**CSS and CSS 3**

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

**Some benefits of using CSS include:**

* Easier to maintain and update
* Greater consistency in design
* More formatting options
* Lightweight code
* Faster download times
* Search engine optimization benefits
* Ease of presenting different styles to different viewers
* Greater accessibility
* Better website speed
* Time-saving
* Better device compatibility
* **What are the disadvantages of CSS?**
* CSS, CSS 1 up to CSS3, result in creating of confusion among  web browsers.
* With CSS, what works with one browser might not always work with another. The web developers need to test for compatibility, running the program across multiple browsers.
* There exists a scarcity of security.
* After making the changes we need to confirm the compatibility if they appear. The similar change affects on all the browsers.
* The programming language world is complicated for non-developers and beginners. Different levels of CSS i.e. CSS, CSS 2, CSS 3 are often quite confusing.
* Browser compatibility (some styles sheet are supported and some are not).
* CSS works differently on different browsers. IE and Opera supports CSS as different logic.
* There might be cross-browser issues while using CSS.
* There are multiple levels which creates confusion for non-developers and beginners.
* **What is the difference between CSS2 and CSS3?**

|  |  |  |
| --- | --- | --- |
| S.No. | CSS | CSS3 |
| 1 | CSS is capable of positioning texts and objects. | On the other hand, CSS3 is capable of making the web page more attractive and takes less time to create. CSS3 is backward compatible with CSS. |
| 2 | Responsive designing is not supported in CSS | CSS3 is the latest version, hence it supports responsive design. |
| 3 | CSS cannot be split into modules. | Whereas CSS3 can be breakdown into modules. |
| 4 | Using CSS, we cannot build 3D animation and transformation. | But in CSS3 we can perform all kinds of animation and transformations as it supports animation and 3D transformations. |
| 5 | CSS is very slow as compared to CSS3 | Whereas CSS3 is faster than CSS. |
| 6 | In CSS we have set of standard colors and it uses basic color schemes only. | Whereas CSS3 has a good collection of HSL RGBA, HSLA, and gradient colors. |
| 7 | In CSS we can only use single text blocks. | But in CSS3 we can use multi-column text blocks |
| 8 | CSS does not support media queries. | But CSS3 supports media queries |
| 9 | CSS codes are not supported by all types of modern browsers. | Being the latest version, CSS3 codes are supported by all modern browsers. |
| 10 | In CSS, designers have to manually develop rounded gradients and corners. | But CSS3 provides advanced codes for setting rounded gradients and corners |
| 11 | There is no special effect like shadowing text, text animation, etc. in CSS. The animation was coded in jQuery and JavaScript. | CSS3 has many advance features like text shadows, visual effects, and a wide range of font styles and colors. |
| 12 | In CSS, the user can add background colors to list items and lists, set images for the list items, etc. | Whereas CSS3 list has a special *display* property defined in it. Even list items also have counter reset properties. |
| 13 | CSS was developed in 1996. | CSS3 is the latest version of CSS and was released in 2005. |
| 14 | CSS is memory intensive. | CSS3 memory consumption is low as compared to CSS. |

* **Name a few CSS style components**

1. **CSS Custom Properties**
2. **Pre-built variations**
3. **Shadow parts**
4. **User forced**

* **What do you understand by CSS opacity?**
* The **opacity** CSS property sets the opacity of an element. Opacity is the degree to which content behind an element is hidden, and is the opposite of transparency
* opacity applies to the element as a whole, including its contents, even though the value is not inherited by child elements. Thus, the element and its children all have the same opacity relative to the element's background, even if they have different opacities relative to one another.
* Using opacity with a value other than 1 places the element in a new [stacking context](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_positioned_layout/Understanding_z-index/Stacking_context).
* When opacity value is set to 0, the element and all of its children are not visible; however, they still register [pointer events](https://developer.mozilla.org/en-US/docs/Web/API/Pointer_events). This can be controlled with the CSS [pointer-events](https://developer.mozilla.org/en-US/docs/Web/CSS/pointer-events) property.
* To change the opacity of a background only, use the [background](https://developer.mozilla.org/en-US/docs/Web/CSS/background) property with a [color value](https://developer.mozilla.org/en-US/docs/Web/CSS/color_value) that allows for an alpha channel
* **How can the background color of an element be changed?**
* the background color of an HTML element using the background-color CSS property and giving it a value of a color.

**p {**

**background-color: pink;**

**}**

* **How can image repetition of the backup be controlled ?**
* The **background-repeat property** in CSS is used to repeat the background image both horizontally and vertically.
* It also decides whether the background image will be repeated or not.

**Syntax:**

background-repeat: repeat|repeat-x|repeat-y|no-repeat|initial|inherit;

* **What is the use of the background-position property?**
* The background-position [CSS](https://developer.mozilla.org/en-US/docs/Web/CSS) property sets the initial position for each background image.
* The position is relative to the position layer set by [background-origin](https://developer.mozilla.org/en-US/docs/Web/CSS/background-origin).

[**Syntax**](https://developer.mozilla.org/en-US/docs/Web/CSS/background-position#syntax)

/\* Keyword values \*/

background-position: top;

background-position: bottom;

background-position: left;

background-position: right;

background-position: center;

/\* <percentage> values \*/

background-position: 25% 75%;

/\* <length> values \*/

background-position: 0 0;

background-position: 1cm 2cm;

background-position: 10ch 8em;

/\* Multiple images \*/

background-position:

0 0,

center;

/\* Edge offsets values \*/

background-position: bottom 10px right 20px;

background-position: right 3em bottom 10px;

background-position: bottom 10px right;

background-position: top right 10px;

/\* Global values \*/

background-position: inherit;

background-position: initial;

background-position: revert;

background-position: revert-layer;

background-position: unset;

* **Which property controls the image scroll in the background?**
* The background image can be set using the [background-image property](https://www.geeksforgeeks.org/css-background-image-property/) that is used to set one or more background images for an element.
* By default, it places the image in the top left corner.
* To specify two or more images, we need to specify the separate URLs with a comma for both images.

**Syntax:**

element\_selector {

background-image: url('url')|none|initial|inherit;

}

* **Why should background and color be used as separate properties?**
* 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.  
  Color is inherited, but background isn’t.
* This can further increase the confusion.
* **How to center block elements using CSS1?**
* The margin can control the position of the block element both horizontally and vertically.
* To center them, we can adjust the margin property such that it is placed in the center.
* **Center block elements using margin property:**We need to specify the margin from left and right such that it looks centered.
* We do not need to do this manually, we have one property value “auto” which will automatically set the margin such that our block element is placed in the center. Use the below CSS property to center your block element.
* **How to maintain the CSS specifications?**

CSS properties should be implemented by browser vendors along with detailed algorithms, code samples and tabular information.

**The Specification also include:**

* The syntax and data types of the language
* Detailed explanation on CSS Selectors
* How you can assign values to properties
* The Cascade (the "C" in CSS)

How inheritance works

The Box Model e.t.c

Explanation on some of these topic are short and easy to understand while others are explained in great detail.

The Specification also specify how stylesheets can be included in your web document and how to target specific media e.g print or screen.

The CSS Specification prior to CSS3 was a single Specification, CSS3 on the other hand is divided into Modules which are Independent Specifications that can be worked on by different author(s) at different paces, that's why we have Selector Level 3 Specification, CSS Color 4, CSS Backgrounds and so on. Some of these modules are revisions of CSS2.1, and some are newly created, but all fall under the banner of CSS3.

The Specification should be your guide if you need to understand how a specific property or feature works behind the scene and how it works with other CSS properties. And if you are comfortable reading algorithms you won't get bored reading the CSS Specification.

[The Specifications can be accessed online](https://drafts.csswg.org/) along with the author(s) name and when it was last updated.

Take your time and familiarize yourself with the Specification because we will be making heavy use of it in the topic: CSS Selectors.

* **What are the ways to integrate CSS as a web page?**
* CSS is used for styling the look and formatting of a document written in HTML. There are three ways to add CSS in HTML
* **Inline CSS**:

Styles added directly to the HTML element.

**<html lang="en">**

**<head>**

**<title>Browser</title>**

**</head>**

**<body>**

**<h1>This is h1</h1>**

**<p>This paragraph doesn't have CSS.</p>**

**<p style="color:red">This paragraph is styled with inline CSS.</p>**

**</body>**

**</html>**

* **Internal CSS**:

Styles defined at the head section of the document.

**<head>**

**<style>**

**p {**

**color: red;**

**}**

**</style>**

**</head>**

* **External CSS**:

Styles defined in a separate file.

**<head>**

**<style>**

**p {**

**color: red;**

**}**

**</style>**

**<title>Browser</title>**

**</head>**

* **What are embedded style sheets?**
* **Cascading Style Sheets** (**CSS**) is a [stylesheet](https://developer.mozilla.org/en-US/docs/Web/API/StyleSheet) language used to describe the presentation of a document written in [HTML](https://developer.mozilla.org/en-US/docs/Web/HTML) or [XML](https://developer.mozilla.org/en-US/docs/Web/XML/XML_introduction) CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.

[**Key resources**](https://developer.mozilla.org/en-US/docs/Web/CSS#key_resources)

* [CSS Introduction](https://developer.mozilla.org/en-US/docs/Web/CSS#css_introduction)

If you're new to web development, be sure to read our [CSS basics](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/CSS_basics) article to learn what CSS is and how to use it.

* [CSS Tutorials](https://developer.mozilla.org/en-US/docs/Web/CSS#css_tutorials)

Our [CSS learning area](https://developer.mozilla.org/en-US/docs/Learn/CSS) contains a wealth of tutorials to take you from beginner level to proficiency, covering all the fundamentals.

* [CSS Reference](https://developer.mozilla.org/en-US/docs/Web/CSS#css_reference)

Our [exhaustive CSS reference](https://developer.mozilla.org/en-US/docs/Web/CSS/Reference) for seasoned Web developers describes every property and concept of CSS.

* **What are the external style sheets?**
* An external style sheet is a separate file where you can declare all the styles that you want to use on your website.
* You then link to the external style sheet from all your HTML pages.
* This means you only need to set the styles for each element once.
* If you want to update the style of your website, you only need to do it in one place
* **What are the advantages and disadvantages of using external style sheets?**

**The advantages of External Style Sheets are**

Using them, the styles of multiple documents can be controlled from one.  
 Classes can be created for use on multiple HTML element types in many documents.  
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.

* **What is the meaning of the CSS selector?**
* selectors are used to target the [HTML](https://developer.mozilla.org/en-US/docs/Glossary/HTML) elements on our web pages that we want to style.
* There are a wide variety of CSS selectors available, allowing for fine-grained precision when selecting elements to style.
* In this article and its sub-articles we'll run through the different types in great detail, seeing how they work
* A CSS selector is the first part of a CSS Rule.
* It is a pattern of elements and other terms that tell the browser which HTML elements should be selected to have the CSS property values inside the rule applied to them. The element or elements which are selected by the selector are referred to as the subject of the selector.
* **What are the media types allowed by CSS?**
* width and height of the viewport
* width and height of the device
* orientation (is the tablet/phone in landscape or portrait mode?)
* resolution

**@media screen and (min-width: 480px) {  
  #leftsidebar {width: 200px; float: left;}  
  #main {margin-left: 216px;}  
}**

* **What is the rule set?**
* A CSS ruleset is various affirmations to various pieces or elements of the document.
* The objective is to apply a bunch of properties for certain distinct qualities to a solitary, or a particular arrangement of components in the connected HTML page.
* **Create Layouts**

**.header {  
  background-color: #F1F1F1;  
  text-align: center;  
  padding: 20px;  
}**