**1. What is 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.

A stylesheet language used to describe the presentation of a document written in a markup language.......

**2. 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.

**3. 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.

**4. 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

**5.** [**What are the various style sheets? - CSS**](https://www.careerride.com/css-style-sheets.aspx)

Inline, external, imported and embedded are the different types of style sheets.......

**6.** [**What are style sheet properties? - CSS**](https://www.careerride.com/css-style-sheet-properties.aspx)

Style sheet properties - CSS Background, CSS Text, CSS Font......

**7. 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.

**8. 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

**9. Name a few prominent CSS frameworks?**

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

* **Bootstrap**

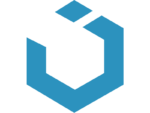
****

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/" \t "_blank). 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.

**10. What are different ways to integrate a CSS into a Web page/HTML page?**

There are three ways to integrate CSS into a Web page  
  
**1) Inline :** HTML elements may have CSS applied to them via the STYLE attribute.  
**2) Embedded :** By placing the code in a STYLE element within the HEAD element.  
**3) Linked/ Imported :** Place the CSS in an external file and link it via a link element.

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.

|  |
| --- |
|  |
| **11. How can imported style sheets be linked? Explain with the help of an example?**  - Imported style sheets are those style sheets which as their name suggests can be combined for use with another sheet. By doing so the designer is able to create a single main sheet which has declarations that apply to the whole site and specific style sheets that apply only to certain specific elements. - In order to import a style sheet the style element must contain the @notation. In case of multiple sheets being imported they will be done so in the order they are imported. The last imported sheet would over write any previous style sheet property.  **For Example:**  <LINK REL=STYLESHEET HREF="major.css" TYPE="text/css">< BR><STYLETYPE="text=css"> <!-- @import url(http://www.css.stylesheet1.css); @import url(http://www.css.stylesheet2.css); --> </STYLE>  Imported Style Sheet allows you to import the custom made style sheet from anywhere, in your HTML document. In this, one main sheet is made that consists of declarations applied to the whole site and other sheet is made, that consists of only specific declarations applied to specific documents. To link imported style sheet to HTML document @import is used in the STYLE element. For example: <STYLE TYPE="text=css"> <!-- @import url(http://www.and.so.on.partial1.css); @import url(http://www.and.so.on.partial2.css); .... Other statements --> </STYLE>  **12. Explain external Style Sheet? How would you link to it?**  1. External Style Sheet can be called as a template/document/file which contains style information and can be linked with more than one HTML documents. 2. Using this the entire site can be formatted and styles just by editing one file. 3. The file is linked with HTML documents via the LINK element inside the HEAD element.  <HEAD> <LINK REL=STYLESHEET HREF="style.css" TYPE="text/css"> </HEAD> |
| External Style Sheet allows you to write the style content just like you do in a normal style sheet, but the advantage of it is that, it can be used anywhere and can be linked with HTML documents. It is very easy, convenient and portable way to use the formatting for the entire site. External Style sheets can be linked with HTML documents by using a LINK element in HEAD area. It has the extension as .css. Example code: <HEAD> <LINK href="special.css" rel="stylesheet" type="text/css"> </HEAD>  External style sheet are made up of css format only, it contains style information that can be linked with the HTML document externally. It is one of the easy and structured way as it keeps the style separate from the structure. It is a convenient way as only one file will be affected if any changes will be made overall. The file is linked through Link tag used inside the HTML Head.  The syntax is as follows:  <HTML> <HEAD> <LINK REL=STYLESHEET HREF="style.css" TYPE="text/css"> </HEAD> </HTML>  **13. What are the advantages and disadvantages of External Style Sheets?**  **The advantages of External Style Sheets are:**  1. Using them, the styles of multiple documents can be controlled from one file. 2. Classes can be created for use on multiple HTML element types in many documents. 3. In complex situations, selector and grouping methods can be used to apply styles.  **The disadvantages of External Style Sheets are:**  - In order to import style information for each document, an extra download is needed. - Until the external style sheet is loaded, it may not be possible to render the document. - For small number of style definitions, it is not viable. |
| **14. What are the advantages and disadvantages of Embedded Style Sheets?**  **The advantages of Embedded Style Sheets are:**  1. It is possible to create classes for use on multiple tag types in the document 2. Under complex situations, selector and grouping methods can be used to apply styles. 3. No extra download is required to import the information.  **The disadvantages of Embedded Style Sheets are:**  1. It is not possible to control the styles for multiple documents from one file, using this method. |
| **15. What are the advantages and disadvantages of Inline Styles?**  **The advantages of Inline Styles are:**  1. It is especially useful for small number of style definitions. 2. It has the ability to override other style specification methods at the local level.  **The disadvantages of Inline Styles are:**  1. It does not separate out the style information from content. 2. The styles for many documents can not be controlled from one source. 3. Selector grouping methods can not be used to handle complex situations. 4. Control classes can not be created to control multiple element types within the document. |
| **16. How can you eliminate the blue border around linked images on web page?**  This can be done by specifying the border property for linked images in your CSS as none: **For Example:**  a img { border: none ; }  However, this makes it difficult for the users to differentiate between the clickable and non-clickable images. |
| **17. What is Tweening?**  1. It is the short form for in-betweening. 2. It is the process of generating intermediate frames between two images. 3. It gives the impression that the first image has smoothly evolved into the second one. 4. It is an important method used in all types of animations. 5. In CSS3, Transforms(matrix,translate,rotate,scale etc) module can be used to achieve tweening. |
| **18. Explain RWD?**  1. RWD is the abbreviation for Responsive web design. 2. In this technique, the designed page is perfectly displayed on every screen size and device, be it desktop, mobile, laptop or any other device. You don’t need to create a different page for each device. |

**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.

**19. 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;

}

}

@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.

**20. 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.

Image sprites are basically a collection of images put into a single image. A web page can contain multiple images and loading them all one by one can be a slow process. By using image sprites only a single image is used and by specifying the area of the image to be displayed the same image can be used multiple times.  
**For ex :** We have a an image.gif which contains the home, forward and back navigation buttons. With the help of css the user can simply specify only the part of the image that is needed. Now the user wants to only display the home part of the image for the home button.  
  
**CSS code:**

img.home  
{  
    width:50px;  
    height: 44px;  
    background: url (image.gif) 0 0 ;  
}

In the above css code only a part of the overall image.gif will be used, in this case the home page button area only. By using image sprites the page loading time can be reduced by substantial margins. The user only needs to know absolute image area to be dispalyed.

**21. What is the use of CSS sprites?**

1. A web page with large number of images takes a longer time to load. This is because each image separately sends out a http request.  
2. The concept of CSS sprite helps in reducing this loading time for a web page by combining various small images into one image. This reduces the numbers of http request and hence the loading time.

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.

|  |
| --- |
|  |
| **22. How embedded style can be linked with HTML documents?**  Embedded style is inside the HTML code only. It is written using the <Style> tag and used under the <Head> structure. It gets applied to the element for which the style will be written.  The syntax of it is as follows:  <HEAD> <STYLE TYPE="text/css"> p {text-indent: 10pt;} h1{text-color: #FFFFFF;} </STYLE> </HEAD> |
| **23. Why imported is an easy way to insert the file?**  Imported style sheet allows you to import the files which are external or combine one style sheet with another. There can be created many files, different style sheets to have different functions. Import function gives the provision to combine many elements or functionality into one. The syntax to import any file is @import notation, which is used inside the <Style> tag. There is a one rule that implies that the last imported sheet will override the previous ones.  The syntax is shown by coding as:  <Link Rel=Stylesheet Href="Main.Css" Type="Text/Css"> <STYLETYPE="text=css"> <!-- @import url(http://www.careerride.css); @import url(http://www.carrerride.main.css); .... your code --> </STYLE>  The <!-- --> is used as a comment for those browsers that doesn’t support css. |
| **24. How do I combine multiple sheets into one?**  Multiple sheets can be combined into by using the <link> tag and the with the title attribute. The title value allows one or more <link> tags to link with each other. After combination that theme will be applied as combined and will be shown to the user.  The syntax of it will be as follows:  <link rel= “text/css” href="default.css" title="combined"> <link rel= “text/css” href="style.css" title="combined"> <link rel= “text/css” href="para.css" title="combined">  Another way to combine the style sheets is the use of import which can be used in the <style> tag and the syntax can be given as follows:  @import url(site\_url); |
| **25. What are the rules in CSS ruleset?**  CSS consists of two types of CSS rules, first is for ruleset which identifies the style and the selector. It combines the style and the selector.  1. Ruleset is a combination of CSS rules, **For Example :** h1{text-color: 15pt;}, where this is the CSS rule. 2. Ruleset is selector + declaration **For Example :** h1 + {text-color: 15pt;} |
| **26. What is CSS selector?**  1. Basically it is a string that identifies the elements to which a particular declaration or set of declarations will apply. 2. It can also be referred to as a link between the HTML document and the style sheet. 3. It is equivalent of HTML elements.  **For example:**  A {text-indent: 12pt}  Here, the selector is A, which is called as type selector.  Class selectors are the selectors which are stand alone to a specific style that is declared. Class attribute can be used to declare the style and make an association with any HTML element.  The syntax for creation of class selector is: .classname. The name can be from a-z, A-z or in digits.  The example code is shown as below:  .head{font: 12em;}, this is a class selector <Body class= “head”> this is the class that is associated with the element </body>. |
|  |
| **27. What is contextual selector?**  Contextual selector specifies a specific occurrence of an element. It is combination of many selectors that are separated by white spaces. In this only the element that matches the specified element will be used not all the elements.  For example the syntax of it is being given as:  TD p code {color: #000000}  The element which is being defined as code will only be displayed as red as its color is being mentioned as red. But this is being done only if it occurs in the p text.  TD p code, h1, em {color: red}  The element code and em will only be displayed in red only when it occurs in h1 or p. |

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.

The contextual selector is a special type of selector that is used to select special occurrences of an element. This selector is a string of individual selectors separated by spaces. It defines a search patterns of the element where only the last element in the pattern is addressed to.  
  
**For Example:**

TD P TEXT {color: blue}

This implies that the TEXT will appear blue in color only if it occurs in the context of the element P which itself occurs in the context of TD.

P.about {color: red}

This implies that any element specified with class .about will appear in red color but only if it occurs in context to

**28. What do you understand by parent-child selector?**

Parent-child selector represents the direct relationship between parent element and child element. It is been created by using two or more (~) tilde separated selectors.  
  
**For example:**

Body ~ p {background-color: red; text: #FF00FF;}

Here the p element gets declared for a specific element and style only that element but if it has some child element then those elements will also get styled.  
  
One more example to show the parent-child relationship as:

Body ~ p ~ em {background-color: red; text: #FF00FF;}

The syntax in HTML will be written as:

<body><p><em> It will show the two colors for em and p</em></p></body>

The parent-child selector as the name suggests is used for selecting the direct descendent of a parent element. This selector is represented by using the “~” tilde symbol between two selectors.  
  
**For Example:**

body ~ li {background: blue; color: red}

This specifies that the LI element will be declared the above style rules only when the LI would be a descendant of the BODY element.

body ~ li ~ em {background: blue; color: white}

In this example the em will be specified the above properties only when it is a descendant of li which itself is a child of body. The parent child selector can be used to specify a particular element when there are multiple instances of the same element.

|  |
| --- |
| **29. What is ‘important’ declaration used in CSS?**  Important declarations are those declarations which have high weightage then normal declarations. These declarations override other declarations of less importance while executing. If suppose there are two statements from two users and both consist of important declaration then the one of the users declaration will override the another users declarations.  **For example:**  Body {background: #FF00FF !important; color: grey}  In this body background has more weight than the color. |
| **30. Is there any provision to include one or more declaration in a selector?**  There is a provision to include one or more declaration in a selector by using the semicolon as this shows the separation of the properties.  **For example:**  Selector {declaration1; declaration2} P {background: white; color: black} |
| **31. What is the use and syntax of class in CSS?**  Class is a group of attributes and properties that uniquely identify style that can be attached to any element. It also defines instances for different element to which the same style can be attached.  The following example shows the use of class in CSS: **P {color:red} :** It will display text color red in all paragraphs. This can be included with each element where the paragraph tag can be used. There can be given one style to one paragraph and another style to other paragraphs. A class may not have any association with the specific element. But any element with which the specific class is attached will have the same style.  **For Example:**  **CSS**  H1.prop1 {color: red} /\* one class of P selector \*/ H2.prop2 {color: blue} /\* another class of P selector \*/ .prop3 {color: green} /\* can be attached to any element \*/  **HTML**  <h1 class=prop1>This paragraph will be red</P> <h2 class=prop2>This paragraph will be blue</P> <p class=prop3>This paragraph will be green</P> <h3 class=prop3>This list item will be green</LI> |
| **32. How grouping happens inside a CSS?**  Grouping in CSS can be done by using the comma (,) separator in between one or more selectors that share the same style. A list of separator can be separated by using a semicolon as well.  **For Example:**  H1 {font-style: italic} p.first {font-style: italic} .name {font-style: italic}  In the above example all the css element share the same style so instead of writing like that it can be written as:  H1, p.first, .name {font-style: italic}  This reduces the size of the style sheet and also save the writing time for those sheets. |

**33. 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.

|  |
| --- |
| Pseudo elements allow the use of the part of element and not the element itself. They are applied to block level element, which is used for the complete block for which the CSS is being written. This allow the subpart of an element to be styled like paragraphs and headings.  **For example:**  selector:pseudo-element {property:value;} p: first-line {text-transform: lowercase;}  It adds the style to the first line of the code in the paragraph. |
| **34. How pseudo-classes are different from pseudo-elements?**  Pseudo classes are used to add style and special effects to some selectors which is being used inside some class.  The syntax it as follows  selector:pseudo-class {property:value;} a:link {color:#FF0000;}  pseudo classes can be combined with other classes as well. The syntax of it will be shown as:  selector.class:pseudo-class {property:value;} a.red:visited {color:#FF0000;} |
| **35. What is the use of a pseudo class? Explain with the help of an example?**  The pseudo classes are used to define the css properties of an element that does not exist otherwise. Pseudo classes enables the user to define the style properties of an element which otherwise would not have been possible. The pseudo class name is always preceded by a colon “ : ”.  Pseudo classes can be used with:  1. Elements 2. Classes 3. ID`s  Some of the common pseudo classes that are used with hyperlink are:  1. **link** defines the properties of the link that have not been visited. 2. **visited** defines the properties if the links that have been visited. 3. **hover** defines the properties of the link when the mouse cursor hovers on them.  **For Example:**  a:link {color: black;}  This will set the color of any hypelink that points to an unvisited link and will appear in black color.  **36. What does cascade and cascading order defines?**  Cascade is a method in CSS that defines the importance of particular style rules. This doesn’t allow any conflict to occur between other elements that is being defined within the same CSS. Declaration with more importance get executed first then others.  The syntax is as follows:  Selector{property: value ! important} /\* increased weight \*/ selector (property: value} /\* normal weight \*/  Cascading order is a sorting method that consists of some rules of the declarations through which it can be sorted and the conflict can be resolved which might occur between it. In this all the declarations has to be found that apply the selector and properties to a specific elements. |
| **37. What are the different sorting methods used inside the cascading order?**  Cascading order is a sorting method that allows many different sorting methods like:  **1. Sort by origin :** In this some rules will be defined like: - Normal weight in provider’s style sheet will override the style sheet rules of the user’s. - Increased weight in user’s style sheet will override normal weight of style sheet of provider.  **2. Sort by selector's specificity :** In this more specific selector will override the selectors that are less specific: **For Example:** - ID selector is the most specific one. - Contextual selectors are less specific. - So, ID selector will override the contextual selector style sheets.  **3. Sort by order specified :** If two selectors are same in the weight and other properties then the specification will be seen for overriding. **For Example:** - Style attribute used for inline style will override all other styles. - Link element used for external style will override imported style. |

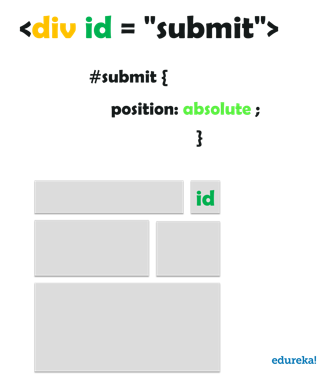
**38. Explain the concept of specificity in CSS?**

**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.

|  |
| --- |
| **39. How inline and block elements are different from each other?**  1. Inline elements don’t have line breaks. An Inline element doesn’t have elements to set width and height. **For Example:** em, strong, etc. codes are few examples of inline elements.  2. Block elements do have line breaks and they define the width by setting a container. It also allowed setting height. It can contain the elements that occur in inline elements.  **For Example:**  width and height max-width and max-height min-width and min-height hi (i=1-6)- heading element p- paragraph element. |
| **40. How to center the block-elements with CSS?**  CSS allows you to style your web-pages and sheets so that you can visualize it in better way.  This can be done in two ways: 1. It can be done by defining the properties like margin-left and right to auto and width can be given any value. **For Example:**  body {width: 30em; background: cyan;} h1 {width: 22em; margin-left: auto; margin-right: auto}  2. It can be done by the use of table like:  Table {margin-left: auto; margin-right: auto; width: 400px;}  This table width is being defined by the content used. These are the methods that are used to center the block element.  **41. How does inheritance work in CSS?**  Inheritance is a concept that is used in HTML and other languages but it is used in CSS as well to define the hierarchy of the element from top level to bottom level. In inheritance child will inherit the properties of parent. In this the restriction is being applied that not all the properties can be applied. Inheritance passes its properties to its children class so that the same property doesn’t have to define the same property. Inherited properties can be overrided by the children class if child uses the same name properties.  **For Example:**  Body {font-size: 10pt;}  And another definition is being defined in the child class  Body {font-size: 10pt;} H1 {font-size: 14pt;}  All the paragraph text will be displayed using the property defined in the body except for the h1 style which will show the text in font 14 only. |
| **42. What are the advantages of the external over inline style methods?**  1. External Style Sheets are useful as it keeps the style and content separately and doesn’t allow it to mix with each other. It can control the styles for multiple documents, whereas inline style mixes the content with the style and make the code messier. 2. External style sheet allows the creation of various classes in a structured way, whereas inline style sheet can’t create or control class elements. 3. External style sheet can use selector and grouping methods to apply styles, whereas inline styles can’t use selector and grouping methods. |
| **43. How do you override the underlining of hyperlinks?**  The concept of override the underlining of hyperlinks in CSS is done by using control statements and using external style sheet.  **For Example:**  A { text-decoration: none; }  Suppose this is being written in CSS file and in the anchor tag in HTML the format is being written as:  <a href="career.html" style="text-decoration: none">link text</a>  So, anything thing written as inline will override the style for the hyperlink used in external style sheet. |
|  |
| Using CSS extra style functionality can be provided, that can control the status of underlining of hyperlinks. You can use style sheets in whatever way you want and can control the look and style of your display. The way in which it can be done is by introducing external level style sheet: A { text-decoration: none } //within an anchor element as: <a HREF="example.htm" STYLE="text-decoration: none">link text</a>  **44. What will happen if we will use floats across the page with 100% width?**  In CSS if float declaration is being made then it add some width in the form of border and it allow it to float even more. So, the width can be used just by adding the border of say 1 pixel in CSS element. |
| **45. What is the Difference between id and class?**  The id and class is being used in HTML and includes the values from CSS. The difference is as follows:  1. Id is an attribute that it uniquely assigns a name to a particular element, whereas class defines an element with certain set of properties which can be used for a complete block. 2. Id can be used for an element and uniquely identifies it, whereas class is used as block element and applied to many tags where it is used. 3. Id restricts the use of properties to one specific element, whereas class applies the rules and properties to specific block or group of elements. |

**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.

|  |
| --- |
| **Class - css interview questions - edureka**  1. Class selector can be given to an overall block. This is sometimes termed as block element as well, whereas ID selector identifies a unique name and a style for a specific element. 2. ID selector declares the style for only one particular element which can be differentiated from other element, whereas Class selector is being given for the whole complete block.  Example of ID selector is being written like this:  #idname {color: red; background: black;}  This can be used in HTML as  <P ID= “idname”>this element only will be affected by the use of this ID element</P>  **46. How to restore the default property value using CSS?**  CSS doesn’t provide any default values that can be used if the user wants to revert back to the old values. The only option which can be performed to restore the default property value is to re-declare the property which you want to revert back to. Rules should be defined for using the selectors like tag name, etc. that you want to override for more specific values. |
| **47. What is the purpose of Nesting Selectors?**  Nesting selectors are the selectors that define a selector inside another selector.  **For Example:**  H1 {color:blue; text-align:center;} .marked {background-color:red;} .marked h1 {color:white;}  In this a particular style is being defined for the element of h1, and another is given for all other elements. Here the class is given as “marked” and one more style given for h1 element within the h1 style. |
| **48. With the help of examples explain grouping and nesting of css selectors?**  Grouping selectors : In css the designer can reduce the code by simply groping together selectors with the same property values.  For Example:  h1 {color: green;} h2 {color: green;} p {color: green;}  As you can see from the above code the all the elements have the same property. To avoid rewriting the same code again and again the user can simply group the selectors by separating each selector with a comma. Grouped selectors :  h1,h2,p {color: green;}  Nesting selectors : This enables the user to specify a style for a selector within a selector. For Example:  p{ color:blue; text-align:center; } .marked { background-color:red; } .marked p { color:white; }  In the above example separate properties are assigned for p and the .marked class. But the last value that is .marked p implies that the property will apply to p elements with class defined as .marked. |

**49. What are the different Media Types included in CSS?**

Media types are not case sensitive and it allows you to define different properties in different media.  
  
The different media type that is included in CSS is:  
  
**1. Aural –** Used for speech and sound synthesizers  
**2. Print –** Used to show the content as they will look when you will use the print command  
**3. Projection -** Used to show the CSS for the projectors  
**4. Handheld -** Used for handheld devices  
<="" b="" style="padding: 0px; margin: 0px;">Used for screens like computers and laptop

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;

 }

}

Media is used to render the design and customize the documents. Media types allow user to load and apply their own selected style sheets. Media control is applied to external style sheets, in this way user can save time by retrieving the sheets from the network itself. Example: Speech based browsers can avoid the download of style sheets which are designed for visual rendering.

**50. What is the use of Media Types in CSS?**

Media types in CSS define the media like audio and video to be used in your HTML document to represent the properties in a better way. The font property can be used for media types as it can be used for print media or screen media. Document requires a defined media to represent the screen that can be read on the paper. It is used as:@media

<html>  
<head>  
<style>  
@media screen  
{  
    p.test {font-family:verdana,sans-serif;font-size:14px;}  
}  
@media print  
{  
    p.test {font-family:times,serif;font-size:10px;}  
}  
</style>  
</head>  
<body>  
----------Your code here----------  
</body>  
</html>

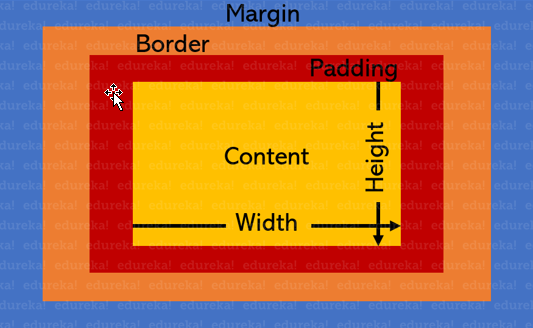
**51. Why CSS Box Model is used? What are the elements that it includes?**

CSS box model defines the design and layout of the styling element. It is a box that shows the elements that will come before others. Like, it consists of margins, borders, padding, and the actual content. It allows a structured way to show the elements in comparison to other elements.  
  
The terms that are used in box model is shown as:  
  
**1. Margin –** It shows the top most layer and it shows the overall structure which consists of border and other elements that is being shown below.  
**2. Border –** It shows the padding and content option and keep a border around it. Border can be affected by background color.  
**3. Padding –** It shows the space between the content blocks. It is affected by the background color.  
**4. Content –**. It shows the actual content like text and images. These all are essential terms that allows you to understand the structure and apply it to your code.

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.

CSS uses the concept of a box model which implies that every HTML element is a box. This term is used when we are talking about design and layout. The CSS box model is actually a box that wraps an HTML element.  
  
A box comprises of the following components:  
  
**1. margins :** Used to clear an area around a border.  
**2. border :** Border goes around the padding and the content.  
**3. padding :** Used to clear the area around the content.  
**4. content :** This contains the actual content of the box that is the text and the images.  
  
It is important to note that when a user sets the height and width of an element, they are doing so only for the content area. To know the fill size of the element the user must also specify the other layers i.e. the padding border and margins.  
  
**For Example:**

p  
{  
    width:220px;  
    padding:10px;  
    border:5px solid gray;  
    margin:0px;  
}

The total width of an element is calculated like this:

Total element width = width + left padding + right padding + left border + right border + left margin + right margin

The total height of an element is calculated like this:

Total element height = height + top padding + bottom padding + top border + bottom border + top margin + bottom margin

**52. Write the css code showing the usage of all the background properties?**

With the help of css there are several ways to change the background properties of an html page. For ex. the following code will place the background image at the center of the body element. Also the background would not be repeated and the scroll would be locked.

BODY  
{  
    font-family : "Verdana, Arial, Helvetica, sans-serif;  
    background-image: url(images/background\_image.gif);  
    background-repeat: no-repeat;  
    background-position: center;  
    background-attachment: fixed;  
    background-color: #FF00FF;  
    color : #FF0000;  
    margin: 20px;  
}

**If background and colour should always be set together, why they exist as separate properties because;**

The reasons for this are as follows:  
  
1. It increases the legibility of the style sheets. The background property is a complex property in CSS. If it is combined with color, the complexity will further increase.  
2. Color is inherited, but background isn’t. This can further increase the confusion.

**53. 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;  
}

|  |
| --- |
|  |
| **54. How can the same properties be defined for different elements without repeating them again and again? Explain with an example?**  In order to specify the same style properties for multiple elements in css we can make use of class. Class in general are a group of of instances of the same elements to which the same style can be assigned or it can be a group of different instances to which a common style can be assigned.  **For Example:**  **CSS code**  P.class1 {color: red} /\* defines a class of P selector \*/ P.class2 {color: blue} /\* defines another class of P selector \*/ .class3 {color: green} /\*this class can be used with any element\*/  **HTML code**  <P class=class1>This paragraph will be colored in red</P> <P class=class2>This paragraph will be colored in blue</P> <P class=class3>This paragraph will be shown in green</P> <LI class=class3>This list item will also be green</LI> |
|  |
| **55. Why should inline styles be used carefully?**  Inline style rules allows a user to attach individual styles to individual elements in an HTML document. The syntax of an inline style rule attached to an HTML element:  <p style=”color: red;”> Red text</p>  Inline styles are generally not the preferred way of applying a style rule and must be used carefully because of the following reasons:  1. The style rules can easily get mixed up in an HTML page code. 2. The entire rule must be specified in the value of the style attribute, this can be tough in case of long rules. 3. By using inline styles the user is unable to take advantage of CSS properties such as grouping selectors and re usability of a style sheet. 4. It is generally advised to use either internal or external style sheets as they are more convenient to manage and can be changed easily as they are at the same place. |

**56. 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.

**57. How do hexadecimal color codes differ from RGB values?**

In CSS a user can specify a color in two ways:  
  
**1. Hexadecimal color codes :** These as their name suggest use six characters to represent a color. It is a combination of numbers and letters. When you use a hexadecimal code to specify a color it should always be precede by a “#”. This ensures that the color is displayed correctly in a browser.  
**Example :**

p {color: #0000FF;}

**2. RGB color values :** Every color can be defined as a mixture of red, green and blue. In CSS the user can also define the value of a color by specifying the RGB values in two different ways:  
  
**- rgb(r,g,b) :** In this type the values can be in between the integers 0 and 255.  
**- rgb(r%,g%,b%) :** The rgb basically represent the percentage of red, green and blue of the color.  
  
The hexadecimal values of a color are the ones that are most widely used.

**58. How is a CSS executed in case of more than one conflicting rule?**

If there are two or more css rules which are pointing towards the same element there are some basic rules that a browser follows in order to know which is the most specific and will be executed / applied.  
  
**For Example:**

p {color: red;}  
p {color: blue;}

In the above case all the p contents would appear blue as this rule was specified the last.  
In css every element is assigned a unique specificity value. Based on the specificity value a style rule is executed.  
  
**For Example:**

p : this has the specificity of 1  
div p : this has a specificity of 2 ( ie 1 + 1 of two selectors )  
.class : has a specificity of 10  
#id : This has a specificity of 100.

In this way whichever selector has the highest specificity will apply in case of multiple selectors pointing at the same element.

**59. Explain with the help of examples how the display property is used in CSS?**

In CSS the user has the ability to define how the content will be displayed. There are three fundamental types of display:  
  
**1. Inline :** The elements are displayed in a line.  
**2. Block :** In this display style a line break is placed before and after every element.  
**3. None :** Does not display the elements.  
  
The syntax of specifying a display style is :

h1 { display: inline; }

CSS provides the user with further more display styles such as:  
  
**1. list-item :** Does the same work as the HTML li element. Displays the content in a list.  
**2. marker :** This property is exclusively used with the pseudo elements :before and :after.  
**3. Tables :** The table properties can be used in similar way to the table elements provided by HTML but css further extends the functionality by providing properties like table-column, table-caption etc.  
  
**For Example:**

div.test  
{  
    display:inline;  
    zoom:1.0;  
}

|  |
| --- |
| **60. How can HTML elements be styled having specific attributes?**  Css allows the user to style the html elements that have specific attributes. It does not only rely on class and id.  **For Example:**  [title] { color:red; }  This will simply color any attribute containing title. Css also allows the user to specify an attribute with a particular value. **For Example:**  [title=test] { color:red; }  This will simply color the text test that appears anywhere in the title attribute. Also the user can specify an attribute with multiple values. **For Example:**  [title~=test] {color:blue;} |
| **61. What are the different provision provided in css to define the dimension of an element?**  In css the user can choose from multiple dimension properties to style an element. The list of css dimension properties are:  **1. height :** This property allows the user to specify the height of a specific element. **2. max-height :** This allows the user to set the maximum height of an element. **3. max-width :** This specifies the maximum width of an element. **4. min-height :** It allows the user to specify the minimum height of an element. **5. min-width :** Used to set the minimum width of an element. **6. width :** This property is used to set the width of an element.  **For Example:**  img.big {height:100px} <!-- specifies the height of an element--> img.big {max-height:100px} <!-- specifies the maximum height of an element--> |
|  |

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

CSS float is useful when you don’t have to give width for an element or you don’t want to keep the element fixed. It allows the elements to be given left or right boundaries to expand and wrap all other elements. Float is basically used for images and for layouts. Elements in this can be made float horizontally i.e. left or right.  
  
**For Example:**

img{float:right;}

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;

|  |
| --- |
| **63. How is the float property implemented in css?**  The css float property allows an element to be pushed to the left or right which allows other elements to wrap around the floated element. The elements specified before the float element will not be effected. Only the elements specified after the floated elements will float around the element.  **For Example:**  img { float:right;}  This implies that if an image is floated towards the right then the text that follows its would flow around it on the left. The user can also float multiple elements together. The elements would float if there is space available for them to float. Any float element will cause the other elements following it to float around it, to avoid this the user can make use of the clear property. The clear property is used to specify the sides of an element on which the floating of elements is not allowed.  **For Example:**  .line {clear:both;}  This would prevent the elements from floating around after the float element. |
| **64. 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.   **65. What is the purpose of the z-index? Explain with the help of an example.**  While using css to position html elements they may overlap each other. The z index i used to specify which element overlaps which element. The z-index is a number which can either have a positive or a negative value. By default the z-index value is zero. In case elements have the same z-index number specified then the browser will layer them according to the order in which they appear in the HTML.  **For Example:**  We have a list of 4 elements each with their defined numbers: element1 - z-index number 25 element2 - z-index number -34 element3 - z-index number 10 element4 - z-index number not defined In this case the order of their stack would be: element1 element3 element4 element2 |
| **66. Explain the meaning of graceful degradation in reference to CSS?**  Generally graceful degradation is a concept that allows a system to continue to operate properly in the event of a failure of a component. In web design the graceful degradation is a very important area. When a developer creates a website he creates it to take advantage of the latest browser support etc. but care should also be taken to render the website properly on older browsers. In this way the designer is able to get a wider audience by stepping down some of the features to provide basic functionality to people with older browsers. For example while specifying an image in the html css code many time an alt tag is used. This means that in case the image cannot be shown in a browser it will instead show the text specified within the alt tag. |
| **67. What is the other alternative to graceful degradation?**  The other concept is know as progressive enhancement. Progressive enhancement focuses on the content of a web rather than the browsers itself. In this way a website can be viewed on different platforms according to the amount of resources available. For ex a user with the latest browser and a high bandwidth connection might get some extra eye candy as compared to a user visiting the same site on a dial-up and old browser. But the overall functionality provided would be the same. Gmail does so as well where users with slow connections are provided a plain html view whereas high bandwidth users get the complete site to access. This concept recently has come into play as mobile devices with Internet surfing capabilities have started to grow and expand its user base. |
| **68. How are shorthand properties are used in css? Give examples.**  One of the primary advantages of css is that it greatly reduces page load times. Writing multiple properties and their values in a single line of css code is known as css shorthand technique. One thing to be kept in mind is that while using the css shorthand technique is that the order of specifying the values of an attribute must be maintained. In case the user wants to keep a value as default it is not needed to be mentioned.  **For Example:**  margin-top: 5em; margin-bottom: 3em; margin-right: 1em; margin-left: 1em Becomes: margin: 5em 1em 3em (top, right and left, bottom);  border-width:5px; border-style:solid; border-color:#fff; Becomes: border:5px solid #fff; |

|  |
| --- |
|  |
|  |
| **69. What is .rolloverfordnd?** .rolloverfordnd is a class selector to which the style gets applied. Dot(.) is used for class selector as it applies to any HTML elements that have the attribute as: class="rollverfordnd". This attribute can be added to multiple elements such that, if separate declarations are defined then the declaration with a specific selector win over any other.  **70. Why my div are of different size in IE?** The specification which is been given for the size of div in IE is as follows: Width of a box (i.e. a containing area) in CSS = width + border +padding (+margin). Total box width is: 300px with a padding of 10px and a border of 5px. This requires the total area of 330px. IE5 uses different rules and out of width it subtracts the padding and border from the specific width so the total width becomes 270px. So, instead of 330px, IE uses 270px that is why the div are of different sizes. |
| **71. Can someone else's Style Sheet be used without their permission?** Style sheet information is written using special language syntax. Yes, someone else’s style sheet can be used without their permission, but it is not recommended, as it might overload the server. So, a local copy should be maintained, created and used instead of referencing a remote copy.  **72. Which style specification method should be used?** Style specification method in your document should be used according to your need and requirements. The methods that can be used are: - External Style Sheets: should be used when you want to apply the same style to multiple documents. - Embedded Style Sheets: should be used if you want to specify styles for a single document. - Inline Styles: should be used if you want to apply style to one or few elements in a document.  **73. What are logical and physical tags?** - Physical tags are known as presentational markup. They are used with recent versions of HTML. It suggests the visual appearance of the content. Ex: Italics, Underline, Bold - Logical tags are not used for appearances, but they are used to show the purpose of the content written. For example em tag is used for emphasis not for any other thing. |
| **74. How the concept of Style sheet is different from HTML?** HTML is an easy way to represent the structure, but when it comes to style then HTML becomes very hard to work with. The concept of style sheet is introduced to make the work easy for you to design without any hassles. Style sheets increases browser capabilities and allow different formatting options which have not been included in HTML. |

**75. 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

**76. 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;

}

**77. 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

**78. 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) |

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

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

h1 {

background-repeat: none;

 }

**80. 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.

**81. 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;

**82. 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;

**83. 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

**84. 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

**85. 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;

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

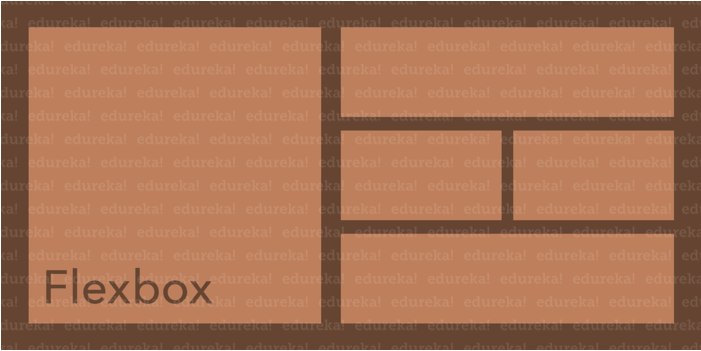
Use the border-radius property to add rounded corners to an image. 50% will make the image circular.

border-radius: 50%;

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

**87. 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.

**88. 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.

**89. 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.

**90. What are the different ways to position a certain element in CSS?**

The position property specifies the type of positioning method used for an element.

There are five different position values:

position: fixed;

position: static;

position: absolute;

position: sticky;

position: relative;

Elements are then positioned using the top, bottom, left, and right properties. However, these properties will not work unless the position property is set first. They also work differently depending on the position value.

**91. What is Block Formatting Context? How does it work?**

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. 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).

**92. What effect would this piece of CSS code have? {box-sizing: border-box;}**

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

**93. What is a CSS pre-processor? When do you recommend a pre-processor be used in a project?**

A CSS preprocessor is a program that lets you generate CSS from the preprocessor’s own unique syntax. There are many CSS preprocessors to choose from, however, most CSS preprocessors will add some features that don’t exist in pure CSS, such as mixin, nesting selector, inheritance selector, and so on. These features make the CSS structure more readable and easier to maintain.

The usage depends on the type of project but the following advantages/disadvantages come with a preprocessor.

**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 with webpack, 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.

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.

**94. 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 the 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.

**95. What are Vendor-Prefixes?**

Browser vendors sometimes add prefixes to experimental or nonstandard CSS properties and [JavaScript](https://www.edureka.co/blog/what-is-javascript/) APIs, so developers can experiment with new ideas while—in theory—preventing their experiments from being relied upon and then breaking web developers’ code during the standardization process. Developers should wait to include the unprefixed property until browser behaviour is standardized.

The major browsers use the following prefixes:

* -webkit- (Chrome, Safari, newer versions of Opera, almost all iOS browsers (including Firefox for iOS); basically, any WebKit based browser)
* -moz- (Firefox)
* -o- (Old, pre-WebKit, versions of Opera)
* -ms- (Internet Explorer and Microsoft Edge)

**96. Give an example of how you would use counter-increment and counter-reset in CSS to create automatic numbering within a webpage?**

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) " ";

}

**97. What is file splitting? When is it used?**

Part of a good CSS architecture is file organization. A monolithic file is fine for solo developers or very small projects. For large projects—sites with multiple layouts and content types, or multiple brands under the same design umbrella—it’s smarter to use a modular approach and split your CSS across multiple files.

Splitting your CSS across files makes it easier to parcel tasks out to teams. One developer can work on typography-related styles, while another can focus on developing grid components. Teams can split work sensibly and increase overall productivity.

Here’s a dummy css structure:

* reset.css: reset and normalization styles; minimal color, border, or font-related declarations
* typography.css: font faces, weights, line heights, sizes, and styles for headings and body text
* layouts.css: styles that manage page layouts and segments, including grids
* forms.css: styles for form controls and labels
* lists.css: list-specific styles
* tables.css: table-specific styles
* carousel.css: styles required for carousel components
* accordion.css: styles for accordion components

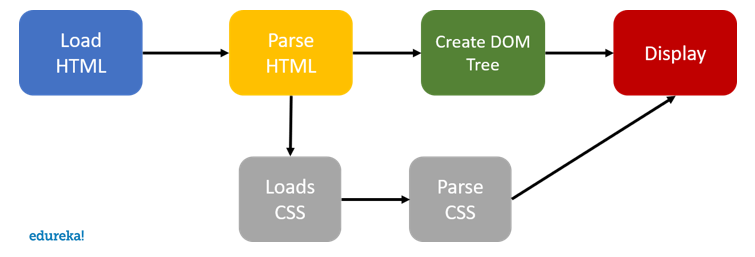
**98. What are *functions/mixins*?**

Functions are blocks of code that return a single value of any Sass data type.  A **mixin** is very similar to a function. The main difference between the two is that mixins output lines of Sass code that will compile directly into CSS styles, while functions return a value that can then become the value for a CSS property or become something that might be passed to another function or mixin.

**99. How does CSS work under the hood?**

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.



**100. Recommend a way to optimize a certain webpage for prints?**

The secret to creating printable pages is being able to identify and control the “content area(s)” of your website. Most websites are composed of a header, footer, sidebars/sub-navigation, and one main content area. Control the content area and most of your work is done. The following are my tips to conquering the print media without changing the integrity of your website.

* Create a stylesheet for print
* Avoid unnecessary HTML tables
* Know which portions of the page don’t have any print value
* Use page breaks
* Size your page for print – max height etc