main players are Sass (also referred to as SCSS), LESS and Stylus, although arguably Sass is the biggest. I’ll be using Sass in the following examples.

18. What is file splitting and why should you use it?

File splitting helps organize your CSS into multiple files, decreasing page load time and making things easier to manage. If you’re working with any of the preprocessors above, you can start splitting up your files.

How you decide to split them up is up to you, but it can be useful to separate files by component. For example, you can have all your button styles in a file called \_buttons.scss or all your header-specific styles in a file called \_header.scss. Then, in your main file, say \_app.scss, you can import those files by writing @import 'buttons';

This way you can also create separate stylesheets for separate areas of your website, where you might not need all styles. For example, if you have a web app, you probably don’t need to load all styles when people land on your homepage, when they haven’t even logged into your app yet. Simply create another file and import only those styles you need.

19. What are variables used for?

Variables are super useful for things like colors, fonts, font sizes, and certain dimensions, as you can be sure you’re always using the same ones, not 4 different versions of roughly the same color.

$primary-font-stack: 'Helvetica', sans-serif;

$primary-color: #fccd48;

body {

color: $primary-color;

font-family: $primary-font-stack;

}

20. What are functions/mixins?

Mixins are a very handy way of adding a number of styles, based on a particular input parameter. For example, you might always want to add fallback styles when adding border-radius, but you don’t necessarily know what value you might want.

@mixin border-radius($radius) {

-webkit-border-radius: $radius;

-moz-border-radius: $radius;

-ms-border-radius: $radius;

border-radius: $radius;

}

.box {

@include border-radius(10px);

}

If you’re in the position of needing to interview someone about their skill and knowledge about CSS, it can be a little hard to think of things to ask on-the-fly. I thought I’d think up and round up some ideas for reference.



**Exercises To Do**

Seeing people’s actual work is just as important as what they say. Seeing people work their through exercises live might be even more important. These are some exercises that aren’t particularly difficult and anybody with CSS experience should be able to do. Watching/listening to them do it could be invaluable. [Collab Mode](https://blog.codepen.io/documentation/pro-features/collab-mode/) on CodePen is kinda perfect for that (just saying).

**Create This Button**

I saw [this idea](https://codepen.io/mobify/pen/GtqKj) on [Mobify’s CodePen account](https://codepen.io/mobify). Give people an image of a button, and tell them:

*Using CSS properties alone, recreate this button:*



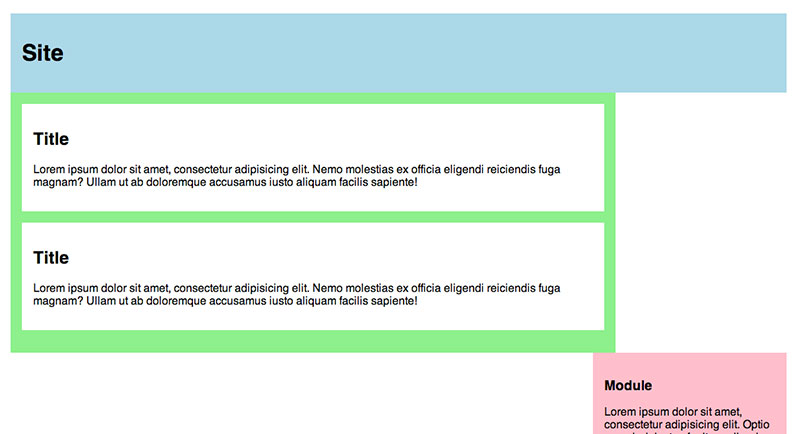
This would be a great test of the candidate’s CSS3 skills. This button is loaded with that kind of stuff. Things I would look for:

* How did they handle [multiple borders](https://css-tricks.com/snippets/css/multiple-borders/)? box-shadow is probably the best way here.
* Did they use text-transform: uppercase; for the text? That would be best.
* How did they handle the stars? Pseudo elements would be a good candidate there. Did they use unicode? Icon font?
* Did they catch the subtle text-shadow?
* How did they split the background in the main part of the button? a linear-gradient with no fade would be a good technique there. Or box-shadow.
* Were they [careful](https://css-tricks.com/public-service-announcement-careful-with-your-nested-border-radii/) with the nested border-radius?

I hope I didn’t ruin Mobify’s hiring process! But honestly, I think this stuff is hard to fake. You can either do it and talk about it, or you can’t.

**Fix The Sidebar**

The right sidebar here has fallen down below the content. [Show me](https://codepen.io/chriscoyier/pen/ClGcF) some different ways you could fix that.



Even though the two columns are 75% and 25% wide and floated opposite ways, the sidebar has fallen. The reason is because the columns don’t actually add up to 100% – they add up to more than that due to the padding. There is a number of ways to fix it:

* Using box-sizing: border-box; on the appropriate elements is the most efficient fix.
* Using calc() on the widths to remove the 1rem padding is another way.
* Putting the padding on an added internal wrapper element instead of the columns is a fix that will work with very deep browser support.
* Adjusting the numbers to make the math work is another way. For instance making the widths of the columns 4% narrower and using 2% for the padding instead.

There are other ways. The more solutions they can think of, the more creative and versatile a problem solver they are.

**Make this fixed width design fluid.**

[Here’s a design.](https://codepen.io/chriscoyier/pen/lDJmf) It’s a fixed 800px wide. Do your best at making it fit the screen more appropriately at any screen size.

This is just one ingredient to responsive design (which they should probably at least be familiar with) but it’s an important one. It can help prove they can think spatially and make reasoned choices about layout. I’d look for:

* Changing the pixel widths to percentages (how did they handle the math?)
* Did they do anything special for large screens or just small?
* Did they attempt to use a responsive images solution?
* Does the new design retain the hierarchy of importance inherit to the original?
* Did they come back at you with questions? (Lots of things to ask here, including what other resources might be available.)
* Did they test it? (To make sure it actually works, and find things like missing meta tags.)

**Replace this logo markup with an image.**

<a href="/" class="logo">Company</a>

Accessible and semantic image replacement has been a CSS topic for a lot of years and the “best way” has [morphed over the years](https://css-tricks.com/examples/ImageReplacement/). Asking them to show you how it could be done a number of ways would be a way to get insight into how long they’ve really been working with CSS. Not only is it directly an important thing to know how to do, knowing how to do it multiple ways demonstrates the depth of their internal toolbox.

**Google how you would find out what the default value for backface-visibility is.**

Being able to Google something quickly and efficiently is a huge part of any developers job. Are they deft at it? Did they find the right answer? Did they go to a specific trusted source in the results page?

Perhaps phrase the question without “Google” and see which search engine they use. No particular bias here, but if it’s not Google, are they as efficient as you would expect them to be with whatever other search engine that is?

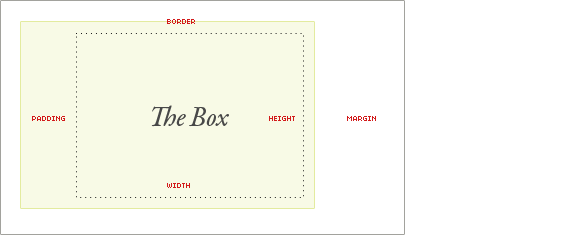
**Questions to Ask**

Darcy Clarke [got the ball rolling](http://www.darcyclarke.me/articles/news/front-end-job-interview-questions/) with this a few years ago. I’m going to update some of those, and add some of my own, and elaborate a bit.

**What is the “Box Model” in CSS? Which CSS properties are a part of it?**

The CSS [box model](https://css-tricks.com/the-css-box-model/) is fundamental to understanding layout and sizing and such. It’s made up of:

* Width and height (or in the absence of that, default values and the content inside)
* Padding
* Border



Margin is related but not technically a part of it. I would give extra points for knowing/mentioning that, but not take away and points for including it.

**What are Sass, Less, and Stylus? Why do people use them? How does something like Compass relate to Sass?**

They are CSS preprocessors. They are an abstraction layer on top of CSS. They are a special syntax/language that compile down into CSS. They make managing CSS easier, with things like variables and mixins to handle vendor prefixes (among other things). They make doing best practices easier, like concatenating and compressing CSS.

Bonus points for knowing how they differ and/or having experience using them. More bonus points for knowing what things like Compass, Bourbon, LESSHat, Nib, etc are and how they relate.

**Name some online resources that you reference when having CSS issues.**

Being good at googling problems you are having is a valuable job skill. There isn’t any shame in it. There is shame in spinning your wheels because “you should know this.” If you don’t have time to do the Googling exercise above, just asking about resources can be telling.

Google is a pretty good answer (since it’s true and we all know it). But being able to name some specific sites is a good indicator they have done it a bunch and are familiar with the places they land and know their favorites. Stuff like MDN (Mozilla Developer Network) is a good answer.

**Describe what a “reset” CSS file does and how it’s useful. Are you familiar with normalize.css? Do you understand how they differ?**

Resets are so wildly common in CSS that anyone who is a front end developer type has surely used them. Do they do so blindly or do they know why? The reason is essentially that different browsers have different default styling for elements, and if you don’t deal with that at all, you risk designers looking unnecessarily different in different browsers and possibly more dramatic breaking problems.

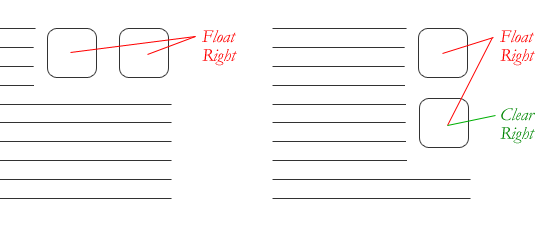
[Normalize](https://necolas.github.io/normalize.css/) you might call a CSS reset alternative. Instead of wiping out all styles, it delivers a set of reasonable defaults. It doesn’t unset things that are already consistent across browsers and reasonable (e.g. bold headers). In that way it does some less than a reset. It also does some more than a reset in that it handles quirks you may never consider, like HTML5 audio element inconsistencies or line-height inconsistencies when you use sub and sup elements.

**What are the various techniques for clearing floats?**

[Floats](https://css-tricks.com/all-about-floats/) are still incredibly common. As this is published, still probably the most cross-browser consistent way to build layout and grids. Anyone who has worked with them is aware of float collapsing. That is, floated element do not add to the height of a parent element. So for example if a parent element contained only floated elements, it would collapse to zero height. You can deal with that like:

* Use a clearfix (bonus points for micro clearfix).
* Float the parent as well.
* Use an overflow property other than “visible” on the parent (bonus points for listing downsides like cutting off shadows).

Bonus points for “create a [new block formatting context](http://www.yuiblog.com/blog/2010/05/19/css-101-block-formatting-contexts/)“. Possibly negative points for something like <br style="clear: both;">

Do they understand directional clearing?

As a bonus question, you could ask them to compare using inline-block and floats for building a grid. Good answer: there are problems either way. With inline-block you need to deal with the whitespace issue. With floats you need to deal with clearing.

**What are sprites and why would use them? How do you go about creating them? What are possible alternatives to sprites?**

[Sprites are](https://css-tricks.com/css-sprites/) essentially multiple images combined into one. Performance is the reason that they are used. Generally speaking, the slowest thing a website can do is request a resource. The fewer requests a site needs to make, the faster it is. Fast = good. Combining what would be many requests into one = good.



Asking how they make sprites would just seal the deal that they are actually very familiar with them. Manually creating sprites is certainly a possibility but it isn’t very efficent. There are helper tools like [SpriteCow](https://css-tricks.com/video-screencasts/105-using-spritecow/) and [SpriteMe](http://spriteme.org/), [Spriting with Compass](http://compass-style.org/help/tutorials/spriting/), or [Grunticon](https://github.com/filamentgroup/grunticon). It’s always interesting to hear a real workflow approach.

Sprites are raster images. When asking about alternatives, good answers might be related to the fact that sprites are often for icons and icons often don’t need to be raster. [SVG Stacks](http://simurai.com/post/20251013889/svg-stacks), [Icon Fonts](https://css-tricks.com/flat-icons-icon-fonts/), [Unicode](http://copypastecharacter.com/)…

**What are some accessibility concerns that come up in CSS?**

[Hidden content](https://css-tricks.com/places-its-tempting-to-use-display-none-but-dont/) is a big one here. It’s only acceptable to use display: none; if you’re both trying to hide things visually and the content itself.

CSS controls colors so color [accessibility](http://colorfilter.wickline.org/) is relevant. [Focus styles](http://a11yproject.com/posts/never-remove-css-outlines/) are also important and directly controlled by CSS.

There is a lot more accessibility stuff that is related to HTML and JavaScript, so mentioning that stuff is great, but I think it’s interesting to target the question solely on CSS to force focused thinking.

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

Bonus points for bringing up specific details like the fact that you can’t transform inline elements.

**What tools do you use for cross-browser testing?**

They should have some kind of strategy. Perhaps a web-based tool like [BrowserStack](http://www.browserstack.com/). Perhaps a VM based tool like [Virtual Box](https://www.virtualbox.org/). Perhaps different actual computers.

Part of the job of front end design is making sure things work everywhere they can (based on decided-upon support). You don’t have to love it, but you can’t hate it. [*“This right here, this is the job. What kind of work were you expecting?”*](http://the-pastry-box-project.net/wilson-miner/2013-june-15/)

**What are some of your favorite web design workflow tools?**

What code editor to they like? Where do they go for inspiration? What experience with version control do they have? What was QA like where they have worked? Support? What different deployment methods have they worked with? Do they know Photoshop or another visual design software alternative? Are the comfortable with the terminal?

Those are just some examples, it’s interesting to hear about any software they use to get the jobs done. Getting a sense of the tools they use (and better, the tools they *like*) is interesting. Bonus points: a sense of excitement about some of the tools.

**Say you found a rendering problem on one of your sites in Internet Explorer 8, which you have decided you are supporting. How would you approach fixing it?**

This would be an alternative question to asking generally about cross browser testing. More specific. Perhaps a more difficult alternative question would be to replace IE 8 with something like “A Google Nexus running Android 2.3.” Would they find a simulator? Would they seek out a design lab? Would they ask the company for an allowance for devices for testing, with some kind of smart plan? Would they find a friend with one?

**What is responsive design all about?**

It’s about making websites work wherever the web is. Different devices with different sizes and different capabilities. Responsive design is about taking one code base and making it work for all of them. Part of that is media queries and different visuals. Part of that is different resources (e.g. different JavaScript to handle touch vs click or different images to accommodate the screen).

**Have you ever worked with a grid layout? What are your thoughts on that?**

Why did they need a grid? How did they build the grid? Was it home grown or did they use a grid tool? Did they like the grid tool? What kind of class names did they use? Did they go mobile-first or desktop-first? Was a help or a hinderance? Do they reach for the grid automatically on any project?

**What are the benefits of SVG?**

[SVG](https://css-tricks.com/using-svg/) is an image format that is vector based. It’s an efficient format for that (small file sizes). You can scale them and they retain their sharpness at any size (bonus points for mentioning raster might have the upper hand at tiny sizes). You can affect parts of them with CSS and JavaScript as well as SVG specific filters that can do things like blurring.

**Have you ever created a print stylesheet for a website?**

Kind of indicative that they’ve “gone the extra mile” with websites before. What approach did they take? How did they test it?

**Say you were tasked with coding a design that used non-standard web fonts, how would you go about it?**

A non-leading way to get them to talk about @font-face and how it works. Talking about how it works as a core CSS technology is great, as well as talking about services that provide the fonts and can make it easier e.g. Google Fonts, Typekit, Font Deck, Cloud Typography, etc.

Bonus points for obscure knowledge like the history of @font-face syntax or [Firefox’s issue](https://www.change.org/petitions/mozilla-firefox-fix-bug-604421) with cross-origin fonts.

**Explain to me what’s going on in this CSS selector:**

[role=navigation] > ul a:not([href^=mailto]) {

}

This selects anchor links that are not email links that are decedents of an unordered list that is the direct child of any element with a role attribute of ‘navigation’.

Being able to verbalize a selector is proof you understand them and evidence they can communicate complex tech subjects.

1. **Question 1. What Is The Difference Between Id And Class?**

**Answer :**

ID identifies and sets style to one and only one occurrence of an element while class can be attached to any number of elements.

1. **Question 2. Can I Include Comments In My Style Sheet?**

**Answer :**

Yes. Comments can be written anywhere where whitespace is allowed and are treated as white space themselves. Anything written between /\* and \*/ is treated as a comment (white space).

**NOTE:** Comments cannot be nested.

[XML Interview Questions](https://www.wisdomjobs.com/e-university/xml-interview-questions.html)

1. **Question 3. Is There Anything That Can T Be Replaced By Style Sheets?**

**Answer :**

Quite a bit actually. Style sheets only specify information that controls display and rendering information. Virtual style elements that convey the NATURE of the content can not be replaced by style sheets, and hyperlinking and multimedia object insertion is not a part of style sheet functionality at all (although controlling how those objects appear IS part of style sheets functionality.) The CSS1specification has gone out of its way to absorb ALL of the HTML functionality used in controlling display and layout characteristics. For more information on the possible properties in CSS, see the Index DOT Css Property Index.

Rule of Thumb: if an HTML element or attribute gives cues as to how its contentsshould be displayed, then some or all of its functionality has been absorbed by stylesheets.

1. **Question 4. How Can I Make A Page Look The Same In E.g. Ns And Msie ?**

**Answer :**

The simple answer is, you can't, and you shouldn't waste your time trying to make it exactly the same. Web browsers are allowed, per definition, to interpret a page as they like, subject to the general rules set down in the HTML and CSS specifications. As a web author you can not have a prior knowledge of the exact situation and/or medium that will be used to render your page, and it's almost always rather counterproductive to try to control that process. There is no necessity for a well-written page to look the same in different browsers. You may want to strive to ensure that it looks good in more than one browser, even if the actual display (in the case of graphical browsers) comes out a bit different. "Looking good" can be achieved by adopting sensible design and guidelines, such as not fixing the size or face of your fonts, not fixing the width of tables, etc… Don't fight the medium; most web users only use one browser and will never know, or bother to find out, that your page looks different, or even "better", in any other browser.

[XML Tutorial](https://www.wisdomjobs.com/e-university/xml-tutorial-181.html)

1. **Question 5. What Are Pseudo-elements?**

**Answer :**

Pseudo-elements are fictional elements that do not exist in HTML. They address the element's sub-part (non-existent in HTML) and not the element itself. In CSS1 there are two pseudo-elements: 'first-line pseudo-element' and 'first-letter pseudo-element'. They can be attached to block-level elements (e.g. paragraphs or headings) to allow typographical styling of their sub-parts.

[CSS3 Interview Questions](https://www.wisdomjobs.com/e-university/css3-interview-questions.html)

1. **Question 6. How Do I Write My Style Sheet So That It Gracefully Cascades With User S Personal Sheet ?**

**Answer :**

You can help with this by setting properties in recommended places. Style rules that apply to the whole document should be set in the BODY element -- and only there. In this way, the user can easily modify document-wide style settings.

1. **Question 7. What Is Property?**

**Answer :**

Property is a stylistic parameter (attribute) that can be influenced through CSS, e.g. FONT or WIDTH. There must always be a corresponding value or values set to each property, e.g. font: bold or font: bold san-serif.

[CSS3 Tutorial](https://www.wisdomjobs.com/e-university/css3-tutorial-205.html) [HTML Interview Questions](https://www.wisdomjobs.com/e-university/html-interview-questions.html)

1. **Question 8. What Can Be Done With Style Sheets That Can Not Be Accomplished With Regular Html?**

**Answer :**

Many of the recent extensions to HTML have been tentative and somewhat crude attempts to control document layout. Style sheets go several steps beyond, and introduces complex border, margin and spacing control to most HTML elements. It also extends the capabilities introduced by most of the existing HTML browser extensions. Background colors or images can now be assigned to ANY HTMLelement instead of just the BODY element and borders can now be applied to anyelement instead of just to tables. For more information on the possible properties in CSS, see the Index DOT Css Property Index.

1. **Question 9. Why Is My External Stylesheet Not Working ?**

**Answer :**

There may be several different reasons behind that, but one very common mistake is to have an external stylesheet that contains HTML markup in some form.

An external stylesheet must contain only CSS rules, and if required, correctly formed CSS comments; never include any HTML syntax, such as <style type="text/css">…

CSS comments are defined as anything that is placed between

/\* (the comment start mark) and

\*/ (the comment end mark). I.e. as follows…

/\* This text right here is a correct CSS comment \*/

CSS comments may span multiple lines in the stylesheet. Nesting of CSS comments is not allowed.

Another reason for external stylesheets (and even embedded and inline stylerules) not to function as expected may be that you have tried to make use of some CSS-features that are not supported in the browser you are using.

External stylesheets shall also be served from the www-server with a MIME-type of 'text/css' in its 'Content Type:' HTTP header.

You may need to negotiate with your server admin to add this MIME type to your server if you are not able to configure the server yourself.

[Dreamweaver Interview Questions](https://www.wisdomjobs.com/e-university/dreamweaver-interview-questions.html)

1. **Question 10. How Do I Quote Font Names In Quoted Values Of The Style Attribute?**

**Answer :**

The attribute values can contain both single quotes and double quotes as long as they come in matching pairs. If two pair of quotes are required include single quotes in double ones or vice versa:

<P STYLE="font-family: 'New Times Roman'; font-size: 90%">  
<P STYLE='font-family: "New Times Roman"; font-size: 90%'>

It's been reported the latter method doesn't work very well in some browsers, therefore the first one should be used.

[HTML Tutorial](https://www.wisdomjobs.com/e-university/html-tutorial-206.html)

1. **Question 11. Styles Not Showing?**

**Answer :**

**There are different ways to apply CSS to a HTML document with a stylesheet, and these different ways can be combined:**

**\*** inline (internal) (Deprecated for XHTML)  
\* embedded (internal)  
\* linked (external) and  
\* @import (external)

**Note:** An external stylesheet is a text file that contains only CSS Styles. HTML comments are not supposed to be in there and can lead to misinterpretation (> is the CSS "Child" selector!).

[UI Developer Interview Questions](https://www.wisdomjobs.com/e-university/ui-developer-interview-questions.html)

1. **Question 12. Are Style Sheets Case Sensitive?**

**Answer :**

No. Style sheets are case insensitive. Whatever is case insensitive in HTML is also case insensitive in CSS. However, parts that are not under control of CSS like font family names and URLs can be case sensitive - IMAGE.gif and image.gif is not the same file.

[XML Interview Questions](https://www.wisdomjobs.com/e-university/xml-practice-tests-181-326158)

1. **Question 13. What Is Cascade?**

**Answer :**

Cascade is a method of defining the weight (importance) of individual styling rules thus allowing conflicting rules to be sorted out should such rules apply to the sameselector.

**Declarations with increased weight take precedence over declaration with normal weight:**

P {color: white ! important} /\* increased weight \*/  
P (color: black} /\* normal weight \*/

[CSS Tutorial](https://www.wisdomjobs.com/e-university/css-tutorial-1198.html)

1. **Question 14. What Is Important Declaration?**

**Answer :**

Important declaration is a declaration with increased weight. Declaration with increased weight will override declarations with normal weight. If both reader's and author's style sheet contain statements with important declarations the author's declaration will override the reader's.

BODY {background: white ! important; color: black}

In the example above the background property has increased weight while the color property has normal.

1. **Question 15. What Is Css Declaration?**

**Answer :**

CSS declaration is style attached to a specific selector. It consists of two parts; property which is equivalent of HTML attribute, e.g. text-indent: and value which is equivalent of HTML value, e.g. 10pt. NOTE: properties are always ended with a colon.

[CSS Interview Questions](https://www.wisdomjobs.com/e-university/css-interview-questions.html)

1. **Question 16. What Is Class Selector?**

**Answer :**

Class selector is a "stand alone" class to which a specific style is declared. Using the CLASS attribute the declared style can then be associated with any HTML element. The class selectors are created by a period followed by the class's name. The name can contain characters a-z, A-Z, digits 0-9, period, hyphen, escaped characters, Unicode characters 161-255, as well as any Unicode character as a numeric code, however, they cannot start with a dash or a digit. (Note: in HTML the value of the CLASS attribute can contain more characters).It is a good practice to name classes according to their function than their appearance.

.footnote {font: 70%} /\* class as selector \*/  
<ADDRESS CLASS=footnote/>This element is associated with the CLASS footnote</ADDRESS>  
<P CLASS=footnote>And so is this</P>

[CSS Advanced Tutorial](https://www.wisdomjobs.com/e-university/css-advanced-tutorial-1199.html)

1. **Question 17. What Is Selector?**

**Answer :**

CSS selector is equivalent of HTML element(s). It is a string identifying to which element(s) the corresponding declaration(s) will apply and as such the link between the HTML document and the style sheet.

For example in P {text-indent: 10pt} the selector is P and is called type selector as it matches all instances of this element type in the document.

in P, UL {text-indent: 10pt} the selector is P and UL (see grouping); in .class {text-indent: 10pt} the selector is .class (see class selector).

[WordPress Interview Questions](https://www.wisdomjobs.com/e-university/wordpress-interview-questions.html)

1. **Question 18. What Is Css Rule At-rule ?**

**Answer :**

**There are two types of CSS rules:**

ruleset and at-rule. At-rule is a rule that applies to the whole style sheet and not to a specific selector only (like in ruleset). They all begin with the @ symbol followed by a keyword made up of letters a-z, A-Z, digits 0-9, dashes and escaped characters, e.g. @import or @font-face.

[CSS3 Interview Questions](https://www.wisdomjobs.com/e-university/css3-interview-questions.html)

1. **Question 19. What Does The Cascading In Cascading Style Sheets Mean?**

**Answer :**

Style Sheets allow style information to be specified from many locations. Multiple (partial) external style sheets can be referenced to reduce redundancy, and both authors as well as readers can specify style preferences.

In addition, three main methods can be employed by an author to add style information to HTML documents, and multiple approaches for style control are available in each of these methods. In the end, style can be specified for a single element using any, or all, of these methods.

[WordPress Tutorial](https://www.wisdomjobs.com/e-university/wordpress-tutorial-1204.html)

1. **Question 20. Why Use Style Sheets?**

**Answer :**

Style sheets allow a much greater degree of layout and display control than has ever been possible thus far in HTML. The amount of format coding necessary to control display characteristics can be greatly reduced through the use of external style sheets which can be used by a group of documents. Also, multiple style sheetscan be integrated from different sources to form a cohesive tapestry of styles for a document. Style sheets are also backward compatible - They can be mixed with HTML styling elements and attributes so that older browsers can view content as intended.

[Pure.CSS Interview Questions](https://www.wisdomjobs.com/e-university/puredotcss-interview-questions.html)

1. **Question 21. How Do I Get Rid Of The Gap Under My Image?**

**Answer :**

Images are inline elements, which means they are treated in the same way as text. Most people kind of know this - they know that if you use 'text-align:center' on an image it will be centred. What many people don't realise is that this means you will have a gap underneath an image. This gap is for the descenders of letters like j,q,p,y and g. To get rid of this gap you need to make the image block-level - like this :

CSS

img {display:block;}

One problem that this can cause is when you want to have a few images next to each other - if they are block-level, they won't be next to each other. To get around that, you can use float:left. Of course, this might present another problem - maybe you don't want the image to float left. In this case, you can use an unordered list like this :

CSS  
ul, li {  
list-style-type:none;  
padding:0;  
margin:0 auto;  
}  
ul {  
width:150px;  
}  
li {  
float:left;  
}  
HTML  
<ul>  
<li><img src="wine.jpg" height="50" width="50" alt="wine" /></li>  
<li><img src="wine.jpg" height="50" width="50" alt="wine" /></li>  
<li><img src="wine.jpg" height="50" width="50" alt="wine" /></li>  
<li><img src="wine.jpg" height="50" width="50" alt="wine" /></li>  
<li><img src="wine.jpg" height="50" width="50" alt="wine" /></li>  
<li><img src="wine.jpg" height="50" width="50" alt="wine" /></li>  
<li><img src="wine.jpg" height="50" width="50" alt="wine" /></li>  
<li><img src="wine.jpg" height="50" width="50" alt="wine" /></li>  
<li><img src="wine.jpg" height="50" width="50" alt="wine" /></li>  
</ul>

1. **Question 22. As A Reader, How Can I Make My Browser Recognize My Own Style Sheet?**

**Answer :**

**Netscape**

It is not possible to do this in Netscape yet (as of version 4.0.)

Internet Explorer 3.0 (Win95/NT)

[It is possible to do this at least in Windows95/NT, but no user interface is provided. Unknown how this might be accomplished on other operating systems.]

1. Open the Registry editor (Start..Run..regedit..ENTER)  
2. Under the 'HKEY\_LOCAL\_MACHINESoftwareMicrosoftInternetExplorerStyles' key, Edit..New..String Value  
3. The new value should be called 'StyleSheet Pathname'  
4. For the value, type in the full directory path of your .css style sheet.

**Internet Explorer 4.0 (Win95/NT)**

1. Under the View menu, select 'Internet Options'.  
2. Under the 'General' tab, choose the 'Accessibility' button.  
3. Choose the 'Format documents using my style sheet' check box and 'Browse...' to the location of your .css style sheet.

[Pure.CSS Tutorial](https://www.wisdomjobs.com/e-university/puredotcss-tutorial-1224.html)

1. **Question 23. How Do I Design For Backward Compatibility Using Style Sheets?**

**Answer :**

Existing HTML style methods (such as <font SIZE> and <b>) may be easily combined with style sheet specification methods. Browsers that do not understand style sheets will use the older HTML formatting methods, and style sheetsspecifications can control the appearance of these elements in browsers that support CSS1.

[XHTML Interview Questions](https://www.wisdomjobs.com/e-university/xhtml-interview-questions.html)

1. **Question 24. What Are Pseudo-classes?**

**Answer :**

Pseudo-classes are fictional element types that do not exist in HTML. In CSS1 there is only one element type which can be classed this way, namely the A element(anchor). By creating three fictional types of the A element individual style can be attached to each class. These three fictional element types are: A as unvisited link, A as active link and A as visited link. Pseudo-classes are created by a colon followed by pseudo-class's name. They can also be combined with normal classes

**example:**

A:link {background: black; color: white}  
A:active {background: black; color: red}  
A:visited {background: transparent; color: black}  
<A HREF....>This anchor (or rather these anchors) will be displayed as declared above</A>  
A.foot:link {background: black; color: white}  
A.foft:active {background; black: color: red}  
A.foot:visited {background: transparent; color: black}  
<A CLASS=foot HREF....>This anchor and all other anchors with CLASS foot will be displayed as declared above</A>

[HTML Interview Questions](https://www.wisdomjobs.com/e-university/html-interview-questions.html)

1. **Question 25. Can Style Sheets And Html Stylistic Elements Be Used In The Same Document?**

**Answer :**

Yes. Style Sheets will be ignored in browsers without CSS-support and HTML stylistic elements used.

[XHTML Tutorial](https://www.wisdomjobs.com/e-university/xhtml-tutorial-1261.html)

1. **Question 26. Can Css Be Used With Other Than Html Documents?**

**Answer :**

Yes. CSS can be used with any ny structured document format. e.g. XML, however, the method of linking CSS with other document types has not been decided yet.

1. **Question 27. Some Examples Of Good And Bad Coding. What Is Wrong With This?**

**Answer :**

<font size="3">Welcome to my page</font>

**Comment:** The font-tag is design and design shouldn’t be in the HTML document. All design should be in the CSS-file! Instead do this:

**In the HTML:**

<h1>Welcome to my page</h1>

**In the CSS:**

h1 { font-size: 2em; }

**One more example:**

<b>An error occurred</b>

This looks right doesn’t it? But if you look up what <b> stands for you quickly find bold. But bold is certainly design, so it still doesn’t belong in the HTML document. A better choice is <em> that stands for emphasis or simply “this piece of text is important”. So instead of saying “this text looks like this” you are saying “this text is important” and you let the looks be decided by the CSS. Seems like a minor change, but it illustrates how to select your tags. Use this instead:

**In the HTML:**

<em>An error occured</em>

**In the CSS:**

em {  
font-weight: bold;  
color: Red;  
}  
**One last example:**  
<table>  
<tr><td><a href="">first link</a></td></tr>  
<tr><td><a href="">second link</a></td></tr>  
...  
</table>

[Dreamweaver Interview Questions](https://www.wisdomjobs.com/e-university/dreamweaver-interview-questions.html)

1. **Question 28. How To Use Css To Separate Content And Design?**

**Answer :**

The idea here is that all sites contain two major parts, the content: all your articles, text and photos and the design: rounded corners, colors and effects. Usually those two are made in different parts of a webpage’s lifetime. The design is determined at the beginning and then you start filling it with content and keep the design fixed.

In CSS you just add the nifty <link>-tag I’ve told you about to the head of yourHTML document and you have created a link to your design. In the HTML documentyou put content only, and that link of yours makes sure it looks right. You can also use the exact same link on many of your pages, giving them all of them the same design. You want to add content? Just write a plain HTML document and think about marking things up like “header” instead of “big blue header” and use CSS to make all headers look the way you want!

1. **Question 29. Why Was The Decision Made To Make Padding Apply Outside Of The Width Of A Box, Rather Than Inside, Which Would Seem To Make More Sense?**

**Answer :**

It makes sense in some situations, but not in others. For example, when a childelement is set to width: 100%, I don't think it should cover the padding of its parent. The box-sizing property in CSS3 addresses this issue. Ideally, the issue should have been addressed earlier, though.

1. **Question 30. Why Call The Subtended Angle A Pixel, Instead Of Something Else (e.g. Subangle)?**

**Answer :**

In most cases, a CSS pixel will be equal to a device pixel. But, as you point out, the definition of a CSS pixel will sometimes be different. For example, on a laser printer, one CSS pixel can be equal to 3x3 device pixels to avoid printing illegibly small text and images. I don't recall anyone ever proposing another name for it. Subangle? Personally, I think most people would prefer the pragmatic "px" to the non-intuitive "sa".

1. **Question 31. How Do I Eliminate The Blue Border Around Linked Images?**

**Answer :**

**in your CSS, you can specify the border property for linked images:**

a img { border: none ; }

However, note that removing the border that indicates an image is a link makes it harder for users to distinguish quickly and easily which images on a web page are clickable.

1. **Question 32. Which Set Of Definitions, Html Attributes Or Css Properties, Take Precedence?**

**Answer :**

CSS properties take precedence over HTML attributes. If both are specified, HTML attributes will be displayed in browsers without CSS support but won't have any effect in browsers with CSS support.

1. **Question 33. What Are Inline, Block, Parent, Children, Replaced And Floating Elements?**

**Answer :**

Inline elements which do not have line breaks. Can occur in block elements or other inline elements, cannot contain block elements.

Inline elements in HTML 3.2; EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP, A, IMG, APPLET, FONT, BASEFONT, BR, SCRIPT, MAP, INPUT, SELECT, TEXTAREA.

Inline elements in HTML 4.0; EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, ABBR, ACRONYM, TT, I, B, BIG, SMALL, SUB, SUP, A, IMG, OBJECT, BR, SCRIPT, MAP, Q, SPAN, BDO, INPUT, SELECT, TEXTAREA, LABEL, BUTTON, (INS, DEL).

Inline elements in HTML 4.0 Transitional; EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE, ABBR, ACRONYM, TT, I, B, U, S, STRIKE, BIG, SMALL, SUB, SUP, A, IMG, APPLET, OBJECT, FONT, BASEFONT, BR, SCRIPT, MAP, Q, SPAN, BDO, IFRAME, INPUT, SELECT, TEXTAREA, LABEL, BUTTON, (INS, DEL).

**Block**

elements which do have line breaks. May occur in other block elements, cannot occur in inline elements, may contain both block and inline elements.

Block elements in HTML 3.2; H1, H2, H3, H4, H5, H6, ADDRESS, P, DL, DT, DD, UL, OL, DIR, MENU, LI, DIV, CENTER, BLOCKQUOTE, PRE, HR, ISINDEX, TABLE, FORM.

Block elements in HTML 4.0; P, H1, H2, H3, H4, H5, H6, UL, OL, PRE, DL, DIV, NOSCRIPT, BLOCKQUOTE, FORM, HR, TABLE, FIELDSET, ADDRESS, (INS, DEL).

Block elements in HTML 4.0 Transitional; P, H1, H2, H3, H4, H5, H6, UL, OL, DIR, MENU, PRE, DL, DIV, CENTER, NOSCRIPT, NOFRAMES, BLOCKQUOTE, FORM, ISINDEX, HR, TABLE, FIELDSET, ADDRESS, (INS, DEL).

Parents and children elements which either contain (parents) or are in the content of (children) other elements, e.g. <P>text<STRONG>text</STRONG>text</P>. P is a parent of STRONG. STRONG is a child of P. If not specified otherwise, children will inherit parent's properties.

Replaced elements which content is replaced. For example content of the IMG element is replaced with an image, content of the INPUT element is replace with a field.

**Floating**

elements which follow the flow of a parent - inline elements.

[UI Developer Interview Questions](https://www.wisdomjobs.com/e-university/ui-developer-interview-questions.html)

1. **Question 34. How Do I Have A Fixed (non-scrolling) Background Image?**

**Answer :**

With CSS, you can use the background-attachment property. The backgroundattachment can be included in the shorthand background property, as in this example:

body {  
background: white url(example.gif) fixed ;  
color: black ;  
}

Note that this CSS is supported by Internet Explorer, Mozilla, Firefox Opera, Safari, and other browsers. In contrast, Microsoft's proprietary BGPROPERTIES attribute is supported only by Internet Explorer.

1. **Question 35. What Is A Style Sheet?**

**Answer :**

Style sheets are the way that standards-compliant Web designers define the layout, look-and-feel, and design of their pages. They are called Cascading Style Sheets or CSS. With style sheets, a designer can define many aspects of a Web page:

\* fonts  
\* colors  
\* layout  
\* positioning  
\* imagery  
\* accessibility

Style sheets give you a lot of power to define how your pages will look. And another great thing about them is that style sheets make it really easy to update your pages when you want to make a new design. Simply load in a new style sheet onto your pages and you're done.

1. **Question 36. What Is Inline Style? How To Link?**

**Answer :**

Inline style is the style attached to one specific element. The style is specified directly in the start tag as a value of the STYLE attribute and will apply exclusively to this specific element occurrence.

<P STYLE="text-indent: 10pt">Indented paragraph</P>

[CSS Interview Questions](https://www.wisdomjobs.com/e-university/css-interview-questions.html)

1. **Question 37. Why Do Style Sheets Exist?**

**Answer :**

SGML (of which HTML is a derivative) was meant to be a device-independent method for conveying a document's structural and semantic content (its meaning.) It was never meant to convey physical formatting information. HTML has crossed this line and now contains many elements and attributes which specify visual style and formatting information. One of the main reasons for style sheets is to stop the creation of new HTML physical formatting constructs and once again separate style information from document content.

1. **Question 38. What Are Style Sheets?**

**Answer :**

Style Sheets are templates, very similar to templates in desktop publishing applications, containing a collection of rules declared to various selectors(elements).

1. **Question 39. What Is Css Rule Ruleset?**

**Answer :**

There are two types of CSS rules: ruleset and at-rule. Ruleset identifies selector orselectors and declares style which is to be attached to that selector or selectors. For example P {text-indent: 10pt} is a CSS rule. CSS rulesets consist of two parts:selector, e.g. P and declaration,

e.g. {text-indent: 10pt}.  
P {text-indent: 10pt} - CSS rule (ruleset)  
{text-indent: 10pt} - CSS declaration  
text-indent - CSS property  
10pt - CSS value

1. **Question 40. What Is Embedded Style? How To Link?**

**Answer :**

Embedded style is the style attached to one specific document. The style information is specified as a content of the STYLE element inside the HEAD element and will apply to the entire document.

The Pack contains nearly 14 plus software . Pick the one which is suited for you Make your PC more useful. Get the free Google Pack.

<HEAD>  
<STYLE TYPE="text/css">  
<!--  
P {text-indent: 10pt}  
-->  
</STYLE>  
</HEAD>

**Note:** The styling rules are written as a HTML comment, that is, between <!-- and --> to hide the content in browsers without CSS support which would otherwise be displayed.

[WordPress Interview Questions](https://www.wisdomjobs.com/e-university/wordpress-interview-questions.html)

1. **Question 41. What Is Id Selector?**

**Answer :**

ID selector is an individually identified (named) selector to which a specific style is declared. Using the ID attribute the declared style can then be associated with one and only one HTML element per document as to differentiate it from all other elements. ID selectors are created by a character # followed by the selector's name. The name can contain characters a-z, A-Z, digits 0-9, period, hyphen, escaped characters, Unicode characters 161-255, as well as any Unicode characteras a numeric code, however, they cannot start with a dash or a digit.

#abc123 {color: red; background: black}

<P ID=abc123>This and only this element can be identified as abc123 </P>

* **What are the three main ways to add CSS to a webpage? Describe the advantages and disadvantages of each method.**

**A**

There are three ways to apply CSS to a webpage—inline, embedded, and as an external style sheet. The three approaches along with their pros and cons are described below.

* + Inline CSS can be written directly into the HTML elements as a style attribute. The primary advantage of inline CSS is the ability to override other style specifications in the single instance of an HTML element that it is applied to. However, this is only feasible if there are a small number of style definitions. It is generally better to use embedded or external style sheets for more complex styles.
  + External style sheets allow the developer to separate style from content, and control multiple HTML documents from a single separate file, making it easy to style the entirety of a site with a single document. It enables complex styling through classes, selectors, and other grouping methods. The disadvantage of an external CSS file is that it must be downloaded first for the HTML file to be properly rendered.
  + Embedded CSS can be written within the <style> tags inside the <head> section of an HTML document. It shares many of the same advantages as the External Style Sheet, with access to classes, selectors, and more complex styling. Embedded CSS has the added advantage of loading with the HTML document—no extra download required. However, that also means that any external HTML documents will not inherit the styling of CSS written within these tags.

HIDE THE ANSWER

* **Q**

**What are CSS media queries and what are they used for?**

**A**

CSS media queries are the filters that make responsive web design (RWD) possible. With the introduction of laptops, tablets, and smartphones with screens of varying sizes and aspect ratios, RWD is crucial for modern day app development. CSS media queries adjust content style based on device characteristics like width, height, orientation, resolution, and display type. When used properly, the end result is a website or app capable of providing a sleek UI/UX across multiple devices.

HIDE THE ANSWER

* **Q**

**What is a CSS preprocessor? Would you recommend using one for this project?**

**A**

A preprocessor is an abstraction layer built on top of CSS. Preprocessors extend the functionality of CSS by offering powerful features like variables, inheritable “classes” called extends, and “function-like” mixins. Sass, LESS, and Stylus are some of the more well-known preprocessors—try asking the developer which one they prefer more. Selecting a preprocessor really boils down to preference, and it can be revealing to see how a particular developer might decide to use one over the other for your project.

HIDE THE ANSWER

* **Q**

**List the basic layout components of the CSS box model with a brief description for each.**

**A**

In many ways, front-end web design is all about managing rectangles, and the CSS box model provides a layout paradigm for HTML elements that you can use to structure a webpage. The basic components are described below.

* + **border:** The border surrounding the padding.
  + **content:** Any text or images within the box.
  + **margin:** The transparent area surrounding borders.
  + **padding:** The transparent area surrounding content.

HIDE THE ANSWER

* **Q**

**How would you implement the basic layout components of the box model in CSS? Give an example.**

**A**

Each element of the box model—border, content, margin, and padding—can be specified independently for each side of the element by listing dimensions in the following order: top, bottom, right, and left. Alternatively, multiple sides can be specified as a group by listing fewer parameters. An example has been provided below.

margin: 50px 100px 100px 50px;

/\* Sets the top, right, bottom and left margins \*/

margin: 25px;

/\* Sets the margin on all sides \*/

padding: 50px 100px;

/\* Sets the top/bottom margin as a group and the right/left margin as a group \*/

HIDE THE ANSWER

* **Q**

**Give an example of how you would use counter-increment and counter-reset in CSS to create automatic numbering within a webpage.**

**A**

The counter-reset and counter-increment properties allow a developer to automatically number CSS elements like an ordered list (<ol>). The counter-reset property resets a CSS counter to a given value. The counter-increment property then increases one or more counter values. Automatic numbering within a webpage is often useful, much like the section headers within this article. An example of how to use counters in CSS is displayed below.

body {

counter-reset: foo;

}

h1 {

counter-reset: bar;

}

h1:before {

counter-increment: foo;

content: "Section " counter(foo) ". ";

}

h2:before {

counter-increment: bar;

content: counter(foo) "." counter(bar) " ";

}

HIDE THE ANSWER

* **Q**

**Describe the following common CSS units of length: cm, em, in, mm, pc, pt, and px.**

**A**

There are many ways to express units of length within CSS, but these are just some of the more common ones.

* + cm: centimeters
  + em: a relative unit of measurement based on the size of a font
  + in: inches
  + mm: millimeters
  + pc: pica, a unit of length equivalent to 12 points, or 1/6 of an inch
  + pt: 1/72 of an inch
  + px: a device-specific relative measurement equivalent to a certain number of pixels on a display

HIDE THE ANSWER

* **Q**

**What are CSS vendor prefixes? Can you name some of the more common ones that you're familiar with?**

**A**

Depending on your project, you might be looking for a CSS developer who can take advantage of experimental non-standard features that are only available on certain platforms. Vendor prefixes are extensions to CSS standards that can be added to these features to prevent incompatibilities from arising when the standard is extended. CSS vendor prefixes for some common platforms are listed below.

* + -webkit-: Android, Chrome, iOS, and Safari
  + -moz-: Mozilla Firefox
  + -ms-: Internet Explorer
  + -o-: Opera

HIDE THE ANSWER

* **Q**

**How do you define a pseudo class in CSS? What are they used for?**

**A**

You can define a pseudo class by listing the selector followed by a colon and finally the pseudo class element. 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.

HIDE THE ANSWER

* **Q**

**How would you select all the PDF links in the code block below with a single line of code?**

**A**

<body>

<p><a href="default.asp" target="\_blank">This is a link</a></p>

<p><a href="mydocument.pdf" target="\_blank">This is a PDF</a></p>

<p><a href="default.asp" target="\_blank">This is a link</a></p>

<p><a href="mydocument.pdf" target="\_blank">This is a PDF</a></p>

</body>

 A big part of CSS is selecting and stylizing particular elements on a page. This question tests a developer’s knowledge of attribute selectors. All the elements of interest on the page happen to be links, but not all links are linking to PDFs. In CSS you can target the ending of the PDF link “.pdf” to quickly select all the PDF files by using the [attribute$=”value”] selector, which selects elements whose value ends with a specified value. In this case, we can use a[href$=“.pdf”] to select all PDF links.

**What is CSS selector specificity and how does it work?**

The browser determines what styles to show on an element depending on the specificity of CSS rules. We assume that the browser has already determined the rules that match a particular element. Among the matching rules, the specificity, four comma-separate values, a, b, c, d are calculated for each rule based on the following:

1. a is whether inline styles are being used. If the property declaration is an inline style on the element, a is 1, else 0.
2. b is the number of ID selectors.
3. c is the number of classes, attributes and pseudo-classes selectors.
4. d is the number of tags and pseudo-elements selectors.

The resulting specificity is not a score, but a matrix of values that can be compared column by column. When comparing selectors to determine which has the highest specificity, look from left to right, and compare the highest value in each column. So a value in column b will override values in columns c and d, no matter what they might be. As such, specificity of 0,1,0,0 would be greater than one of 0,0,10,10.

In the cases of equal specificity: the latest rule is the one that counts. If you have written the same rule into your stylesheet (regardless of internal or external) twice, then the lower rule in your style sheet is closer to the element to be styled, it is deemed to be more specific and therefore will be applied.

I would write CSS rules with low specificity so that they can be easily overridden if necessary. When writing CSS UI component library code, it is important that they have low specificities so that users of the library can override them without using too complicated CSS rules just for the sake of increasing specificity or resorting to !important.

**References**

* <https://www.smashingmagazine.com/2007/07/css-specificity-things-you-should-know/>
* <https://www.sitepoint.com/web-foundations/specificity/>

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/css-questions.md#table-of-contents)

**What's the difference between "resetting" and "normalizing" CSS? Which would you choose, and why?**

* **Resetting** - Resetting is meant to strip all default browser styling on elements. For e.g. margins, paddings, font-sizes of all elements are reset to be the same. You will have to redeclare styling for common typographic elements.
* **Normalizing** - Normalizing preserves useful default styles rather than "unstyling" everything. It also corrects bugs for common browser dependencies.

I would choose resetting when I have a very customized or unconventional site design such that I need to do a lot of my own styling and do not need any default styling to be preserved.

**References**

* <https://stackoverflow.com/questions/6887336/what-is-the-difference-between-normalize-css-and-reset-css>

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/css-questions.md#table-of-contents)

**Describe floats and how they work.**

Float is a CSS positioning property. Floated elements remain a part of the flow of the page, and will affect the positioning of other elements (e.g. text will flow around floated elements), unlike position: absolute elements, which are removed from the flow of the page.

The CSS clear property can be used to be positioned below left/right/both floated elements.

If a parent element contains nothing but floated elements, its height will be collapsed to nothing. It can be fixed by clearing the float after the floated elements in the container but before the close of the container.

The .clearfix hack uses a clever CSS [pseudo selector](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/css-questions.md#describe-pseudo-elements-and-discuss-what-they-are-used-for) (:after) to clear floats. Rather than setting the overflow on the parent, you apply an additional class clearfix to it. Then apply this CSS:

.clearfix:after {

content: ' ';

visibility: hidden;

display: block;

height: 0;

clear: both;

}

Alternatively, give overflow: auto or overflow: hidden property to the parent element which will establish a new block formatting context inside the children and it will expand to contain its children.

**References**

* <https://css-tricks.com/all-about-floats/>

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/css-questions.md#table-of-contents)

**Describe z-index and how stacking context is formed.**

The z-index property in CSS controls the vertical stacking order of elements that overlap. z-index only affects elements that have a position value which is not static.

Without any z-index value, elements stack in the order that they appear in the DOM (the lowest one down at the same hierarchy level appears on top). Elements with non-static positioning (and their children) will always appear on top of elements with default static positioning, regardless of HTML hierarchy.

A stacking context is an element that contains a set of layers. Within a local stacking context, the z-index values of its children are set relative to that element rather than to the document root. Layers outside of that context — i.e. sibling elements of a local stacking context — can't sit between layers within it. If an element B sits on top of element A, a child element of element A, element C, can never be higher than element B even if element C has a higher z-index than element B.

Each stacking context is self-contained - after the element's contents are stacked, the whole element is considered in the stacking order of the parent stacking context. A handful of CSS properties trigger a new stacking context, such as opacity less than 1, filter that is not none, and transform that is notnone.

*Note: What exactly qualifies an element to create a stacking context is listed in this long set of*[*rules*](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Positioning/Understanding_z_index/The_stacking_context#The_stacking_context)*.*

**References**

* <https://css-tricks.com/almanac/properties/z/z-index/>
* <https://philipwalton.com/articles/what-no-one-told-you-about-z-index/>
* <https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Positioning/Understanding_z_index/The_stacking_context>

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/css-questions.md#table-of-contents)

**Describe Block Formatting Context (BFC) and how it works.**

A Block Formatting Context (BFC) is part of the visual CSS rendering of a web page in which block boxes are laid out. Floats, absolutely positioned elements, inline-blocks, table-cells, table-captions, and elements with overflow other than visible (except when that value has been propagated to the viewport) establish new block formatting contexts.

Knowing how to establish a block formatting context is important, because without doing so, the containing box will not [contain floated children](https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Block_formatting_context#Make_float_content_and_alongside_content_the_same_height). This is similar to collapsing margins, but more insidious as you will find entire boxes collapsing in odd ways.

A BFC is an HTML box that satisfies at least one of the following conditions:

* The value of float is not none.
* The value of position is neither static nor relative.
* The value of display is table-cell, table-caption, inline-block, flex, or inline-flex.
* The value of overflow is not visible.

In a BFC, each box's left outer edge touches the left edge of the containing block (for right-to-left formatting, right edges touch).

Vertical margins between adjacent block-level boxes in a BFC collapse. Read more on [collapsing margins](https://www.sitepoint.com/web-foundations/collapsing-margins/).

**References**

* <https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Block_formatting_context>
* <https://www.sitepoint.com/understanding-block-formatting-contexts-in-css/>

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/css-questions.md#table-of-contents)

**What are the various clearing techniques and which is appropriate for what context?**

* Empty div method - <div style="clear:both;"></div>.
* Clearfix method - Refer to the .clearfix class above.
* overflow: auto or overflow: hidden method - Parent will establish a new block formatting context and expand to contains its floated children.

In large projects, I would write a utility .clearfix class and use them in places where I need it. overflow: hidden might clip children if the children is taller than the parent and is not very ideal.

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/css-questions.md#table-of-contents)

**Explain CSS sprites, and how you would implement them on a page or site.**

CSS sprites combine multiple images into one single larger image. It is a commonly-used technique for icons (Gmail uses it). How to implement it:

1. Use a sprite generator that packs multiple images into one and generate the appropriate CSS for it.
2. Each image would have a corresponding CSS class with background-image, background-position and background-size properties defined.
3. To use that image, add the corresponding class to your element.

**Advantages:**

* Reduce the number of HTTP requests for multiple images (only one single request is required per spritesheet). But with HTTP2, loading multiple images is no longer much of an issue.
* Advance downloading of assets that won't be downloaded until needed, such as images that only appear upon :hover pseudo-states. Blinking wouldn't be seen.

**References**

* <https://css-tricks.com/css-sprites/>

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/css-questions.md#table-of-contents)

**How would you approach fixing browser-specific styling issues?**

* After identifying the issue and the offending browser, use a separate style sheet that only loads when that specific browser is being used. This technique requires server-side rendering though.
* Use libraries like Bootstrap that already handles these styling issues for you.
* Use autoprefixer to automatically add vendor prefixes to your code.
* Use Reset CSS or Normalize.css.
* If you're using Postcss (or a similar transpiling library), there may be plugins which allow you to opt in for using modern CSS syntax (and even W3C proposals) that will transform those sections of your code into corresponding safe code that will work in the targets you've used.

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/css-questions.md#table-of-contents)

**How do you serve your pages for feature-constrained browsers? What techniques/processes do you use?**

* Graceful degradation - The practice of building an application for modern browsers while ensuring it remains functional in older browsers.
* Progressive enhancement - The practice of building an application for a base level of user experience, but adding functional enhancements when a browser supports it.
* Use [caniuse.com](https://caniuse.com/) to check for feature support.
* Autoprefixer for automatic vendor prefix insertion.
* Feature detection using [Modernizr](https://modernizr.com/).
* Use CSS Feature queries [@support](https://developer.mozilla.org/en-US/docs/Web/CSS/@supports)

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/css-questions.md#table-of-contents)

**What are the different ways to visually hide content (and make it available only for screen readers)?**

These techniques are related to accessibility (a11y).

* width: 0; height: 0. Make the element not take up any space on the screen at all, resulting in not showing it.
* position: absolute; left: -99999px. Position it outside of the screen.
* text-indent: -9999px. This only works on text within the block elements.
* Meta tags. For example by using Schema.org, RDF, and JSON-LD.
* WAI-ARIA. A W3C technical specification that specifies how to increase the accessibility of web pages.

Even if WAI-ARIA is the ideal solution, I would go with the absolute positioning approach, as it has the least caveats, works for most elements and it's an easy technique.

**References**

* <https://www.w3.org/TR/wai-aria-1.1/>
* <https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA>
* <http://a11yproject.com/>

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/css-questions.md#table-of-contents)

**Have you ever used a grid system, and if so, what do you prefer?**

Before Flex became popular (around 2014), the float-based grid system was the most reliable because it still has the most browser support among the alternative existing systems (flex, grid). Bootstrap was using the float approach until Bootstrap 4 which switched to the flex-based approach. As of writing (2020), flex is the recommended approach for building grid systems and has [decent browser support](https://caniuse.com/#search=flex).

For the adventurous, they can look into [CSS Grid Layout](https://css-tricks.com/snippets/css/complete-guide-grid/), which uses the shiny new grid property; it is even better than flex for building grid layouts and will be the de facto way to do so in the future.

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/css-questions.md#table-of-contents)

**Have you used or implemented media queries or mobile-specific layouts/CSS?**

Yes. An example would be transforming a stacked pill navigation into a fixed-bottom tab navigation beyond a certain breakpoint.

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/css-questions.md#table-of-contents)

**Are you familiar with styling SVG?**

Yes, there are several ways to color shapes (including specifying attributes on the object) using inline CSS, an embedded CSS section, or an external CSS file. Most SVG you'll find around the web use inline CSS, but there are advantages and disadvantages associated with each type.

Basic coloring can be done by setting two attributes on the node: fill and stroke. fill sets the color inside the object and stroke sets the color of the line drawn around the object. You can use the same CSS color naming schemes that you use in HTML, whether that's color names (that is red), RGB values (that is rgb(255,0,0)), Hex values, RGBA values, etc.

<rect

x="10"

y="10"

width="100"

height="100"

stroke="blue"

fill="purple"

fill-opacity="0.5"

stroke-opacity="0.8"

/>

The above fill="purple" is an example of a *presentational attribute*. Interestingly, and unlike inline styles like style="fill: purple" which also happens to be an attribute, presentational attributes can be [overriden by CSS](https://css-tricks.com/presentation-attributes-vs-inline-styles/) styles defined in a stylesheet. So, if you did something like svg { fill: blue; } it would override the purple fill we've defined.

**References**

* <https://developer.mozilla.org/en-US/docs/Web/SVG/Tutorial/Fills_and_Strokes>

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/css-questions.md#table-of-contents)

**Can you give an example of an @media property other than screen?**

Yes, there are four types of @media properties (including *screen*):

* all - for all media type devices
* print - for printers
* speech - for screenreaders that "reads" the page out loud
* screen - for computer screens, tablets, smart-phones etc.

Here is an example of print media type's usage:

@media print {

body {

color: black;

}

}

**References**

* <https://developer.mozilla.org/en-US/docs/Web/CSS/@media#Syntax>

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/css-questions.md#table-of-contents)

**What are some of the "gotchas" for writing efficient CSS?**

Firstly, understand that browsers match selectors from rightmost (key selector) to left. Browsers filter out elements in the DOM according to the key selector and traverse up its parent elements to determine matches. The shorter the length of the selector chain, the faster the browser can determine if that element matches the selector. Hence avoid key selectors that are tag and universal selectors. They match a large number of elements and browsers will have to do more work in determining if the parents do match.

[BEM (Block Element Modifier)](https://bem.info/) methodology recommends that everything has a single class, and, where you need hierarchy, that gets baked into the name of the class as well, this naturally makes the selector efficient and easy to override.

Be aware of which CSS properties [trigger](https://csstriggers.com/) reflow, repaint, and compositing. Avoid writing styles that change the layout (trigger reflow) where possible.

**References**

* <https://developers.google.com/web/fundamentals/performance/rendering/>
* <https://csstriggers.com/>

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/css-questions.md#table-of-contents)

**What are the advantages/disadvantages of using CSS preprocessors?**

**Advantages:**

* CSS is made more maintainable.
* Easy to write nested selectors.
* Variables for consistent theming. Can share theme files across different projects.
* Mixins to generate repeated CSS.
* Sass features like loops, lists, and maps can make configuration easier and less verbose.
* Splitting your code into multiple files. CSS files can be split up too but doing so will require an HTTP request to download each CSS file.

**Disadvantages:**

* Requires tools for preprocessing. Re-compilation time can be slow.
* Not writing currently and potentially usable CSS. For example, by using something like [postcss-loader](https://github.com/postcss/postcss-loader) with [webpack](https://webpack.js.org/), you can write potentially future-compatible CSS, allowing you to use things like CSS variables instead of Sass variables. Thus, you're learning new skills that could pay off if/when they become standardized.

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/css-questions.md#table-of-contents)

**Describe what you like and dislike about the CSS preprocessors you have used.**

**Likes:**

* Mostly the advantages mentioned above.
* Less is written in JavaScript, which plays well with Node.

**Dislikes:**

* I use Sass via node-sass, which is a binding for LibSass written in C++. I have to frequently recompile it when switching between node versions.
* In Less, variable names are prefixed with @, which can be confused with native CSS keywords like @media, @import and @font-face rule.

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/css-questions.md#table-of-contents)

**How would you implement a web design comp that uses non-standard fonts?**

Use @font-face and define font-family for different font-weights.

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/css-questions.md#table-of-contents)

**Explain how a browser determines what elements match a CSS selector.**

This part is related to the above about [writing efficient CSS](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/css-questions.md#what-are-some-of-the-gotchas-for-writing-efficient-css). Browsers match selectors from rightmost (key selector) to left. Browsers filter out elements in the DOM according to the key selector and traverse up its parent elements to determine matches. The shorter the length of the selector chain, the faster the browser can determine if that element matches the selector.

For example with this selector p span, browsers firstly find all the <span> elements and traverse up its parent all the way up to the root to find the <p> element. For a particular <span>, as soon as it finds a <p>, it knows that the <span> matches and can stop its matching.

**References**

* <https://stackoverflow.com/questions/5797014/why-do-browsers-match-css-selectors-from-right-to-left>

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/css-questions.md#table-of-contents)

**Describe pseudo-elements and discuss what they are used for.**

A CSS pseudo-element is a keyword added to a selector that lets you style a specific part of the selected element(s). They can be used for decoration (:first-line, :first-letter) or adding elements to the markup (combined with content: ...) without having to modify the markup (:before, :after).

* :first-line and :first-letter can be used to decorate text.
* Used in the .clearfix hack as shown above to add a zero-space element with clear: both.
* Triangular arrows in tooltips use :before and :after. Encourages separation of concerns because the triangle is considered part of styling and not really the DOM.

**References**

* <https://css-tricks.com/almanac/selectors/a/after-and-before/>

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/css-questions.md#table-of-contents)

**Explain your understanding of the box model and how you would tell the browser in CSS to render your layout in different box models.**

The CSS box model describes the rectangular boxes that are generated for elements in the document tree and laid out according to the visual formatting model. Each box has a content area (e.g. text, an image, etc.) and optional surrounding padding, border, and margin areas.

The CSS box model is responsible for calculating:

* How much space a block element takes up.
* Whether or not borders and/or margins overlap, or collapse.
* A box's dimensions.

The box model has the following rules:

* The dimensions of a block element are calculated by width, height, padding, borders, and margins.
* If no height is specified, a block element will be as high as the content it contains, plus padding (unless there are floats, for which see below).
* If no width is specified, a non-floated block element will expand to fit the width of its parent minus padding.
* The height of an element is calculated by the content's height.
* The width of an element is calculated by the content's width.
* By default, paddings and borders are not part of the width and height of an element.

**References**

* <https://www.smashingmagazine.com/2010/06/the-principles-of-cross-browser-css-coding/#understand-the-css-box-model>

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/css-questions.md#table-of-contents)

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

* 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.

**References**

* <https://www.paulirish.com/2012/box-sizing-border-box-ftw/>

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/css-questions.md#table-of-contents)

**What is the CSS display property and can you give a few examples of its use?**

* none, block, inline, inline-block, flex, grid, table, table-row, table-cell, list-item.

| **display** | **Description** |
| --- | --- |
| none | Does not display an element (the element no longer affects the layout of the document). All child element are also no longer displayed. The document is rendered as if the element did not exist in the document tree |
| block | The element consumes the whole line in the block direction (which is usually horizontal) |
| inline | Elements can be laid out beside each other |
| inline-block | Similar to inline, but allows some block properties like setting width and height |
| table | Behaves like the <table> element |
| table-row | Behaves like the <tr> element |
| table-cell | Behaves like the <td> element |
| list-item | Behaves like a <li> element which allows it to define list-style-type and list-style-position |

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/css-questions.md#table-of-contents)

**What's the difference between inline and inline-block?**

I shall throw in a comparison with block for good measure.

|  | **block** | **inline-block** | **inline** |
| --- | --- | --- | --- |
| Size | Fills up the width of its parent container. | Depends on content. | Depends on content. |
| Positioning | Start on a new line and tolerates no HTML elements next to it (except when you add float) | Flows along with other content and allows other elements beside it. | Flows along with other content and allows other elements beside it. |
| Can specify width and height | Yes | Yes | No. Will ignore if being set. |
| Can be aligned with vertical-align | No | Yes | Yes |
| Margins and paddings | All sides respected. | All sides respected. | Only horizontal sides respected. Vertical sides, if specified, do not affect layout. Vertical space it takes up depends on line-height, even though the border and padding appear visually around the content. |
| Float | - | - | Becomes like a block element where you can set vertical margins and paddings. |

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/css-questions.md#table-of-contents)

**What's the difference between a relative, fixed, absolute and statically positioned element?**

A positioned element is an element whose computed position property is either relative, absolute, fixed or sticky.

* static - The default position; the element will flow into the page as it normally would. The top, right, bottom, left and z-index properties do not apply.
* relative - The element's position is adjusted relative to itself, without changing layout (and thus leaving a gap for the element where it would have been had it not been positioned).
* absolute - The element is removed from the flow of the page and positioned at a specified position relative to its closest positioned ancestor if any, or otherwise relative to the initial containing block. Absolutely positioned boxes can have margins, and they do not collapse with any other margins. These elements do not affect the position of other elements.
* fixed - The element is removed from the flow of the page and positioned at a specified position relative to the viewport and doesn't move when scrolled.
* 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.

**References**

* <https://developer.mozilla.org/en/docs/Web/CSS/position>

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/css-questions.md#table-of-contents)

**What existing CSS frameworks have you used locally, or in production? How would you change/improve them?**

* **Bootstrap** - Slow release cycle. Bootstrap 4 has been in alpha for almost 2 years. Add a spinner button component, as it is widely used.
* **Semantic UI** - Source code structure makes theme customization extremely hard to understand. Its unconventional theming system is a pain to customize. Hardcoded config path within the vendor library. Not well-designed for overriding variables unlike in Bootstrap.
* **Bulma** - A lot of non-semantic and superfluous classes and markup required. Not backward compatible. Upgrading versions breaks the app in subtle manners.

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/css-questions.md#table-of-contents)

**Have you played around with the new CSS Flexbox or Grid specs?**

Yes. Flexbox is mainly meant for 1-dimensional layouts while Grid is meant for 2-dimensional layouts.

Flexbox solves many common problems in CSS, such as vertical centering of elements within a container, sticky footer, etc. Bootstrap and Bulma are based on Flexbox, and it is probably the recommended way to create layouts these days. Have tried Flexbox before but ran into some browser incompatibility issues (Safari) in using flex-grow, and I had to rewrite my code using inline-blocks and math to calculate the widths in percentages, it wasn't a nice experience.

Grid is by far the most intuitive approach for creating grid-based layouts (it better be!) but browser support is not wide at the moment.

**References**

* <https://philipwalton.github.io/solved-by-flexbox/>

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/css-questions.md#table-of-contents)

**Can you explain the difference between coding a website to be responsive versus using a mobile-first strategy?**

Note that these two 2 approaches are not exclusive.

Making a website responsive means the some elements will respond by adapting its size or other functionality according to the device's screen size, typically the viewport width, through CSS media queries, for example, making the font size smaller on smaller devices.

@media (min-width: 601px) {

.my-class {

font-size: 24px;

}

}

@media (max-width: 600px) {

.my-class {

font-size: 12px;

}

}

A mobile-first strategy is also responsive, however it agrees we should default and define all the styles for mobile devices, and only add specific responsive rules to other devices later. Following the previous example:

.my-class {

font-size: 12px;

}

@media (min-width: 600px) {

.my-class {

font-size: 24px;

}

}

A mobile-first strategy has 2 main advantages:

* It's more performant on mobile devices, since all the rules applied for them don't have to be validated against any media queries.
* It forces to write cleaner code in respect to responsive CSS rules.

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/css-questions.md#table-of-contents)

**How is responsive design different from adaptive design?**

Both responsive and adaptive design attempt to optimize the user experience across different devices, adjusting for different viewport sizes, resolutions, usage contexts, control mechanisms, and so on.

Responsive design works on the principle of flexibility - a single fluid website that can look good on any device. Responsive websites use media queries, flexible grids, and responsive images to create a user experience that flexes and changes based on a multitude of factors. Like a single ball growing or shrinking to fit through several different hoops.

Adaptive design is more like the modern definition of progressive enhancement. Instead of one flexible design, adaptive design detects the device and other features and then provides the appropriate feature and layout based on a predefined set of viewport sizes and other characteristics. The site detects the type of device used and delivers the pre-set layout for that device. Instead of a single ball going through several different-sized hoops, you'd have several different balls to use depending on the hoop size.

Both have these methods have some issues that need to be weighed:

* Responsive design can be quite challenging, as you're essentially using a single albeit responsive layout to fit all situations. How to set the media query breakpoints is one such challenge. Do you use standardized breakpoint values? Or, do you use breakpoints that make sense to your particular layout? What if that layout changes?
* Adaptive design generally requires user agent sniffing, or DPI detection, etc., all of which can prove unreliable.

**References**

* <https://developer.mozilla.org/en-US/docs/Archive/Apps/Design/UI_layout_basics/Responsive_design_versus_adaptive_design>
* <http://mediumwell.com/responsive-adaptive-mobile/>
* <https://css-tricks.com/the-difference-between-responsive-and-adaptive-design/>

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/css-questions.md#table-of-contents)

**Have you ever worked with retina graphics? If so, when and what techniques did you use?**

*Retina* is just a marketing term to refer to high resolution screens with a pixel ratio bigger than 1. The key thing to know is that using a pixel ratio means these displays are emulating a lower resolution screen in order to show elements with the same size. Nowadays we consider all mobile devices *retina* defacto displays.

Browsers by default render DOM elements according to the device resolution, except for images.

In order to have crisp, good-looking graphics that make the best of retina displays we need to use high resolution images whenever possible. However using always the highest resolution images will have an impact on performance as more bytes will need to be sent over the wire.

To overcome this problem, we can use responsive images, as specified in HTML5. It requires making available different resolution files of the same image to the browser and let it decide which image is best, using the html attribute srcset and optionally sizes, for instance:

<div responsive-background-image>

<img

src="/images/test-1600.jpg"

sizes="

(min-width: 768px) 50vw,

(min-width: 1024px) 66vw,

100vw"

srcset="

/images/test-400.jpg 400w,

/images/test-800.jpg 800w,

/images/test-1200.jpg 1200w

"

/>

</div>

It is important to note that browsers which don't support HTML5's srcset (i.e. IE11) will ignore it and use src instead. If we really need to support IE11 and we want to provide this feature for performance reasons, we can use a JavaScript polyfill, e.g. Picturefill (link in the references).

For icons, I would also opt to use SVGs and icon fonts where possible, as they render very crisply regardless of resolution.

**References**

* <https://css-tricks.com/responsive-images-youre-just-changing-resolutions-use-srcset/>
* <http://scottjehl.github.io/picturefill/>
* <https://aclaes.com/responsive-background-images-with-srcset-and-sizes/>

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/css-questions.md#table-of-contents)

**Is there any reason you'd want to use translate() instead of absolute positioning, or vice-versa? And why?**

translate() is a value of CSS transform. Changing transform or opacity does not trigger browser reflow or repaint but does trigger compositions; whereas changing the absolute positioning triggers reflow. transform causes the browser to create a GPU layer for the element but changing absolute positioning properties uses the CPU. Hence translate() is more efficient and will result in shorter paint times for smoother animations.

When using translate(), the element still occupies its original space (sort of like position: relative), unlike in changing the absolute positioning.

**References**

* <https://www.paulirish.com/2012/why-moving-elements-with-translate-is-better-than-posabs-topleft/>