

Bottom of Form

Follow Us

* [HTML5](https://www.tutorialstonight.com/html/html-introduction.php)
* [CSS3](https://www.tutorialstonight.com/css/css-introduction.php)
* [Javascript](https://www.tutorialstonight.com/js/)
* [Bootstrap 4](https://www.tutorialstonight.com/bootstrap/bootstrap-introduction.php)
* [Python](https://www.tutorialstonight.com/python/python-tutorial.php)
* [Articles](https://www.tutorialstonight.com/password-generator-in-javascript.php)

CSS Tutorial

[CSS Introduction](https://www.tutorialstonight.com/css/css-introduction.php) [CSS Implementation](https://www.tutorialstonight.com/css/css-implementation.php) [CSS Syntax](https://www.tutorialstonight.com/css/css-syntax.php) [CSS Selector](https://www.tutorialstonight.com/css/css-selectors.php) [CSS Class](https://www.tutorialstonight.com/css/css-class.php) [CSS Id](https://www.tutorialstonight.com/css/css-id.php) [CSS Color](https://www.tutorialstonight.com/css/css-color.php) [CSS Background](https://www.tutorialstonight.com/css/css-background.php) [CSS Units](https://www.tutorialstonight.com/css/css-units.php) [CSS fonts](https://www.tutorialstonight.com/css/css-font.php) [CSS text](https://www.tutorialstonight.com/css/css-text.php) [CSS Links](https://www.tutorialstonight.com/css/css-links.php) [CSS Lists](https://www.tutorialstonight.com/css/css-lists.php) [CSS Tables](https://www.tutorialstonight.com/css/css-tables.php) [CSS Height /width](https://www.tutorialstonight.com/css/css-height-and-width.php)

CSS box model

[CSS Box Model](https://www.tutorialstonight.com/css/css-box-model.php) [CSS Borders](https://www.tutorialstonight.com/css/css-borders.php) [CSS Margins](https://www.tutorialstonight.com/css/css-margins.php) [CSS Padding](https://www.tutorialstonight.com/css/css-padding.php)

CSS advanced

[CSS Outline](https://www.tutorialstonight.com/css/css-outline.php) [CSS Overflow](https://www.tutorialstonight.com/css/css-overflow.php) [CSS Cursor](https://www.tutorialstonight.com/css/css-cursor.php) [CSS Position](https://www.tutorialstonight.com/css/css-position.php) [CSS Display](https://www.tutorialstonight.com/css/css-display.php) [CSS shadow](https://www.tutorialstonight.com/css/css-shadow.php) [CSS Float](https://www.tutorialstonight.com/css/css-float.php) [CSS Align](https://www.tutorialstonight.com/css/css-align.php) [CSS Icons](https://www.tutorialstonight.com/css/css-icons.php) [CSS Buttons](https://www.tutorialstonight.com/css/css-button.php) [CSS Form](https://www.tutorialstonight.com/css/css-form.php) [CSS Combinators](https://www.tutorialstonight.com/css/css-combinators.php) [CSS Specificity](https://www.tutorialstonight.com/css/css-specificity.php) [CSS 2D transforms](https://www.tutorialstonight.com/css/css-2D-transformation.php) [CSS 3D transforms](https://www.tutorialstonight.com/css/css-3D-transformation.php) [CSS transition](https://www.tutorialstonight.com/css/css-transition.php) [CSS Animation](https://www.tutorialstonight.com/css/css-animation.php) [CSS variable](https://www.tutorialstonight.com/css/css-variable.php) [CSS Pagination](https://www.tutorialstonight.com/css/css-pagination.php) [CSS flexbox](https://www.tutorialstonight.com/css/css-flexbox.php) [CSS Media Queries](https://www.tutorialstonight.com/css/css-media-queries.php)

CSS MISCELLANEOUS

[How to center a div in CSS](https://www.tutorialstonight.com/how-to-center-a-div-in-css.php) [How to center an image in CSS](https://www.tutorialstonight.com/how-to-center-an-image-in-css.php)

[Description: Description: Description: Description: Ezoic](https://www.ezoic.com/what-is-ezoic/)report this ad

**CSS Tutorial - An Introduction To CSS**

[❮ Home](https://www.tutorialstonight.com/) [Next ❯](https://www.tutorialstonight.com/css/css-implementation.php)

What are the 2 basic things that we need to design a static website?

It's the [HTML](https://www.tutorialstonight.com/html/html-introduction.php) and CSS. HTML creates the structure of a website and CSS gives shapes, designs, and colors to the website.

In this CSS tutorial you are going to learn all about CSS, its use, what it can do, why do we need it, and how to use it. You will learn how to create a website with CSS and how to use it to create a website that looks good and works well.

**What is CSS?**

CSS stands for **Cascading Style Sheets**. CSS is a language that describes the style of an HTML document.

CSS describes how HTML elements should be displayed its size, its color, its position, its orientation all are decided by CSS.

CSS is a stylesheet language, a language for describing the presentation of a document written in a markup language.

CSS has an old history and its current version is called **CSS3**. CSS3 is a combination of the old version + some new specifications like a media query, namespace, selector, etc.

It works with [HTML](https://www.tutorialstonight.com/html/html-introduction.php) files so before learning CSS you must be knowing HTML.

Here is how a CSS code looks like.

body {

margin: 0;

}

p { /\* Element \*/

font-size: 20px;

color: #000;

}

.btn { /\* Button class \*/

background-color: #3bcece;

color: #fff;

padding: 10px;

border-radius: 5px;

}

CSS

**How to use CSS in HTML**

You can use CSS with HTML in 3 different ways:

1. Inline CSS - CSS is written inside HTML tags.
2. Internal CSS - CSS is written inside HTML files within the <style> tags.
3. External CSS - CSS is written in a separate file and linked to the HTML file.

CSS lets you embed its code directly in HTML elements or within an HTML file or even let you create an external CSS file just for CSS code which you can link to single or multiple HTML files.

Learn in detail [how to use CSS in HTML](https://www.tutorialstonight.com/how-to-use-css-in-html.php).

**See What Difference CSS Can Make**

Following is an example of a webpage. Given below HTML code for the webpage without any CSS.

Example

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>CSS - Introduction</title>

</head>

<body>

<div class="container">

<div class="first">

<h1>Hello, World!</h1>

<p>The most famous line for a programmer.</p>

</div>

<div class="box">

<h2>Learning CSS</h2>

<p>Learning CSS will let you design beautiful webpages, moving objects, amazing animations and different shapes on your webpage.</p>

</div>

<button>Start Learn CSS With Tutorials Tonight!</button>

</div>

</body>

</html>

HTML

[▶ Play Code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=webpage-without-css)

Output:

Now let's add some CSS and make it look cool. See how the look of the webpage changed.

Example

body {

margin: 0;

font-family: sans-serif;

}

.container {

background-color: #ffe8cb;

border: 5px solid #a87569;

}

h1 {

font-size: 40px;

text-align: center;

}

.first p {

margin-left: 20px;

}

.box {

padding: 20px;

background-color: #c49185;

}

button {

display: block;

font-size: 18px;

margin: 20px auto;

padding: 10px 20px;

border-radius: 25px;

background: linear-gradient(to right, #a87569, #ffe8cb);

border: 1px solid #a87569;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=adding-css-to-webpage)

Output:

You can see how CSS code changes the look of the webpage. Well, this is just a glimpse of what CSS can do, you will be able to do even more after you learn it.

**Note:** If you don't understand CSS code currently then don't bother we will look at all properties in detail in the coming sections.

**What Can CSS Do?**

HTML was designed only for describing the content of the web page not for formatting it.

When formatting tags like <font> and attributes like the color were introduced in HTML 3.2 then it becomes a nightmare for web developers to manage large websites. They had to place font and color information on every page and that looked so overwhelming.

To solve this problem, the **World Wide Web Consortium (W3C) created CSS**, which was able to design websites more beautifully and was clean. It removed the style formatting from the HTML page.

Here is a list of a few things that CSS can do:

* Changing font properties, like size, color, font-family, font-weight, etc

Adding color to text elements

* Adding background properties like background color, background image, etc
* Creating different shapes
* Adding margin and padding to the elements
* Designing layouts
* Creating effects like animation, hover, etc

**Why you should learn CSS?**

If you want to become a **web developer** then CSS is among the 3 necessary languages (HTML, CSS, [JavaScript](https://www.tutorialstonight.com/js/)) that you must learn.

Here are some key points why you should learn CSS:

* **Web development**: It is a must-learn language for web development
* **Unique design**: Using this you will be able to design unique looking websites
* **Access over elements**: It gives you access to almost every element that is visible on the screen
* **Control**: You can style any element according to your wish
* **Adding effects**: It can add effects on web pages that look fabulous
* **Create animations**: Using CSS you can create animations like moving objects or shapes
* **Responsive websites**: Learning CSS will let you design responsive websites that fit any device sizes
* **Learning frameworks**: If you learn and understand it then you be able to learn and work on many CSS based frameworks like [Bootstrap](https://www.tutorialstonight.com/bootstrap/bootstrap-introduction.php)

**History of CSS**

CSS was first proposed on 10th Oct 1994 by Håckon Wieum Lie. In that year Microsoft's Internet Explorer 3 was released with very limited support of CSS.

In those days browsers were unable to add complete CSS and also had bugs. It took more than 3 years when any browser reached full implementation of CSS.

It has multiple versions and each version is built upon the previous version, typically adding new features. Newer versions are a subset of one or more levels of CSS built for a particular device and user interface.

* **CSS 1** - First CSS specification was published on 17th Dec 1996.
* **CSS 2** - Developed by W3C and published in May 1998.
* **CSS 2.1** - CSS level 2 revision 1, W3C published it on 7th June 1998.
* **CSS 3** - Earliest draft of CSS3 was published in June 1999.

**Advantages of CSS**

CSS provides powerful control over HTML documents. It is also easy to learn and understand.

There are a lot of advantages of learning CSS:

* **Easy to maintain** - If you want to change any kind of styling just go to styling, change, and all your elements will adjust according to the new change.
* **Handle multiple webpages**: CSS saves lots of time when used externally because it can control multiple page layouts at once.
* **Faster page loading** - Before CSS tags like color, border, size, font, etc were repeated many times which led to slower loading. But by using CSS we need to mention property and value just once.
* **High Control** - We can control the style of the whole website using a single CSS page. Change just once here and the whole website shows your changes.
* **Multiple devices friendly** - Using CSS we can create pages that can adjust their decorations according to the device.

**Example Of CSS Code**

Here is a working example of HTML + CSS.

Example

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>CSS code example</title>

<style>

body {

margin: 0;

--space-height: 300px;

--ball-size: 80px;

}

.space {

width: 100vw;

height: var(--space-height)

}

.ball {

width: var(--ball-size);

height: var(--ball-size);

border-radius: 50%;

background: #b6b0b2;

margin: 20px;

animation: rolling 3s infinite linear;

}

@keyframes rolling {

0% {

transform: translate(0, calc(var(--space-height) / 2 - var(--ball-size) / 2));

}

25% {

transform: translate(calc(50vw - 40px), calc(var(--space-height) - var(--ball-size)));

}

50% {

transform: translate(calc(100vw - 80px), calc(var(--space-height) / 2 - var(--ball-size) / 2));

}

75% {

transform: translate(calc(50vw - 40px), 0);

}

100% {

transform: translate(0, calc(var(--space-height) / 2 - var(--ball-size) / 2));

}

}

button {

display: block;

background: #c90557;

color: #fff;

border: none;

padding: 10px 20px;

width: 95%;

margin: 0 auto;

border-radius: 5px;

cursor: pointer;

font-size: large;

transition: all 0.3s ease-in-out;

}

button:hover {

background: #940340;

}

</style>

</head>

<body>

<div class="space">

<p class="ball"></p>

</div>

<button>Hover me!</button>

</body>

</html>

HTML

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=example-of-css-code)

**Conclusion**

CSS is a powerful tool to control the style of the webpages. It is easy to learn and understand. It is also easy to maintain. It can be used to control multiple pages at once. It can be even be used to control the whole website at once.

Top of Form



Bottom of Form

Top of Form

# HOW TO INCLUDE CSS IN HTML

In this tutorial, we will learn how to include CSS in HTML files with working examples and demonstrations.

To beautify an HTML file you need to use CSS but how will you include CSS code in your HTML file or how will you connect another CSS file to your HTML file.

There are 3 CSS implementation methods:

1. [External CSS](https://www.tutorialstonight.com/css/css-implementation.php" \l "external-css)
2. [Internal CSS](https://www.tutorialstonight.com/css/css-implementation.php" \l "internal)
3. [Inline CSS](https://www.tutorialstonight.com/css/css-implementation.php" \l "inline)

## 1. How to include external CSS file in HTML

To write CSS code you can make a separate CSS file and then link it with your HTML file (recommended way to include CSS). This separate file is called an **external stylesheet**.

To connect an external stylesheet with your HTML file use <link> element.

The <link> element is used in the head section of the HTML file.

To connect a CSS file give the URL of the stylesheet to the href attribute and set rel="stylesheet".

Suppose your file name is "external.css" that lies in the same folder as your HTML file then you can write <link rel="stylesheet" href="external.css">.

<head>

<link rel="stylesheet" href="external.css">

</head>

HTML

[▶ Run the code](https://www.tutorialstonight.com/free-online-html-editor.php?file=implimentation-external" \t "_blank)

Write your CSS code in an external file and save it with a .css extension.

This external CSS file should purely contain CSS code.

body {

background-color: #182C61;

}

h2 {

color: #CAD3C8;

}

p {

font-size: 20px;

color: white;

}

CSS

### Including multiple external stylesheet

You can add multiple external style sheet to your HTML file using multiple <link> elements.

CSS properties of both external stylesheets are added to the HTML file.

If there are 2 different values for the same CSS property for any element then priority will be given to that stylesheet which below in the head element.

<head>

<link rel="stylesheet" href="external-1.css">

<link rel="stylesheet" href="external-2.css">

<link rel="stylesheet" href="external-3.css">

</head>

HTML

### Benefits of using external style

External style is used by most websites and has lots of benefits, Some of which are mentioned below:

* **Clean**: Since you, CSS classes and properties are stored in a separate file so your HTML file looks clean.
* **Management**: Change at a single property can reflect all over the webpage.
* **Easy to manage**: Easy to maintain bigger websites. If you have 100s of the webpage that use the same CSS codes then it is a nice move to use the external stylesheet.
* **Faster webpage**: The webpage loads quicker once the CSS file has been cached.
* **Can change all page at once**: If you have 100s of the webpage and you want some change in your design, then you would not want to go to each page and change. Just change in external stylesheet at 1 place and you are done.

## 2. Internal CSS in HTML

Another way to include CSS in an HTML file is internal CSS.

The CSS is written in the head section of the HTML file inside <style> element.

The Internal style is mainly used when you need some other CSS property for the web page other than external. The internal style is used for a single page style within the page.

Internal style makes webpage load slower and also makes HTML documents larger.

<head>

<style>

body {

background-color: #182C61;

}

h2 {

color: #CAD3C8;

}

p {

font-size: 20px;

color: white;

}

</style>

</head>

HTML

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=implimentation-internal" \t "_blank)

## 3. inline CSS in HTML

Inline style is used to add a unique style to a single element.

The inline style has the highest priority among internal and external CSS. Inline style overwrites external or internal CSS.

The inline element is used by elements by assigning style attributes to them and defining CSS property inside single-quotes or double-quotes. Example <p style="color: blue;">Hello World!</p>.

<body style="background-color:#182C61">

<h2 style="color:#CAD3C8">Learning How to add inline CSS.</h2>

<p style="color:white;font-size:20px">Added inline styles to the elements.</p>

</body>

HTML

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=implimentation-inline" \t "_blank)

## Order of Cascading

In CSS, when there is 2 or more than 2 CSS property for an element and all have the same [specificity](https://www.tutorialstonight.com/css/css-specificity.php" \t "_blank) then the style rule that comes last in the file will be used.

* If in an external stylesheet there are such rules than the last rule in the stylesheet will be used.
* If there are 2 different external stylesheets that have such rules than the rule of the stylesheet that is linked kast in the head section is used.

Points to remember:

1. There are 3 ways to include CSS in HTML file: external, internal and inline CSS.
2. Multiple stylesheets can be added to a single HTML file.
3. The inline style has the highest priority, it overwrites external and internal CSS.

**CSS SYNTAX**

In this CSS tutorial, we will discuss the basic syntax of CSS and how to write CSS code.

**CSS Syntax**

The CSS syntax consists of some set of rules that defines how to writes CSS code correctly so that the browser engine may correctly identify specific CSS features of the page.

CSS ruleset have 3 parts:

1. Selector - selectors select elements from the HTML file and tells which element is to be styled.
2. CSS property - property defines the CSS feature that you want to use, like color, padding, border, etc.
3. CSS value - each CSS property has a set of values in which you can choose one or set some numerical value with units, like 10px, center, auto, etc.

Note: CSS property and CSS value are separated by a colon (:).

A single set of property and value is called a **declaration**. If there are multiple sets of CSS declaration then you need to separate them by a semicolon (;)

The last semicolon of a ruleset is optional.

selector {

property1: value1;

property2: value2;

property3: value3;

}

CSS

In the example below there are 3 declarations and 1 ruleset.

h1 {

color: cyan;

padding: 15px;

background: silver;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=css-syntax" \t "_blank)

**Grouping selectors in CSS**

In case you want to apply the same style for multiple selectors then it is not necessary to define CSS rulesets individually for each selector.

To write the same CSS ruleset for multiple selectors simply write CSS once and **separate the selectors by a comma** as shown in the example below.

h1, h2, h3 {

color: #964915;

margin: 10px;

padding: 5px;

background: silver;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=syntax-group" \t "_blank)

Note: If you are unable to understand the code above then don't bother we will look at each of the above property in the coming chapters.

**CSS Property with multiple values**

Some of the CSS properties have multiple sets of values that are mentioned at a time. Example: border, shadow, font property etc.

You can give values to these properties separated by spaces.

p {

border: 2px solid teal;

box-shadow: 3px 3px 5px 5px silver;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=syntax-multi" \t "_blank)

Points to remember:

1. CSS is written in form of rulesets.
2. Each ruleset has 3 different parts selector, CSS property and CSS value.
3. Multiple selectors can be separated using a comma.

**CSS SELECTORS**

[❮ Prev](https://www.tutorialstonight.com/css/css-syntax.php) [Next ❯](https://www.tutorialstonight.com/css/css-class.php)

Before applying CSS on HTML elements we must first learn how to select the element first. In this tutorial, we will learn about what is CSS selectors and different CSS selectors type.

**Introducing CSS selectors**

The CSS selector is used to select an HTML element in the webpage to style.

It is part of the CSS ruleset. CSS provides us with various ways to select an element. Some of the ways are mentioned below.

As you can see in the above image CSS selector target single or multiple HTML elements that match the selection.

When the selector is matched with HTML elements then CSS declarations defined in the block is applied to those HTML elements.

Note: In CSS there is no selector or [combinator](https://www.tutorialstonight.com/css/css-combinators.php) to select parent, sibling of parent elements and child of parent sibling.

**Types of CSS selectors**

There are several different types of CSS selectors. We will discuss here few main and most commonly used selectors.

1. [Universal selector CSS](https://www.tutorialstonight.com/css/css-selectors.php#universal)
2. [Element selector](https://www.tutorialstonight.com/css/css-selectors.php#element)
3. [Class selector CSS](https://www.tutorialstonight.com/css/css-selectors.php#class)
4. [Id selector CSS](https://www.tutorialstonight.com/css/css-selectors.php#id)
5. [Group selector](https://www.tutorialstonight.com/css/css-selectors.php#group)
6. [Attribute selector](https://www.tutorialstonight.com/css/css-selectors.php#attribute)

**1. Universal selector CSS**

The universal selector (\*) selects all the elements on the webpage. Many developers use this selector to remove all the default margins and padding.

Universal selector starts with an asterisk symbol (\*).

\* {

margin: 0;

padding: 0;

color: steelblue;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=selector-universal)

The CSS code above in the example selects all the elements in the webpage and remove all the [margins](https://www.tutorialstonight.com/css/css-margins.php), [padding](https://www.tutorialstonight.com/css/css-padding.php) and set color to texts.

**2. Element selector**

If you want to select all the same element in the webpage then use an element selector.

The element selector selects all the elements in the webpage on the basis of its name. To select the element write the element name and provide CSS properties in the block.

p {

color: blue;

text-align: center;

}

a {

color: red;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=selector-element)

In the above example there are 2 element selectors, 1st select all the paragraph in the webpage and 2nd select all the anchor texts.

**3. CSS class selector**

The class selector is used to select the element which has a class attribute. It is the most commonly used selector in CSS.

It is written with a period(.) followed by the class name and the CSS code declaration is given in curly braces.

To select any HTML element just give the class attribute to the element with the name of the class you want to select.

.box{

padding: 10px;

border: 1px solid blue;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=selector-class)

You can add multiple classes to a single element. The element will reflex all CSS declarations of added class.

.class1 {

color: red;

}

.class2 {

font-size: 25px;

}

.class3 {

text-align: center;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=selector-multiclass)

**4. Id selector CSS**

The id selector is used to select a unique targeted element.

The id attribute of the element is used to target the element.

To select an element with id use a hash (#) symbol with id name.

#user {

color: red;

font-size: 20px;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=selector-id)

**5. Group selector**

You can group multiple same or different types of selectors for the same style definition.

Suppose we have the same CSS declarations for the multiple-element as shown in the example below.

p {

color : red;

background-color : yellow;

text-align : center;

}

h1 {

color : red;

background-color : yellow;

text-align : center;

}

h3 {

color : red;

background-color : yellow;

text-align : center;

}

CSS

Instead of defining CSS property like as above we can use group selector.In group selector we define single CSS code and selector is separated with comma.See the example below:

p,

h1,

h3 {

color : red;

background-color : yellow;

text-align : center;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=selector-group)

You can also use different selectors together as a group selector on an element.

p,

.heading,

#tag {

color: red;

background-color: yellow;

text-align: center;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=selector-group-id)

**6. CSS attribute selector**

The attribute selector selects the element in the webpage based on their attribute or by the value of the given attribute.

To select an element based on an attribute write the attribute name in a square bracket and give its value in quotes [attribute="value"].

[title="my title"] {

color: red;

background-color: yellow;

text-align: center;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=attribute-selector)

You can also select all elements with a given attribute regardless of their value.

[title] {

color: red;

background-color: yellow;

text-align: center;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=attribute-selector-all)

Points to remember:

1. CSS selector is used to select elements in the webpage to apply CSS ruleset on it.
2. The most commonly used selector is the **class selector**.

**CSS CLASS SELECTOR**

[❮ Prev](https://www.tutorialstonight.com/css/css-selectors.php) [Next ❯](https://www.tutorialstonight.com/css/css-id.php)

The CSS classes allow us to define multiple CSS properties for a specific class and when that class is added to any element then all the specified CSS property is reflected in the HTML element.

Classes are defined either in the head section within the style tag or in the external stylesheet.

Once a class is defined it can be used as many times you want to use.

**Defining a class**

A class defined by a **class\_name** preceded by a dot **(".")** .

Syntax:

.class\_name { property: value; }

Defining class in head section:

.class1{

background-color: #182C61;

margin: 10px; /\* There should be no space between value and unit \*/

font-size: 20px;

}

.class2{

color: #CAD3C8;

padding: 25px;

}

CSS

Defining classes in external stylesheet.

.class1{

background-color: #182C61;

margin: 10px; /\* There should be no space between value and unit \*/

font-size: 20px;

}

.class2{

color: #CAD3C8;

padding: 25px;

}

CSS

**Adding class to the element**

A class is added to an element using class attribute. Like shown below.

<p class="class1">This is a paragraph with single class.</p>

HTML

Multiple classes can be added to an element. like shown below.

<p class="class1 class2 class3">This is a paragraph with multiple class.</p>

HTML

A working example for CSS class.

<!DOCTYPE html>

<html>

<head>

<style>

.class1{

background-color: #182C61;

color: white;

font-size: 20px;

padding: 15px;

width: 100%;

}

</style>

</head>

<body>

<h2>Learning about CSS classes.</p>

<p class="class1">This paragraph is added with class1.</p>

<p class="class1">This paragraph is added with class1.</p>

<p class="class1">This paragraph is added with class1.</p>

</body>

</html>

HTML

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=css-class)

Output:

This paragraph is added with class1.

This paragraph is added with class1.

This paragraph is added with class1.

**CSS ID SELECTOR**

[❮ Prev](https://www.tutorialstonight.com/css/css-class.php) [Next ❯](https://www.tutorialstonight.com/css/css-color.php)

CSS id selector is used to select an element using the id attribute.

An id is used for many purposes like

* selecting element using javascript for various purposes
* creating page link
* Adding CSS property to the element

Here in this section we will mainly focus on CSS implementation of id.

**CSS Id**

Same as CSS class we can use CSS id to add style to the element it is defined to. From CSS point of view there is no difference between class and id.

Unlike CSS class id can't be used more than once in any webpage.

An id is defined by **id\_name** preceded by a hash symbol ("**#**") .

#id\_name { property: value; }

#id1{

background-color: #182C61;

margin: 10px; /\* There should be no space between value and unit \*/

font-size: 20px;

}

#id2{

color: #CAD3C8;

padding: 25px;

}

CSS

**Adding Id to an element**

An id is added to an element by using the id attribute. As shown below

<p id="id1">This is a paragraph with an id.</p>

HTML

<!DOCTYPE html>

<html>

<head>

<style>

#id1{

background-color: #182C61;

color: white;

font-size: 20px;

padding: 15px;

width: 100%;

}

</style>

</head>

<body>

<h2>Learning about CSS id.</p>

<p id="id1">This paragraph is added with id1.</p>

</body>

</html>

HTML

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=css-id)

Output:

This paragraph is added with id1.

# CSS COLOR

[❮ Prev](https://www.tutorialstonight.com/css/css-id.php) [Next ❯](https://www.tutorialstonight.com/css/css-background.php)

Color values in CSS are used to specify colors to HTML elements. Generally this property is used to specify color of the text.

Example:

Red

Green

Blue

Gold

Sienna

.box {

border : 3px solid red;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=css-color" \t "_blank)

## Defining CSS color

CSS provides us various methods by which we can define color values.

The three main methods to define colors in CSS are:

1. [RGB value](https://www.tutorialstonight.com/css/css-color.php" \l "rgb)
2. [HEX value](https://www.tutorialstonight.com/css/css-color.php" \l "hex)
3. [HSL value](https://www.tutorialstonight.com/css/css-color.php" \l "hsl)

### 1).RGB value

One of the ways to define color in CSS is RGB value.color value can be specified using **rgb()** property.

rgb(red, green,blue)

**rgb()** property is defined by setting three numerical values separated by commas in this property. Each define value for red,green and blue.

Example:

<h2 style="color:rgb(200,20,200)">You are learning RGB values.</h2>

<p style="color:rgb(200,20,200)">i am colored using RGB value.</p>

HTML

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=color-rgb" \t "_blank)

### 2).HEX value

Another way to define color in CSS is HEX value.color in HEX() is defined using the hexadecimal value.

Color is defined as **#rrggbb()** where rr represent color red,gg represent color green and bb represent color blue.

Example:

<h2 style="color:#b93bca">You are learning HEX values.</h2>

<p style="color:#b93bca">i am colored using HEX value.</p>

HTML

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=color-hex" \t "_blank)

### 3).HSL value

It is also a way to define color in CSS.color value can be specified using **hsl()** property.

**hsl()** property is defined by setting three values separated by commas in this property.

hsl(hue,saturation,lightness)

In **hsl()** property hue defines color. Its degree on color wheel 0 is red,120 is green and 240 is blue. Saturation is a shade of grey in percentage value 0% means grey shade 100% means full color. lightness is also a percentage value where 0% is black,50% is nothing and 100% is white.

Example:

<h2 style="color:hsl(120,120%,120%)">You are learning HSL values.</h2>

<p style="color:rgb(120,120%,120%)">i am colored using HSL value.</p>

HTML

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=color-hsl" \t "_blank)

**CSS BACKGROUND**

[❮ Prev](https://www.tutorialstonight.com/css/css-color.php) [Next ❯](https://www.tutorialstonight.com/css/css-units.php)

CSS background properties are used to style background of HTML element.There are 5 different CSS background property:

1. [CSS background color](https://www.tutorialstonight.com/css/css-background.php" \l "color)
2. [CSS background image](https://www.tutorialstonight.com/css/css-background.php" \l "image)
3. [background repeat](https://www.tutorialstonight.com/css/css-background.php" \l "repeat)
4. [background attachment](https://www.tutorialstonight.com/css/css-background.php" \l "attachment)
5. [CSS background position](https://www.tutorialstonight.com/css/css-background.php" \l "position)

**1).CSS background color**

background-color is a CSS property that is used to define the background color of any HTML element.You can specify color value in any way like rgb,hex,hsl or color-name itself.

h2{

background-color:cyan;

}

p{

background-color:Green;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=background-color" \t "_blank)

**2).CSS background image**

background-image is a CSS property that is used to define background color of any HTML element.Path of image is given using url() property.

background-image:url('path\_of\_image')

body{

background-image:url('background-image.jpg');

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=background-image" \t "_blank)

**3).background repeat**

background-image repeats itself both horizontally and vertically by default.So either to avoid repetition or to direct repetition in one direction we use background-repeat property.

We use background-repeat: repeat-x; to repeat background-image in x direction.

body{

background-image:url('background-image2.jpg');

background-repeat:repeat-x;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=background-repeatx" \t "_blank)

We use background-repeat: repeat-y; to repeat background-image in y direction.

body{

background-image:url('background-image2.jpg');

background-repeat:repeat-y;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=background-repeaty" \t "_blank)

We use background-repeat: no-repeat; for background-image not to repeat in any direction.

body{

background-image:url('background-image2.jpg');

background-repeat:no-repeat;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=background-norepeat" \t "_blank)

**4).background attachment**

This property specifies whether the background-image will scroll or be fixed relative to the webpage.

body{

background-image:url('background-image2.jpg');

background-attachment:scroll;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=background-attachment" \t "_blank)

**5).CSS background position**

This property define initial position of background-image on the web page.By default image starts from top-left cornor.

We can have background-position values like top,bottom,left,right,center etc.

body{

background-image:url('background-image2.jpg');

background-repeat:no-repeat;

background-position:center;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=background-position" \t "_blank)

Bottom of Form

**CSS UNITS**

[❮ Prev](https://www.tutorialstonight.com/css/css-background.php) [Next ❯](https://www.tutorialstonight.com/css/css-font.php)

CSS has several measurement units for length and size. These measuring units are used to specify a non-zero value to the CSS properties.

CSS units are used where we need to define size of length. EX - height,width,border-size,padding,font-size etc.

CSS unit values are specified by a numerical value followed by a unit and there should be no space between number and unit.  
Ex:-

* height: 25 px; is WRONG WAY there should be no space between value and unit
* height: 25px; is RIGHT WAY

There are two types of length unit:

1. [Absolute unit](https://www.tutorialstonight.com/css/css-units.php" \l "absolute)
2. [Relative unit](https://www.tutorialstonight.com/css/css-units.php" \l "relative)

**1).Absolute unit**

The absolute length unit is of fixed length. It shows exactly specified size on any screen.

Device to device screen size varies so absolute length units are not recommended. It is dependent on an output medium. It can be used when the desired output is known.

These are absolute units with their description:

| **Units** | **Description** |
| --- | --- |
| cm | 1cm = 1 centimeter (standard length unit) |
| mm | 1mm(millimeter) = 1/10 centimeter |
| in | 1in(inch) = 2.54 centimeter |
| pt | 1pt(point) = 1/72 inch (0.3527mm) |
| px | 1px(pixel) = 0.75pt |
| pc | 1pc(picas) = 12pt |

h1{

margin: 1cm; /\*centimeter\*/

}

h2{

font-size: 5mm; /\*millimeter\*/

}

h3{

margin: 1in; /\*inch\*/

}

h4{

font-size: 10pt; /\*point\*/

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=units-absolute" \t "_blank)

**2).Relative unit**

The relative unit length specifies length according to another length property. It is flexible with the device to device and better to use in different size devices. It changes with different rendering medium.

| **Units** | **Description** |
| --- | --- |
| em | It is relative to the current font size of the element.1em means the same size of font 2em means 2 times of current font-size. |
| ex | It is relative to the current x-height of the font. |
| vh | It is related to the height of viewport.1vh=1% of the height of the viewport. |
| vw | It is related to the width of viewport.1vh=1% of the width of the viewport. |
| % | It is relative to the font size of the parent element. |

.box{

width: 60vw;

border: 1px solid silver;

background-color: #00669956;

}

.em-font{

font-size: 2em;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=units-relative" \t "_blank)

Output:

This paragraph has 2 times its regular font.

**This box has 60% width of the viewport.** This box has 60% width of the viewport. This box has 60% width of the viewport. This box has 60% width of the viewport.

**CSS FONTS**

[❮ Prev](https://www.tutorialstonight.com/css/css-units.php) [Next ❯](https://www.tutorialstonight.com/css/css-text.php)

To make HTML documents (webpage) more readable and easy to overview we have different size, forms, shapes and different other visible properties of texts.

We can provide texts of these different properties using **CSS font properties**.

The CSS font properties are as follows:

1. [CSS font family](https://www.tutorialstonight.com/css/css-font.php#family)
2. [CSS font size](https://www.tutorialstonight.com/css/css-font.php#size)
3. [CSS font style](https://www.tutorialstonight.com/css/css-font.php#style)
4. [CSS font-weight](https://www.tutorialstonight.com/css/css-font.php#weight)
5. [CSS font-variant](https://www.tutorialstonight.com/css/css-font.php#variant)

**1).CSS font family**

It is divide into two parts:

* Generic family: This family includes **Monospace**, **serif** and **sans-serif**.
* Font family: This includes font family like **Monospace**, **serif** and **arial**, **Monospace**, **serif** and **New Times Roman** etc.

**sans-Serif**

* Arial, sans-serif:
* Helvetica, sans-serif:
* Gill sans, sans-serif:
* Lucida, sans-serif:
* Helvetica Narrow, sans-serif
* sans-serif

**Serif**

* Times, serif:
* Times New Roman, serif:
* Polatino, serif:
* Bookman, serif:
* New Century Schoolbook, serif
* serif

Using CSS property font-family we specify family to the HTML elements.

h1 {

font-family:"sans-serif";

}

h2 {

font-family:"monospace";

}

p {

font-family:"Times New Roman";

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=font-family)

Output:

This is shown in sans-serif.

This is shown in monospace.

This is shown in Times New Roman.

**2).CSS font size**

CSS font-size property is used to set font size.

Using this property we can resize the font choosing any scale like px,em,%(percentage) etc.

h1 {

font-size:40px;

}

h2 {

font-size:1.8em;

}

p {

font-size:120%;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=font-size)

**3).CSS font style**

CSS font-style property is used to set the font style of an HTML document.

Using this property we can choose font style like normal,*oblique* or *italic*.

* normal - font is normal.
* oblique - font is little leaning.
* italic - font is italic.oblique and italic looks similar.

h2 {

font-style:normal;

}

h3 {

font-style:oblique;

}

p {

font-style:italic;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=font-style)

**4).CSS font weight**

CSS font-weight property is used to define weight of font and manages the boldness of font.

font-weight ranges from 100 to 900 and three more values i.e bold,bolder and lighter.

<h2 style="font-variant:normal">This heading has normal font-variant property.</h2>

<p style="font-variant:small-caps">This line has small-caps font-variant property.</p>

HTML

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=font-weight)

**5).CSS font variant**

CSS font-variant property is used to define whether or not text be displayed in capital letters.

font-variant:small-caps makes small text capital but those tests which convert to capital are smaller in size than those of original capital text present in paragraph.

<h2 style="font-variant:normal">This heading has normal font-variant property.</h2>

<p style="font-variant:small-caps">This Paragraph Has Small-caps Font-variant Property.</p>

HTML

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=font-varient)

**CSS TEXT**

[❮ Prev](https://www.tutorialstonight.com/css/css-font.php) [Next ❯](https://www.tutorialstonight.com/css/css-links.php)

CSS provides us the ability to style the text using its **text formatting properties**.

Some of its text formatting properties are :

1. [CSS text color](https://www.tutorialstonight.com/css/css-text.php" \l "color)
2. [CSS text align](https://www.tutorialstonight.com/css/css-text.php" \l "align)
3. [CSS text shadow](https://www.tutorialstonight.com/css/css-text.php" \l "shadow)
4. [CSS text direction](https://www.tutorialstonight.com/css/css-text.php" \l "direction)
5. [CSS text indent](https://www.tutorialstonight.com/css/css-text.php" \l "indent)
6. [CSS text decoration](https://www.tutorialstonight.com/css/css-text.php" \l "decoration)
7. [CSS text transform](https://www.tutorialstonight.com/css/css-text.php" \l "transform)
8. [CSS word spacing](https://www.tutorialstonight.com/css/css-text.php" \l "word_spacing)
9. [CSS letter spacing](https://www.tutorialstonight.com/css/css-text.php" \l "letter_spacing)
10. [CSS white space](https://www.tutorialstonight.com/css/css-text.php" \l "white_space)

**1).CSS Text Color**

Using color property in CSS we can set the color of text of any HTML element.

Color values can be specified either by:

* Color name - like green,blue,silver etc.
* RGB value - like rgb(240,20,43).
* HEX value - like #aacb00.
* HSL value - like hsl(210,50%,60%).

h2 {

color: red;

}

h3 {

color: rgb(237, 10, 245);

}

h4 {

color: #808b17;

}

p {

color: hsl(220, 50%, 60%);

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=text-color" \t "_blank)

**2).CSS Text align**

text-align property in CSS is used to align text horizontally on the webpage.

Value of text-align can be left,right and center.

<p style="text-align:left">This paragraph is left aligned.</p>

<p style="text-alignt:center">This paragraph is in center.</p>

<p style="text-align:right">This paragraph is right aligned.</p>

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=text-align" \t "_blank)

**3).CSS Text shadow**

text-shadow property is used to create shadow effect on the texts.

Value of text-shadow has 4 parts of value.Example: ***text-shadow : 8px 5px 10px magenta;***

In the value of text-shadow first part specify value of horizontal shift of text-shadow,second part specify value of vertical shift of text-shadow,third value specify blur radius and last value specify color of shadow.

<h2 style="text-shadow:8px 5px 10px magenta">This paragraph had shadow property.</h2>

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=text-shadow" \t "_blank)

**4).CSS Text direction**

direction property is used to set direction of text of the element.

<h4 style="direction: rtl">This paragraph right to left aligned.</h4>

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=text-direction" \t "_blank)

**5).CSS Text indent**

text-indent property is used to set indentation of first line of any paragraph.

<p style="text-indent: 40px">This paragraph has an indentation of 40px. first line of this paragraph will have an indentation of 40px and other line will start from left.</p>

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=text-indent" \t "_blank)

**6).CSS Text decoration**

text-decoration property is used to decorate text by creating underline,overline,line-through or none.

It is used to remove the underline from any link.

<p style="text-decoration:underline">This paragraph is underlined.</p>

<p style="text-decoration:line-through">This paragraph has a line-through.</p>

<p style="text-decoration:overline">This paragraph has an overline.</p>

<p style="text-decoration:none">This paragraph is not decorated.</p>

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=text-decoration" \t "_blank)

**7).CSS Text transform**

The text-transform property is used to transform the text either in uppercase or in lowercase.

Using this property one can convert uppercase into lowercase and lowercase into uppercase also the first letter of the word can be capitalised.

<p style="text-transform:capitalise">This paragraph is capitalised.</p>

<p style="text-transform:lowercase">Text of this paragraph is changed to lowercase.</p>

<p style="text-transform:uppercase">Text of this paragraph is changed to uppercase.</p>

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=text-transform" \t "_blank)

**8).CSS Word spacing**

The word-spacing property is used to set the spaces between the words in an HTML document.

<p style="word-spacing:15px">This paragraph has 15px word spacing</p>

<p style="word-spacing:-5px">Text of this paragraph has -5px word spacing.</p>

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=text-wordspacing" \t "_blank)

**9).CSS Letter spacing**

The letter-spacing property is used to set the spaces between the letters in an HTML document.

<p style="letter-spacing:6px">This paragraph has 5px letter spacing</p>

<p style="letter-spacing:-2px">Text of this paragraph has -3px letter spacing.</p>

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=text-letterspacing" \t "_blank)

**10).CSS White space**

The white-space property shows how white space inside an element is handled.

p.nowrap{

white-space:nowrap;

}

p.wrap{

white-space:wrap;

}

p.pre{

white-space:pre;

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=text-whitespace" \t "_blank)

**CSS LINKS**

[❮ Prev](https://www.tutorialstonight.com/css/css-text.php) [Next ❯](https://www.tutorialstonight.com/css/css-lists.php)

CSS let us style hyper link.Hyper links can be styled with the following specifications which shows their state:

1. [a:link](https://www.tutorialstonight.com/css/css-links.php" \l "link) This specify a normal unvisited link
2. [a:visited](https://www.tutorialstonight.com/css/css-links.php" \l "visited) This specify a visited link
3. [a:hover](https://www.tutorialstonight.com/css/css-links.php" \l "hover) This specify a state of link when mouse pointer is over the link
4. [a:active](https://www.tutorialstonight.com/css/css-links.php" \l "active) This specify an active link

**1.) a:link (Normal link)**

a:link represents the normal state of any link when no activity has been done over the link.

we can set color,font,style,background-color or any style on a normal link using CSS.

a:link{

font-size:20px;

color:red;

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=css-link-link" \t "_blank)

**2. a:visited**

The a:visited specify the state of link when link is visited (when link is clicked at least once).

To show the link is visited generally the color of the link is changed.

a:link{

font-size:20px;

color:red;

}

a:visited{

color:cadetblue;

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=css-link-visited" \t "_blank)

**3. a:hover (mouse over)**

The a:hover specifies a state of link when mouse pointer is over it.

Hyperlink can be made to change behaviour when the mouse pointer is over it.

a:link{

font-size:20px;

color:red;

}

a:visited{

color:cadetblue;

}

a:hover{

font-size:25px;

color:blue;

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=css-link-hover" \t "_blank)

**4. a:active (just clicked)**

The a:active specifies the state of link when link is active (when mouse is just clicked).

To show the link is active generally the color of the link is changed.

a:link{

font-size:20px;

color:red;

}

a:visited{

color:cadetblue;

}

a:hover{

font-size:25px;

color:blue;

}

a:active{

color:gold;

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=css-link-active" \t "_blank)

**CSS LISTS**

[❮ Prev](https://www.tutorialstonight.com/css/css-links.php) [Next ❯](https://www.tutorialstonight.com/css/css-tables.php)

List is an effective way to present similarly related data.

List can be ordered or unordered. Ordered list shows list contents using numbers while unordered list shows list content using bullets.

In CSS we have following list properties to style list:

1. [CSS list-style-type](https://www.tutorialstonight.com/css/css-lists.php#type) This allows the user to change the shape of bullets or markers.
2. [CSS list-style-position](https://www.tutorialstonight.com/css/css-lists.php#position) Control content to be inside or outside of the bullet.
3. [CSS list-style-image](https://www.tutorialstonight.com/css/css-lists.php#image) This property replaces marker/bullet with specified image.
4. [CSS list-style](https://www.tutorialstonight.com/css/css-lists.php#style) This property let us use multiple list property at once.

**1.)CSS list style type**

The list-style-type property allows us to change the bullet(marker) model.

In **unordered list** bullets can be made in square,circle and disc.

<ul style="list-style-type:circle">

<li>Book</li>

<li>Pen</li>

<li>table</li>

</ul>

<ul style="list-style-type:square">

<li>Book</li>

<li>Pen</li>

<li>table</li>

</ul>

<ul style="list-style-type:disc">

<li>Book</li>

<li>Pen</li>

<li>table</li>

</ul>

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=list-type)

The **ordered list** marker can be made in number,lower alphabet,upper alphabet,roman etc.

<ol style="list-style-type:decimal">

<li>Book</li>

<li>Pen</li>

<li>table</li>

</ol>

<ol style="list-style-type:upper-alpha">

<li>Book</li>

<li>Pen</li>

<li>table</li>

</ol>

<ol style="list-style-type:upper-roman">

<li>Book</li>

<li>Pen</li>

<li>table</li>

</ol>

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=list-type-ordered)

**2.) CSS list style position**

The list-style-position property specifies whether the content of the list will be inside or outside of the bullet.

<ul style="list-style-position:outside">

<li>I am outside the bullet.I am outside the bullet.I am outside the bullet.</li>

<li>I am outside the bullet.I am outside the bullet.I am outside the bullet.</li>

</ul>

<ul style="list-style-position:inside">

<li>I am inside the bullet.I am inside the bullet.I am inside the bullet.</li>

<li>I am inside the bullet.I am inside the bullet.I am inside the bullet.</li>

</ul>

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=list-position" \t "_blank)

**3.) An image as bullet (list-style-image)**

The list-style-image property can be used to replace general bullets with an image.

<ol style="list-style-image:url('marker-image.jpg')">

<li>Book</li>

<li>Pen</li>

<li>table</li>

</ol>

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=list-image" \t "_blank)

**4.) CSS list shorthand (list-style)**

The list-style property lets us use all list properties at once like - *list-style-type,list-style-position and list-style-image*.

<ol style="list-style:disc outside url('assets/marker-image.jpg')">

<li>Book</li>

<li>Pen</li>

<li>table</li>

</ol>

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=list-all" \t "_blank)

Note: In **list-style** property if somehow the image is unable to load then the bullet will replace the image.

**CSS TABLES**

[❮ Prev](https://www.tutorialstonight.com/css/css-lists.php) [Next ❯](https://www.tutorialstonight.com/css/css-height-and-width.php)

Table helps us to present data relations in an effective manner.

Tables created in an HTML document can be styled using CSS table properties.

Example:

|  |  |  |
| --- | --- | --- |
| **Name** | **Age** | **Hobby** |
| John (Author) | 32 | Learning |
| Jeany | 25 | Exploration |
| Leo | 20 | Entertainment |
| Jerry | 5 | Playing |

**CSS table style**

Using CSS we can set following properties to the table :

1. [border-collapse](https://www.tutorialstonight.com/css/css-tables.php" \l "collapse)
2. [cell-spanning](https://www.tutorialstonight.com/css/css-tables.php" \l "spanning)
3. [Table Caption](https://www.tutorialstonight.com/css/css-tables.php" \l "caption)
4. [Text align](https://www.tutorialstonight.com/css/css-tables.php" \l "align)
5. [Background](https://www.tutorialstonight.com/css/css-tables.php" \l "background)

1.) Border collapse

To create a collapsed border table we need to use CSS property border-collapse.

table, td, th {

border: 1px solid black ;

border-collapse: collapse ;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=table-bordercollapse" \t "_blank)

**2).Cell spanning multiple rows or column**

Cells of tables span only one row and one column but we can increase their spanning to multiple rows or columns.

For spanning multiple rows or columns we can use the attribute rowspan for multiple row span and colspan for multiple column span.

As we can see in the example below:

<table style="width:100%" border="1">

<tr>

<th>Name</th>

<th colspan="2">Books</th>

</tr>

<tr>

<th rowspan="2">Shakespeare</th>

<td>King Lear</td>

<td>Romeo and juliet</td>

</tr>

<tr>

<th>Chetan Bhagat</th>

<td>3 mistakes of my life</td>

<td>Half Girlfriend</td>

</tr>

</table>

HTML

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=table-span" \t "_blank)

**3.) Table Caption**

We can add caption to any table. Caption provide basic information about the table.

For creating a caption to the table <caption> tag is used. This tag must be immediately used after the table tag.

To style caption we create classes and use caption-side property.

<table style="width:100%" border="1">

<caption style="caption-side:bottom">Books and their authors</caption>

<tr>

<th>Name</th>

<th colspan="2">Books</th>

</tr>

<tr>

<th rowspan="2">Shakespeare</th>

<td>King Lear</td>

<td>Romeo and juliet</td>

</tr>

<tr>

<th>Chetan Bhagat</th>

<td>3 mistakes of my life</td>

</tr>

</table>

HTML

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=table-caption" \t "_blank)

**4.) Text alignment in table**

In the grids of row and column texts can be aligned horizontally and vertically.

To align text horizontally use text-align property.

To align text vertically use vertical-align property.

th{

text-align:center;

}

tr{

height:15px;

vertical-align:bottom;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=table-align" \t "_blank)

**5.) Background color in table**

Using CSS we can add background color to the table element.In example we have created a zebra effect using nth.child() and background-color property.

tr:nth-child(even){

background-color:silver;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=table-background" \t "_blank)

**CSS HEIGHT AND WIDTH**

[❮ Prev](https://www.tutorialstonight.com/css/css-tables.php) [Next ❯](https://www.tutorialstonight.com/css/css-box-model.php)

CSS have property height and width which is used to set respectively height and width of the HTML element.

The padding defined is setted inside the element with already set height and width.

This element has 50% width and 150px height.

**CSS height/width values**

| **Values** | **Description** |
| --- | --- |
| auto | Browser automatically calculate the height/width.This is default. |
| inherit | The height/width is inherited by parent element. |
| length | It set height/width in px,pt,cm etc. |
| initial | It set height/width to its default value. |
| % | It set height/width using percentage of container they are inside. |

.box{

height:150px;

width:70%;

background-color:rgba(0,101,153,0.44);

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=css-height)

**CSS min-height**

Sometimes we need to set the minimum height of any HTML element or container. CSS min-height property is used to set the minimum height of the container.

The min-height value can be given in percentage, pixel, vh etc.

Note: when min-height is more than window size then a scroll bar will be created.

.min-height-prop{

min-height:600px;

background-color:rgba(0,101,153,0.44);

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=css-min-height)

**CSS min-width**

CSS min-width property is used to set the minimum width of any element or container.

The min-width value can be given in percentage, pixel, vh etc.

Note: when min-width is more than window size then a scroll bar will be created.

.min-width-prop{

min-width:120%;

background-color:rgba(0,101,153,0.44);

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=css-min-width)

**CSS max-height**

CSS max-height property is used to set the maximum height of HTML element or container.

After the maximum limit has been crossed CSS properties like background-color are not effective on overflown content on screen.

To set max-height you can use px, percentage, vh etc.

.max-height-prop{

max-height: 80px;

background-color: rgba(0,101,153,0.44);

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=css-max-height" \t "_blank)

**CSS max-width**

CSS max-width property is used to set the maximum width of HTML element or container.

After the maximum limit has been crossed CSS properties like background-color are not effective on overflown content on screen.

To set max-width you can use px, percentage, vh etc.

.max-width-prop{

max-width: 50%;

background-color: rgba(0,101,153,0.44);

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=css-max-width" \t "_blank)

**CSS BOX MODEL**

[❮ Prev](https://www.tutorialstonight.com/css/css-height-and-width.php) [Next ❯](https://www.tutorialstonight.com/css/css-borders.php)

Every HTML element can be compared with a box. Treating HTML elements as box we can visualise an element having following property:

* Content
* Padding
* Border
* Margin
* Height
* Width

**Margin Of The Box**

**Border Of The Box**

**Padding Of The Box**

CONTENT

**Effective Height and width of element**

From the above figure you can see that effective height and width of the content are :

* Total height = margin-top + margin-bottom + border-top + border-bottom + padding-top + padding-bottom + height
* Total width = margin-left + margin-right + border-left + border-right + padding-left + padding-right + width

div{

margin:15px;

border: 5px solid silver;

padding: 20px;

height: 50px;

width: 300px;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=box-model)

**CSS BORDERS**

[❮ Prev](https://www.tutorialstonight.com/css/css-box-model.php) [Next ❯](https://www.tutorialstonight.com/css/css-margins.php)

CSS border properties are used to design and shape borders of elements in web pages.

The CSS border can specify color, border type, width of border etc. These are following properties in CSS for border:

1. [CSS border-style](https://www.tutorialstonight.com/css/css-borders.php#style)
2. [CSS border-color](https://www.tutorialstonight.com/css/css-borders.php#color)
3. [CSS border-width](https://www.tutorialstonight.com/css/css-borders.php#width)
4. [CSS border-radius](https://www.tutorialstonight.com/css/css-borders.php#radius)

**1.) CSS Border Style**

The border-style specifies the type of border you want on a webpage.

The CSS properties border-style can have values like:

* solid - Border created is solid.
* dashed - Border created is dashed.
* dotted - Border created is dotted.
* double - Border created is double border.
* hidden - Border created is hidden.
* none - No border is created.
* groove - Groove defines a 3D grooved border.Effect depends on the color value of the border.
* ridge - Ridge defines a 3D ridged border.Effect depends on the color value of the border.
* inset - Inset defines a 3D inset border.Effect depends on the color value of border.

<p style="border-style:solid">Border created is solid.</p>

<p style="border-style:dashed">Border created is dashed.</p>

<p style="border-style:dotted">Border created is dotted.</p>

<p style="border-style:double">Border created is double.</p>

<p style="border-style:hidden">Border created is hidden.</p>

<p style="border-style:none">Border created is none.</p>

<p style="border-style:groove">Groove defines a 3D grooved border.</p>

<p style="border-style:ridge">ridge defines a 3D ridged border.</p>

<p style="border-style:inset">Inset defines a 3D inset border.</p>

HTML

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=border-style)

**2.) CSS Border Color**

Borders can be given color using border-color property.

The color value of border-color property can be given in form of:

* color name - like red,green,blue etc.
* HEX value - like #432512.
* RGB value - like rgb(45,90,124).
* HSL value - like hsl(95,50%,50%).

Note - Before setting border-color you must set border-style because to set color to border it must exist first.

p.color1{

border-style:solid;

border-color:red;

}

p.color2{

border-style:solid;

border-color:#d423bd;

}

p.color3{

border-style:solid;

border-color:rgb(124,45,20);

}

p.color4{

border-style:solid;

border-color:hsl(90,50%,50%);

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=border-color)

**3).CSS Border Width**

CSS border-width property controls the width of the border.

The width value of border-width property can be given in any valid scale.

Using border width property we can also control width of each side of the box by using following property:

* border-top-width - It is the width of top of the box.
* border-right-width - It is the width of the right of the box.
* border-bottom-width - It is the width of bottom of the box.
* border-left-width - It is the width of left of the box.

Note: Before setting border-width you must set border-style because to set width to border it must exist first.

p.width1{

border-style:solid;

border-width:5px;

}

p.width2{

border-style:solid;

border-top-width:6px;;

}

p.width3{

border-style:solid;

border-right-width:0px;

}

p.widtp{

border-style:solid;

border-left-width:6px;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=border-width)

Instead of using CSS property to specify width for each side of the box you can also set values like: border-width: top right bottom left.

Using this method you can set different values to different sides.

p.width{

border-style:solid;

border-width: 1px 3px 0px 5px;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=border-singlewidth)

**4.) CSS Border Radius**

The CSS border-radius property is used to set radius to the borders.

p.radius{

border-style: solid;

border-color: lightpink;

border-radius: 10px;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=border-radius)

**CSS MARGINS**

[❮ Prev](https://www.tutorialstonight.com/css/css-borders.php) [Next ❯](https://www.tutorialstonight.com/css/css-padding.php)

The margin property is used to specify space around the HTML element.The specified space is transparent and has no background color.

This paragraph has a margin of 50px.

Using this property we can specify top,bottom,left and right margin independently.These are following CSS margin properties.

**CSS margin properties**

| **Property** | **Description** |
| --- | --- |
| margin | This sets all the value of margin. |
| margin-top | It sets the top margin of the element. |
| margin-right | It sets the right margin of the element. |
| margin-bottom | It sets the bottom margin of the element |
| margin-left | It sets the left margin of the element. |

**CSS margin values**

| **Values** | **Description** |
| --- | --- |
| auto | Browsers automatically calculate the margin. |
| inherit | It inherits the margin from the parent element. |
| unit/length | It sets margin in px,pt,cm etc. |
| % | It sets the margin using the percentage of width of the container. |

**Margin on element**

Specifying CSS margin property and its value we can create margin over the element.

<p style="margin:none;background-color:silver">This paragraph has no margin.</p>

<p style="margin:25px;background-color:silver">This has a margin of 25px.</p>

HTML

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=css-margin" \t "_blank)

**Set different margin**

We can specify different margin values to different side of the elements by using

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

.set\_margin{

margin-top:50px;

margin-right:30px;

margin-bottom:10px;

margin-left:25px;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=margin-specific" \t "_blank)

**CSS margin shorthand**

Margin shorthand property is used to specify different margins to different sides of an element using just one attribute margin. Example : ***margin : 20px 30px 40px 50px***

The values in shorthand margin are set anticlockwise starting from top=20px, right=30px, bottom=40px and ends at left=50px.

.shorthand{

margin: 20px 30px 40px 50px;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=margin-shorthand" \t "_blank)

**CSS PADDING**

[❮ Prev](https://www.tutorialstonight.com/css/css-margins.php) [Next ❯](https://www.tutorialstonight.com/css/css-outline.php)

The CSS padding property is used to create space around the element's content.

This paragraph has padding of 50px from border and 20px padding inside element.Unlike margin property padding property generate space only for content.

Using padding property we can specify top,bottom,left and right padding independently.These are following padding properties.

**CSS padding properties**

| **Property** | **Description** |
| --- | --- |
| padding | This sets all the value of padding. |
| padding-top | It sets the top padding of the element. |
| padding-right | It sets the right padding of the element. |
| padding-bottom | It set bottom padding of the element. |
| padding-left | It set left padding of the element. |

**CSS padding values**

| **Values** | **Description** |
| --- | --- |
| length | It set padding in px,pt,cm etc. |
| % | It set padding using the percentage of the container. |

**Setting padding to element**

Specifying CSS padding property and its value we can create padding over the element.

<p style="padding:none;">This paragraph has no padding.</p>

<p style="padding:25px">This has a padding of 25px.</p>

HTML

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=css-padding)

**Set different padding**

We can specify different margin values to different side of the elements by using

* padding-top
* padding-right
* padding-bottom
* padding-left

.set\_padding{

padding-top:50px;

padding-right:30px;

padding-bottom:30px;

padding-left:25px;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=padding-specific)

**CSS padding shorthand**

Padding shorthand property is used to specify different padding to different side of element using just one attribute padding.Example : ***padding : 20px 30px 40px 50px***

The values in shorthand padding are set anticlockwise starting from top=20px,right=30px,bottom=40px and ends at left=50px.

.shorthand{

padding: 20px 30px 40px 50px;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=padding-shorthand)

# CSS Outline - Something Out of the Box Model

[❮ Prev](https://www.tutorialstonight.com/css/css-padding.php) [Next ❯](https://www.tutorialstonight.com/css/css-overflow.php)

## CSS Outline

CSS outline is a line that is drawn around the HTML element just outside of the border to make the element stand out.

The CSS outline looks just like the border, but it is drawn outside of the border. The outline is not part element's box model but the border is part of the element's box model.

CSS outlines are are generally used to create focus on certain areas like buttons, form, links, etc.

You can see the outline is around the element that lies outside the black border.

Example of CSS outline:

Example

.para {

outline: 10px solid #5b79a2;

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=css-outline" \t "_blank)

# Syntax

/\* Just type or style \*/

outline: type; /\* (solid | dotted ...) \*/

/\* width & type \*/

outline: width type;

/\* color and type \*/

outline: width color;

/\* width, color and style \*/

outline: width color style;

The order of the properties is not important. You can use any of the three properties in any order.

The other 2 properties are width and color are optional but you have to mention the type or style of the outline every time you use them.

Example

.para1 { outline: solid }

.para2 { outline: 6px double}

.para3 { outline: dotted red }

.para4 { outline: dashed 4px #00f }

.para5 { outline: 4px dashed #00f }

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=css-outline-combination" \t "_blank)

## CSS Outline Property

The CSS outline has the following properties:

1. [outline-style](https://www.tutorialstonight.com/css/css-outline.php" \l "style)
2. [outline-color](https://www.tutorialstonight.com/css/css-outline.php" \l "color)
3. [outline-width](https://www.tutorialstonight.com/css/css-outline.php" \l "width)
4. [outline-offset](https://www.tutorialstonight.com/css/css-outline.php" \l "offset)
5. [outline](https://www.tutorialstonight.com/css/css-outline.php" \l "outline) (shorthand)

### 1. CSS Outline Style

The outline-style property sets a style of an element's outline. Default outline-style is none.

The outline-style can have one following values:

* solid - It defines a solid outline.
* dashed - It defines a dashed outline.
* dotted - It defines a dotted outline.
* double - It defines a double outline.
* groove - It defines a 3D grooved outline.
* inset - It defines a 3D inset outline.
* outset - It defines the 3D outset outline.
* ridged - It defines a 3D ridged outline.
* none - It defines no outline.
* hidden - It defines a hidden outline.

Example

p.solid { outline-style: solid; }

p.dashed { outline-style: dashed; }

p.dotted { outline-style: dotted; }

p.double { outline-style: double; }

p.groove { outline-style: groove; }

p.inset { outline-style: inset; }

p.outset { outline-style: outset; }

p.ridge { outline-style: ridge; }

p.none { outline-style: none; }

p.hidden { outline-style: hidden; }

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=outline-style" \t "_blank)

### 2. CSS Outline Color

The outline-color property sets the color to outline. The color can be set by any method like color name, HEX value, RGB value, HSL value, etc.

The default color of the outline is the same as the text color.

Example

.color1{

outline-style: solid;

outline-color: red;

}

.color2{

outline-style: solid;

outline-color: rgb(127, 65, 243);

}

.color3{

outline-style: solid;

outline-color: #23d43c;

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=outline-color" \t "_blank)

### 3. CSS Outline width

The outline-width property sets the thickness of the element's outline. outline-width has the following values:

* thin - It defines an outline of generally 1px.
* medium - It defines an outline of generally 3px.
* thick - It defines an outline of generally 5px.
* Any width - It defines a outline in px, pt, cm, etc.

Example

.thin {

outline-width: thin;

}

.medium {

outline-width: medium;

}

.thick {

outline-width: thick;

}

.fivepx {

outline-width: 5px;

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=outline-width" \t "_blank)

### 4. CSS Outline offset

The outline-offset property sets a gap between the border and outline. outline-offset value can be given in any valid unit. Default outline-offset value is 0.

Example

.offset-px {

outline-offset: 5px;

}

.offset-rem{

outline-offset: 1rem;

}

.offset-cm{

outline-offset: 1cm;

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=outline-offset" \t "_blank)

### 5. CSS Outline shorthand property

The outline is a shorthand property for all CSS outline properties, which is used to set multiple outline property values in a single line.

Using outline property we can specify one, two, or all three values of outline property, where outline-style is required property, outline-width and outline-color are optional properties. The order of these three doesn't matter.

Example

.order1 {

outline: solid;

}

.order2 {

outline: solid red;

}

.order3 {

outline: solid 10px;

}

.order4 {

outline: solid 10px red;

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=outline-shorthand" \t "_blank)

## Difference Between Border And Outline

The outline looks very similar to the border both are different from each other.

The difference between outline and border are as follows:

| **CSS outline** | **CSS border** |
| --- | --- |
| The outline doesn't take up space and can overlap other elements nearby | Borders do take space and does not overlap with another element |
| Outline do not change shape and size of element | Border change the shape and size of element |
| Outline can't be given different values to its different edges, it's the same in all directions | Borders can be given different values to the different sides of the border |
| Outline can't have another shape than a rectangle | Borders can have any shape between square and circle |

## CSS Text Outline

The text outline is a technique to draw a line around text. It can be used to indicate some special text or to highlight some text.

The text-stroke property sets the outline of the text. It can be used to highlight text or to indicate some special text.

The property that sets text outline is under experiment and is only supported by the browsers that extend support to 'webKit'. So the syntax also uses the prefix of webKit. The property is -webkit-text-stroke.

# Syntax

-webkit-text-stroke: color width;

/\* or \*/

-webkit-text-stroke: width color;

Example

.para1 {

-webkit-text-stroke: 2px #f00;

}

.para2 {

-webkit-text-stroke: #f00 2px;

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=text-outline" \t "_blank)

## Conclusion

If you want to give an HTML element an effect of the border without changing the shape of the element, then use the outline property. The outline is not the part of element's box-model hence it does not even interact with surrounding elements.

To give outline to texts you can use -webkit-text-stroke property.

## Frequently Asked Questions

1. What is outline?

The outline is a rectangular line that is drawn around an element outside the border. It does not change the shape of the element.

1. How do I make an outline in CSS?

To make an outline in CSS, you can use the outline property. The syntax is outline: style width color.

1. What is the use of outline in CSS?

The outline is used to give an effect of the border without changing the shape of the element.

1. What is CSS outline-color?

The outline-color property sets the color of an outline.

**CSS OVERFLOW**

[❮ Prev](https://www.tutorialstonight.com/css/css-outline.php) [Next ❯](https://www.tutorialstonight.com/css/css-cursor.php)

Some content of a web page may be too big to fit into the area.CSS overflow property is used to handle contents which overflow its block level container.

Max-height give to this box is 100px.So as the content overflow it will create an overflow for height.which can be scrolled. Lorem ipsum dolor sit amet consectetur adipisicing elit. Quae autem nostrum praesentium ab modi eveniet voluptate enim assumenda. Autem impedit totam omnis officia ratione laboriosam vel debitis dolores dicta facilis! Lorem ipsum dolor sit, amet consectetur adipisicing elit. Fugiat nihil consequuntur perferendis dolores, voluptas dolorem repudiandae architecto molestiae atque quisquam sequi doloremque aspernatur illo corporis commodi ut suscipit maiores!

Syntax

overflow: value;

**CSS overflow values**

| **Value** | **Description** |
| --- | --- |
| visible | It is default value.It specifies that content is displayed outside the box and content is not clipped. |
| hidden | It specifies that content that overflowed is clipped and has no scroll bar. |
| scroll | It specifies that the extra content is clipped and to see those, a scroll bar is given. |
| auto | It also provides a scroll bar like above property but in this scroll bar is provided only when it's necessary. |

**Overflow visible**

The overflow is visible by default.When the overflow value is set to visible it means the content is not clipped and extra content will be visible outside the box.

div.visible{

overflow: visible;

height: 200px;

width: 300px;

background-color: #00669949;

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=overflow-visible" \t "_blank)

Output:

**The content overflow and the content is visible outside the box.**Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque consequatur mollitia enim quia et Itaque consequatur mollitia incidunt esse nihil dolore excepturi iste, nostrum amet quaerat cupiditate nisi dolor, exercitationem numquam!

**Overflow hidden**

The overflow hidden property clips the content.When the overflow value is set to hidden it means the content is clipped and extra content will be hidden.

.hidden{

overflow: hidden;

height: 200px;

width: 300px;

background-color: #00669949;

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=overflow-hidden" \t "_blank)

Output:

**The content overflow and the content which is outside the box is hidden.**Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque consequatur mollitia enim quia et incidunt esse nihil dolore excepturi iste, nostrum amet quaerat cupiditate nisi dolor,nostrum amet quaerat cupiditate exercitationem numquam!

**Overflow scroll**

The overflow scroll property clips the content but extra content is visible by a scroll bar which is generated.

The scroll bar can be horizontal or vertical.

.scroll{

overflow: scroll;

height: 200px;

width: 300px;

background-color: #00669949;

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=overflow-scroll" \t "_blank)

Output:

**The content overflows and the content which is outside the box can be seen by scrolling.**Lorem ipsum dolor sit amet consectetur adipisicing elit. Itaque consequatur mollitia enim quia et incidunt esse nihil dolore excepturi iste, nostrum amet quaerat cupiditate nisi dolor, exercitationem numquam!

**Overflow auto**

If the overflow is clipped then auto property creates a scroll to make overflowed content visible.

The scroll bar created by auto can be horizontal or vertical.

.auto{

overflow: auto;

height: 200px;

width: 300px;

background-color: #00669949;

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=overflow-auto" \t "_blank)

Output:

**The content overflows and the content which is outside the box can be seen by scroll created by auto value.**Lorem ipsum dolor sit amet consectetur adipisicing elit. Libero, eligendi fugiat. Labore facere provident dicta voluptatibus ea obcaecati quas asperiores hic, rem praesentium adipisci aliquid nisi. Rerum reprehenderit dolores laborum.

**CSS CURSOR**

[❮ Prev](https://www.tutorialstonight.com/css/css-overflow.php) [Next ❯](https://www.tutorialstonight.com/css/css-position.php)

CSS cursor property in CSS specifies the type of cursor to be displayed when the mouse is over that element.

|  |  |
| --- | --- |
| Default property: | auto |
| Version: | CSS 2,3 |
| Animation: | no |
| Applies to: | all element |
| Inherited: | Yes |

**Showcase of different cursor type**

Take your mouse pointer over these buttons and see how the pointer changes.

auto default none initial inherit context-menu help pointer Progress wait cell crosshair text vertical-text alias copy move no-drop not-allowed all-scroll col-resize row-resize n-resize e-resize s-resize w-resize ne-resize nw-resize se-resize se-resize ew-resize ns-resize nesw-resize nwse-resize zoom-in zoom-out grab grabbing

Here is an example:

<!DOCTYPE html>

<html>

<head>

<style>

.cursor{

display: flex;

flex-wrap: wrap;

}

.cursor-button{

display: inline-block;

background-color: #006699;

color: white;

text-align: center;

border: 1px solid gray;

margin: 0px 5px 5px 0px;

padding: 5px;

min-width: 85px;

border-radius: 5px;

}

</style>

</head>

<body>

<h2>Take your mouse pointer over the buttons and see how mouse pointer change.</h2>

<span class="cursor-button" style=cursor:auto;>auto</span>

<span class="cursor-button" style=cursor:default;>default</span>

<span class="cursor-button" style=cursor:none;>none</span>

<span class="cursor-button" style=cursor:initial;>initial</span>

<span class="cursor-button" style=cursor:inherit;>inherit</span>

<span class="cursor-button" style=cursor:context-menu;>context-menu</span>

<span class="cursor-button" style=cursor:help;>help</span>

<span class="cursor-button" style=cursor:pointer;>pointer</span>

<span class="cursor-button" style=cursor:progress;>progress</span>

<span class="cursor-button" style=cursor:wait;>wait</span>

<span class="cursor-button" style=cursor:cell;>cell</span>

<span class="cursor-button" style=cursor:crosshair;>crosshair</span>

<span class="cursor-button" style=cursor:text;>text</span>

<span class="cursor-button" style=cursor:vertical-text;>vertical-text</span>

<span class="cursor-button" style=cursor:alias;>alias</span>

<span class="cursor-button" style=cursor:copy;>copy</span>

<span class="cursor-button" style=cursor:move;>move</span>

<span class="cursor-button" style=cursor:no-drop;>no-drop</span>

<span class="cursor-button" style=cursor:not-allowed;>not-allowed</span>

<span class="cursor-button" style=cursor:all-scroll;>all-scroll</span>

<span class="cursor-button" style=cursor:col-resize;>col-resize</span>

<span class="cursor-button" style=cursor:row-resize;>row-resize</span>

<span class="cursor-button" style=cursor:n-resize;>n-resize</span>

<span class="cursor-button" style=cursor:e-resize;>e-resize</span>

<span class="cursor-button" style=cursor:s-resize;>s-resize</span>

<span class="cursor-button" style=cursor:w-resize;>w-resize</span>

<span class="cursor-button" style=cursor:ne-resize;>ne-resize</span>

<span class="cursor-button" style=cursor:nw-resize;>nw-resize</span>

<span class="cursor-button" style=cursor:se-resize;>se-resize</span>

<span class="cursor-button" style=cursor:se-resize;>sw-resize</span>

<span class="cursor-button" style=cursor:ew-resize;>ew-resize</span>

<span class="cursor-button" style=cursor:ns-resize;>ns-resize</span>

<span class="cursor-button" style=cursor:nesw-resize;>nesw-resize</span>

<span class="cursor-button" style=cursor:nwse-resize;>nwse-resize</span>

<span class="cursor-button" style=cursor:zoom-in;>zoom-in</span>

<span class="cursor-button" style=cursor:zoom-out;>zoom-out</span>

<span class="cursor-button" style=cursor:grab;>grab</span>

<span class="cursor-button" style=cursor:grabbing;>grabbing</span>

</body>

</html>

HTML

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=css-cursor)

**CSS POSITION**

[❮ Prev](https://www.tutorialstonight.com/css/css-cursor.php) [Next ❯](https://www.tutorialstonight.com/css/css-display.php)

CSS position property is used to position an element on the web page.

CSS position property defines different types of positioning methods for the elements.

**CSS Position property**

There are five different types of positioning methods for elements:

1. [static](https://www.tutorialstonight.com/css/css-position.php#static)
2. [relative](https://www.tutorialstonight.com/css/css-position.php#relative)
3. [fixed](https://www.tutorialstonight.com/css/css-position.php#fixed)
4. [absolute](https://www.tutorialstonight.com/css/css-position.php#absolute)
5. [sticky](https://www.tutorialstonight.com/css/css-position.php" \l "sticky)

Syntax

position: value;

**1).Position static**

Position static is the default position value for any HTML element. It positions the element according to normal flow of the page.

The element positioned static is not affected by top,bottom,left and right properties.

p.static{

position: static;

background-color: silver;

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=position-static" \t "_blank)

**2).Position relative**

Position relative value sets the element relative to its normal value.

The position: relative; is followed by offsets either top,bottom,left and right.

p.relative{

position: relative;

top: 150px;

left: 40px;

background-color: silver;

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=position-relative" \t "_blank)

**3).Position fixed**

Position fixed value is used to set an element fixed to the page at any spot. Regardless of the page scroll it doesn't move the element is fixed to its position.

The top, bottom, left and right property is used to set the element where it should be located.

div.fixed{

position: fixed;

top: 400px;

left: 400px;

background-color: gold;

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=position-fixed" \t "_blank)

**4).Position absolute**

Position absolute value is used to set the positioning of the element according to the nearest positioned parent. Incase none of its parents is positioned then the element will position itself according to the document body.

Note- The parent is called positioned only when it has any position value other than static.

The top, bottom, left and right property is used to set the element if required.

div.parent{

position: relative;

top: 50px;

left: 40px;

background-color: gold;

}

div.child{

position: absolute;

top: 10px;

background-color: gold;

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=position-absolute" \t "_blank)

**5).Position sticky**

Position sticky value is used to make element **fix** or **stick** at any location. Beyond that location the element is free to move or scroll.

Note - This property needs at least one value among top,bottom,left and right.

Position sticky value toggle between **relative** and **fixed**. This property is generally used to stick any element or note at the top of a web page.

div.sticky{

position: sticky;

top: 20px;

background-color: gold;

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=position-sticky" \t "_blank)

**CSS Display**

[❮ Prev](https://www.tutorialstonight.com/css/css-position.php) [Next ❯](https://www.tutorialstonight.com/css/css-shadow.php)

CSS display property controls how the elements of HTML will look on the web page.

Every element in HTML has a default display property and using CSS we can control this property.

Default value of display property is inline.

**syntax**

display : value;

**CSS Display Values**

1. [display: inline;](https://www.tutorialstonight.com/css/css-display.php" \l "inline)
2. [display: block;](https://www.tutorialstonight.com/css/css-display.php" \l "block)
3. [display: inline-block;](https://www.tutorialstonight.com/css/css-display.php" \l "inblock)
4. [display: inherit;](https://www.tutorialstonight.com/css/css-display.php" \l "inherit)
5. [display: none;](https://www.tutorialstonight.com/css/css-display.php" \l "none)

| **value** | **properties** |
| --- | --- |
| inline | Displays the element as inline element |
| block | Displays the element as a block element |
| contents | It makes the container disappear only content is visible. |
| flex | It makes the element block level flex element |
| grid | It makes the element block level grid element |
| inline-block | It displays the element as inline-block.Unlike inline here you can add height and width to the element |
| inline-flex | It displays the element as inline-flex |
| inline-grid | It displays the element as inline-grid |
| inline-table | It displays the element as inline-table |
| list-item | It displays the element as list item |
| run-in | It displays the element block or inline and it depends on the content |
| table | Element behaves like a table. i.e - <table> |
| table-caption | Elements behave like table-caption. i.e - <caption> |
| table-cell | Element behaves like a table-cell. i.e - <td> |
| table-column | Element behaves like a table-column. i.e - <col> |
| table-row | Element behaves like a table-row. i.e - <tr> |
| initial | Set display value to initial value |
| inherit | Takes display value from the parent element |
| none | The element is not visible |

From the above list here are few important examples.

**1).Display inline**

The inline value displays the element within the line.It does not change line or take any extra space.

As we have mentioned above every HTML element have some default display value.So the elements having inline as a default value are :

* <span>
* <em>
* <a>
* <b> etc.

p.inline{

display:inline;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=display-inline" \t "_blank)

2).Display block

The block value displays the elements from the new line as content reached maximum available width for that element.

The elements having inline as a default value are :

* <p>
* <h1 - h6>
* <div>
* <section> etc.

b.block,

em.block{

display:block;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=display-block" \t "_blank)

3).Display inline-block

The inline-block value displays the elements within the line. Using inline-block value you can also set the height and width of the elements.

The elements having inline-block as a default value are :

* <button>
* <select>

.inlineblock{

display: inline-block;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=display-inlineblock" \t "_blank)

body{

display: inline;

}

p.my\_class{

display: inherit;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=display-inherit" \t "_blank)

**5).Display None**

The none display value removes the element.The element does not take any space.

.my\_class{

display: none;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=display-none" \t "_blank)

# CSS SHADOW EFFECT

[❮ Prev](https://www.tutorialstonight.com/css/css-display.php) [Next ❯](https://www.tutorialstonight.com/css/css-float.php)

CSS shadow property is used to create shadow effect for HTML elements and texts.

These are CSS shadow properties:

1. [text-shadow](https://www.tutorialstonight.com/css/css-shadow.php" \l "text)
2. [box-shadow](https://www.tutorialstonight.com/css/css-shadow.php" \l "box)

## 1).CSS text shadow

The text-shadow property is used to apply shadow to the text.

Using text-shadow property you can add horizontal or vertical shadow with different color and blur radius.

This table summarizes use and properties of text-shadow:

|  |  |
| --- | --- |
| Default value | none |
| Inherited | Yes |
| Animation | yes |
| Applies to | all elements |
| Version | CSS3 |

### i).Simple text shadow

Syntax:

text-shadow: offset-x offset-y;

p{

text-shadow: 2px 2px;

font-size: 20px;

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=shadow-text-simple" \t "_blank)

Output:

This paragraph has shadow property.

### ii).Text shadow with different color

Syntax:

text-shadow: offset-x offset-y color;

p{

text-shadow: 2px 2px blue;

font-size: 20px;

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=shadow-text-color" \t "_blank)

Output:

This paragraph has blue shadow.

### iii).Text shadow with different color and blur-radius

Syntax:

text-shadow: offset-x offset-y blur-radius color;

p{

text-shadow: 2px 2px 5px blue;

font-size: 20px;

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=shadow-text-blur" \t "_blank)

Output:

This paragraph has blue shadow with blur-radius 5px.

### iv).Multiple shadow

Syntax:

text-shadow: shadow1,shadow2,shadow3...;

*#Where shadow = offset-x offset-y blur-radius color*

p{

text-shadow: 2px 2px 5px blue, 4px 4px 5px red, 6px 6px 5px #006699;

font-size: 20px;

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=shadow-text-multiple" \t "_blank)

Output:

This paragraph has blue shadow with blur-radius 5px.

## 2).CSS box shadow

The box-shadow property is used to create shadow for elements.

CSS have the same property values for box-shadow as in text-shadow.

div{

box-shadow: 10px 10px gray;

width: 200px;

height: 100px;

background-color: silver;

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=shadow-box-simple" \t "_blank)

Output:

This is a div element.

Creating CSS box-shadow with blur effect.

div{

box-shadow: 10px 10px 10px gray;

width: 200px;

height: 100px;

background-color: silver;

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=shadow-box-blur" \t "_blank)

Output:

This is a div element.

**CSS FLOAT**

[❮ Prev](https://www.tutorialstonight.com/css/css-shadow.php) [Next ❯](https://www.tutorialstonight.com/css/css-align.php)

CSS float property floats the element left or right of its container. CSS float property defines flow of content. Once float value is defined element is removed from normal flow of webpage and behaves like defined.

It is used to push HTML elements or containers *left* or *right*.

**CSS Float properties**

| **Value** | **Description** |
| --- | --- |
| left | It pushes element to left. |
| right | It pushes element to right. |
| inherit | It inherit float value from parent element. |
| none | It is default value of float.It specifies that element is not floated. |

**CSS float left**

The element having property float:left; is pushed left to the webpage.

.box-left{

float: left;

width: 10vw;

border: 1px solid silver;

background-color: #00669950;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=float-left" \t "_blank)

Output:

**Floating Left**

**CSS float right**

The element having property float:right; is pushed right to the webpage.

.box-right{

float: right;

width: 10vw;

border: 1px solid silver;

background-color: #00669950;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=float-right" \t "_blank)

Output:

**Floating Right**

**CSS clear property**

CSS clear property is used to control the element which will float beside that element.

If we specify value clear: left; then no floating element will be allowed to the left side of the element.

.float-left{

float: left;

border: 1px solid brown;

}

.clear{

clear: left;

border: 1px solid silver;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=float-clear" \t "_blank)

Output:

**Floating left**

This div element does not allow any element to float its left side.

**CSS ALIGN**

[❮ Prev](https://www.tutorialstonight.com/css/css-float.php) [Next ❯](https://www.tutorialstonight.com/css/css-icons.php)

CSS provides us various methods to align elements on the web page.Some of the text and box aligning methods are discussed below.

**CSS text align**

The text-align is used to align text inside any container.

The text-align has following values:

* left - It aligns the text to the left of the container.It is default value.
* right - It aligns the text to the right of the container.
* center - It aligns the text to the center of the container.
* justify - When the value specified is 'justify' the text fills the space of the container exactly
* start - It aligns the text from the starting of the container.When the direction: rtl; then starting position is right so it will start from right side and will look as if it's right aligned.
* end - It aligns the text from the ending of the container.When the direction: rtl; then ending position is left so it will end at left side and will look as if it's left aligned.
* inherit - It inherits text-align value from its parent element.

.left{

text-align: left;

background-color: #00669950;

}

.start{

text-align: start;

background-color: #00669950;

direction: rtl;

}

.justify{

text-align: justify;

width: 200px;;

background-color: #00669950;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=align-text" \t "_blank)

**Align element horizontally center**

To align any element or container horizontally center use margin property and set its value to auto.It will create equal left and right margins which makes the container align to center.

To see the effect you must set the width value of the container which is less than 100% width value otherwise the effect will not be visible.

.horizontally-center{

margin: auto;

width: 200px;

border: 3px solid #006699;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=align-horizontally" \t "_blank)

Output:

This container is in center.

**vertically center - using padding**

We can align elements vertically center by using padding property.Just Specify equal top and bottom padding value and it will align vertically center.

.vertically-center{

padding: 50px 0px;

border: 3px solid #006699;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=align-vertically-padding" \t "_blank)

**vertically center - using line height**

We can align elements vertically center by using line-height property.Specify equal line-height as the height of the container.

.vertically-center{

line-height: 100px;

height: 100px;

border: 3px solid #006699;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=align-vertically-line-height" \t "_blank)

**CSS ICONS**

[❮ Prev](https://www.tutorialstonight.com/css/css-align.php) [Next ❯](https://www.tutorialstonight.com/css/css-button.php)

Icon is a comprehensible symbol which is used to display something in the form of sign or symbol on the computer screen.

HOME

SERVER

SEARCH

CLONE

DOWNLOAD

TREE

How to add icons

To add icons to the webpage we use icon library. These icon libraries create these icons using CSS so you can add other CSS properties to it too; like color,shadow,size etc. These icons are embedded in webpage by using HTML tags like <i> or <span>

There are many icon libraries which you can use like

* Font awesome icons
* Google icons
* Bootstrap icons

These icons libraries are added in the **head** section using **link** tag.

**Font awesome icons**

Add the icon library link in the head section of the webpage and use classes defined by font awesome in <i> tag or <span> tag. You can also add CSS style to the icons like color,shadow,font-size etc. To increase or decrease the size of icons use classes like : *fa-xs*,*fa-sm*,*fa-lg*,*fa-2x*,*fa-3x*, ... ,*fa-10x*.

<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet"

href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

</head>

<body>

<div style="text-align:center">

<i class="fa fa-book"></i>

<i class="fa fa-anchor"></i>

<i class="fa fa-search fa-2x"></i>

<i class="fa fa-heart" style="color:red"></i>

<i class="fa fa-twitter" style="color:cyan"></i>

</div>

</body>

</html>

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=icons-fa" \t "_blank)

**OUTPUT:**

**Google icons**

Add icon library link in head section of webpage and use class material-icons in <i> tag or <span> tag and name of the icon is mentioned in between the tag. You can also add CSS style to the icons like color,shadow,font-size etc. To increase or decrease the size of icons you can use CSS property font-size.

<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet"

href="https://fonts.googleapis.com/icon?family=Material+Icons">

</head>

<body>

<div style="text-align:center">

<i class="material-icons">home</i>

<i class="material-icons">attachment</i>

<i class="material-icons">search</i>

<i class="material-icons" style="color:red">favourite</i>

<i class="material-icons" style="color:cyan">computer</i>

</div>

</body>

</html>

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=icons-google" \t "_blank)

**OUTPUT:**

*home* *attachment* *search* *favorite* *computer*

**Bootstrap icons**

Add the icon library link in the head section of the webpage and use classes defined by Bootstrap icons in <i> tag or <span> tag. You can also add CSS style to the icons like color,shadow,font-size etc. To increase or decrease the size of icons you can use CSS property font-size.

<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet"

href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">

</head>

<body>

<div style="text-align:center">

<i class="glyphicon glyphicon-home" style="font-size:1.5em;color:blue"></i>

<i class="glyphicon glyphicon-user" style="font-size:2em"></i>

<i class="glyphicon glyphicon-remove" style="font-size:2.5em"></i>

<i class="glyphicon glyphicon-envelope" style="color:red;font-size:3em"></i>

<i class="glyphicon glyphicon-cloud" style="color:cyan;font-size:3.5em"></i>

</div>

</body>

</html>

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=icons-bootstrap" \t "_blank)

**CSS BUTTONS**

[❮ Prev](https://www.tutorialstonight.com/css/css-icons.php) [Next ❯](https://www.tutorialstonight.com/css/css-form.php)

A button is used to perform certain tasks when clicked.

Example:

* A button can be used to invoke certain kind of function

Fibonacci is sum of two preceding numbers

Fibonacci here...

click to see fibonacci series

* A button can be used to take you to another page

**styling button using CSS**

Using CSS properties we can design a button according to our need.

.button1{

background-color: #38577b;

color: white;

padding: 10px;

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=css-button" \t "_blank)

**Button border**

Using border property you can define the border of the button.

.button1{

border: 2px solid #38577b;

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=button-border" \t "_blank)

Output:

[My button](https://www.tutorialstonight.com/css/css-button.php) [My button](https://www.tutorialstonight.com/css/css-button.php) [My button](https://www.tutorialstonight.com/css/css-button.php) [My button](https://www.tutorialstonight.com/css/css-button.php)

**Button size**

Using font-size property you can adjust the size of the button.

.button1{

font-size: 12px;

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=button-size" \t "_blank)

Output:

[My button](https://www.tutorialstonight.com/css/css-button.php) [My button](https://www.tutorialstonight.com/css/css-button.php) [My button](https://www.tutorialstonight.com/css/css-button.php) [My button](https://www.tutorialstonight.com/css/css-button.php)

**Button shadow**

Using box-shadow property you can get button shadow.

.button1{

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

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=button-shadow" \t "_blank)

Output:

[My Button](https://www.tutorialstonight.com/css/css-button.php)

**Button Hover**

Using :hover selector you can define hover property.

.button1:hover{

background-color: #38577b;

color: white;

text-decoration: none;

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

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=button-hover" \t "_blank)

Output:

[My Button](https://www.tutorialstonight.com/css/css-button.php) [My Button](https://www.tutorialstonight.com/css/css-button.php) [My Button](https://www.tutorialstonight.com/css/css-button.php) [My Button](https://www.tutorialstonight.com/css/css-button.php)

**CSS FORM**

[❮ Prev](https://www.tutorialstonight.com/css/css-button.php) [Next ❯](https://www.tutorialstonight.com/css/css-combinators.php)

[Forms are made using HTML](https://www.tutorialstonight.com/html/html-form.php) but we can improve its look using CSS. using CSS you can designe a beautiful form by creating borders, background color, hover effect, focus property, transition etc

you can control width, height and many other property of input field using CSS. you can also add image and focus property to the input fields.

Top of Form

First name:  
  
Last name:  


Bottom of Form

**CSS for label**

You can add CSS property like font-size, color, shadow etc tolabel of form.

label{

color:red;

font-size:22px;

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=form-label" \t "_blank)

**Output:**

Top of Form

Name: 

Bottom of Form

**Setting CSS property to input field**

You can select input field and define CSS property like width,border,shadow,margin etc.

input{

border: 2px solid red;

width:100%;

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=form-input" \t "_blank)

**Output:**

Top of Form

Name: Password: 

Bottom of Form

**Styling input field by its type**

Instead of slecting all input you can select input fields by its type and specify CSS property.

input[type=text]{

border: 2px solid red;

width:100%;

}

input[type=password]{

border: 2px solid green;

width:100%;

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=form-type" \t "_blank)

**Output:**

Top of Form

Name: Password: 

Bottom of Form

**Creating animated Search bar using CSS**

To create animated search bar you can define input text field and specify CSS properties for it one for normal styling and one of cursor focus.

input[type=text]{

border: 2px solid silver;

width: 15%;

padding: 10px;

background-image: url('search-icon.jpg');

background-repeat: none;

}

input[type=text]:focus{

border: 2px solid green;

width:90%;

background-color:#0066992c;

transition:0.5s;

padding: 10px 35px;

font-size: 20px;

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=form-search" \t "_blank)

**CSS COMBINATORS**

[❮ Prev](https://www.tutorialstonight.com/css/css-form.php) [Next ❯](https://www.tutorialstonight.com/css/css-specificity.php)

CSS combinators is a combination of two or more other [selectors](https://www.tutorialstonight.com/css/css-selectors.php) and creates a relationship between them. The combinators also provide location of the content in the HTML document and is used to target specific elements to style.

There are four types of selector in CSS:

1. [Descendant selector](https://www.tutorialstonight.com/css/css-combinators.php#descendant)
2. [Adjacent sibling (+)](https://www.tutorialstonight.com/css/css-combinators.php#adjacent)
3. [Child selector (>)](https://www.tutorialstonight.com/css/css-combinators.php#child)
4. [General sibling selector (~)](https://www.tutorialstonight.com/css/css-combinators.php#general)

**1).Descendant selector**

The descendant selector selects all the elements that are children of specified previous selectors.

This combinator is written by a combination of other selectors separated by **space**.

Example - div ul li This will select all li elements which are nested inside ul and ul is nested inside div elements.

div ul li{

color: red;

background-color: #0066992c;

margin: 5px;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=combinator-descandent)

**2).Adjacent sibling**

The adjacent selector selects the elements that are adjacent to the specified selector. Here adjacent means immediately after the specified selector.

This combinator is written by a combination of other selectors separated by **plus (+)**.

Example - h2 + p This will select p element if it is immediately after h2 element.

h3 + p{

color: brown;

font-size: 1.5em;

background-color: #0066992c;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=combinator-adjacent)

**3).Child selector**

The child selector selects all the elements that are the child of the specified selector. Here adjacent means immediately after the specified selector.

This combinator is written by combination of other selectors and **greater than (>)** symbols in between.

Example - div > p This will select div as parent and select all the p elements that are their own children of it.

div > p{

color: white;

font-size: 1.5em;

background-color: #006699;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=combinator-child)

**4).General sibling selector**

The general sibling selector selects all the sibling elements that follows the first selector element.

This combinator is written by combination of other selectors and **tilde (~)** symbol in between.

Example - div ~ p This will select p element that are siblings of div element.

div ~ p{

color: white;

font-size: 1.5em;

background-color: #006699;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=combinator-general)

Top of Form



Bottom of Form

Follow Us

* [HTML5](https://www.tutorialstonight.com/html/html-introduction.php)
* [CSS3](https://www.tutorialstonight.com/css/css-introduction.php)
* [Javascript](https://www.tutorialstonight.com/js/)
* [Bootstrap 4](https://www.tutorialstonight.com/bootstrap/bootstrap-introduction.php)
* [Python](https://www.tutorialstonight.com/python/python-tutorial.php)
* [Articles](https://www.tutorialstonight.com/password-generator-in-javascript.php)

CSS Tutorial

[CSS Introduction](https://www.tutorialstonight.com/css/css-introduction.php) [CSS Implementation](https://www.tutorialstonight.com/css/css-implementation.php) [CSS Syntax](https://www.tutorialstonight.com/css/css-syntax.php) [CSS Selector](https://www.tutorialstonight.com/css/css-selectors.php) [CSS Class](https://www.tutorialstonight.com/css/css-class.php) [CSS Id](https://www.tutorialstonight.com/css/css-id.php) [CSS Color](https://www.tutorialstonight.com/css/css-color.php) [CSS Background](https://www.tutorialstonight.com/css/css-background.php) [CSS Units](https://www.tutorialstonight.com/css/css-units.php) [CSS fonts](https://www.tutorialstonight.com/css/css-font.php) [CSS text](https://www.tutorialstonight.com/css/css-text.php) [CSS Links](https://www.tutorialstonight.com/css/css-links.php) [CSS Lists](https://www.tutorialstonight.com/css/css-lists.php) [CSS Tables](https://www.tutorialstonight.com/css/css-tables.php) [CSS Height /width](https://www.tutorialstonight.com/css/css-height-and-width.php)

CSS box model

[CSS Box Model](https://www.tutorialstonight.com/css/css-box-model.php) [CSS Borders](https://www.tutorialstonight.com/css/css-borders.php) [CSS Margins](https://www.tutorialstonight.com/css/css-margins.php) [CSS Padding](https://www.tutorialstonight.com/css/css-padding.php)

CSS advanced

[CSS Outline](https://www.tutorialstonight.com/css/css-outline.php) [CSS Overflow](https://www.tutorialstonight.com/css/css-overflow.php) [CSS Cursor](https://www.tutorialstonight.com/css/css-cursor.php) [CSS Position](https://www.tutorialstonight.com/css/css-position.php) [CSS Display](https://www.tutorialstonight.com/css/css-display.php) [CSS shadow](https://www.tutorialstonight.com/css/css-shadow.php) [CSS Float](https://www.tutorialstonight.com/css/css-float.php) [CSS Align](https://www.tutorialstonight.com/css/css-align.php) [CSS Icons](https://www.tutorialstonight.com/css/css-icons.php) [CSS Buttons](https://www.tutorialstonight.com/css/css-button.php) [CSS Form](https://www.tutorialstonight.com/css/css-form.php) [CSS Combinators](https://www.tutorialstonight.com/css/css-combinators.php) [CSS Specificity](https://www.tutorialstonight.com/css/css-specificity.php) [CSS 2D transforms](https://www.tutorialstonight.com/css/css-2D-transformation.php) [CSS 3D transforms](https://www.tutorialstonight.com/css/css-3D-transformation.php) [CSS transition](https://www.tutorialstonight.com/css/css-transition.php) [CSS Animation](https://www.tutorialstonight.com/css/css-animation.php) [CSS variable](https://www.tutorialstonight.com/css/css-variable.php) [CSS Pagination](https://www.tutorialstonight.com/css/css-pagination.php) [CSS flexbox](https://www.tutorialstonight.com/css/css-flexbox.php) [CSS Media Queries](https://www.tutorialstonight.com/css/css-media-queries.php)

CSS MISCELLANEOUS

[How to center a div in CSS](https://www.tutorialstonight.com/how-to-center-a-div-in-css.php) [How to center an image in CSS](https://www.tutorialstonight.com/how-to-center-an-image-in-css.php)

[Description: Description: Ezoic](https://www.ezoic.com/what-is-ezoic/)report this ad

# CSS SPECIFICITY

[❮ Prev](https://www.tutorialstonight.com/css/css-combinators.php) [Next ❯](https://www.tutorialstonight.com/css/css-2D-transformation.php)

CSS specificity determines which rule set is to be applied if there are multiple CSS rules pointing to the same element.

CSS selectors are provided with different values and if multiple CSS rules are pointing to the same element then on the basis of those selector values CSS rules are ranked.

The specificity value on the basis of selectors ranges from 0 to 1000.

On the basic of different selectors specificity value is given as follows:

* inline style - Inline style has highest value of specificity among other selectors.Its is 1000. Can be represented as ( 1, 0, 0, 0)
* id - Id has second highest specificity value.Its is 100. Can be represented as ( 0, 1, 0, 0)
* class,pseudo-class,attribute selector - These have third highest value of specificity among other selectors.Its is 10. Can be represented as ( 0, 0, 1, 0)
* element or pseudo element - These have lowest specificity value.Its is 1. Can be represented as ( 0, 0, 0, 1)

## Calculation of specificity

specificity is calculated on the basis of values given as we have mentioned above. For multiple numbers the selector value is added together.

Example 1:

div p{

color: red;

background-color: light-yellow;

}

p.add-class{

color: blue;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=specificity-element)

Here in this example class will be applied to **p** because specificity for element rule is 1+1(two elements)=2 but specificity of class is 10(1 class)+1(1 element)=11.

### Example 2:

div p.add-color{

color: red;

background-color: light-yellow;

}

p#id-color{

color: blue;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=specificity-class)

Here in this example id will be applied to **p** because specificity for id rule is 100(1 id)+1(1 element)=101 but specificity of class is 10(1 class)+2(2 element)=12.

<hr size=2 width="100%" align=center>

### Example 3:

<style>

p#id-color{

color: blue;

}

</style>

<p style="color:red">This paragraph is red.</p>

HTML

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=specificity-inline)

<hr size=2 width="100%" align=center>

Here in this example inline style will be applied to **p** because specificity for inline rule is 1000(inline) but specificity of id is 100(1 id)+1(1 element)=101.

## Equal specificity

When there are two CSS rules having the same Specificity value then the rule that is defined lower in the style sheet is applied.

p{

color: blue;

}

p{

color: red;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=specificity-equal)

**CSS 2D TRANSFORMS**

[❮ Prev](https://www.tutorialstonight.com/css/css-specificity.php) [Next ❯](https://www.tutorialstonight.com/css/css-3D-transformation.php)

Using CSS3 2D transform feature HTML elements can be scaled, skewed, moved, translated and rotated in 2-dimensional shape.

2D transform

A transformed element doesn't affect the surrounding element, it gets overlapped over them. The space taken by a transformed element is the same space when it's not transformed.

There are following CSS 2D transformation methods that can be applied over HTML elements are as follows:

1. [CSS translate](https://www.tutorialstonight.com/css/css-2D-transformation.php#translate)
2. [CSS rotate](https://www.tutorialstonight.com/css/css-2D-transformation.php#rotate)
3. [CSS scale](https://www.tutorialstonight.com/css/css-2D-transformation.php#scale)
4. [CSS skew](https://www.tutorialstonight.com/css/css-2D-transformation.php#skew)
5. [CSS matrix](https://www.tutorialstonight.com/css/css-2D-transformation.php#matrix)

**1.) CSS translate**

The CSS translate() method is used to translate or move an HTML element on the scale of x and y coordinate.

The translate() takes two argument **x** and **y**. The arguments values **x** and **y** could be negative also.   
Example - **transform: translate( x,y );**

.my-div {

transform: translate(100px, 100px);

-moz-transform: translate(100px, 100px);

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=2D-transform-translate)

Output:

This div element is translated 50px to right and 50px to bottom by using CSS translate.

**2.) CSS rotate**

The CSS rotate() method is used to rotate the HTML element clockwise or anticlockwise.

The rotate() takes argument in degree. The argument values can be negative as well as positive. Positive value rotates the element in clockwise whereas negative value rotates element in anticlockwise direction.   
Example - **transform: rotate( 20deg );** or **transform: rotate( -20deg );**

.my-div {

transform: rotate( 90deg);

-moz-transform: rotate( 90deg);

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=2D-transform-rotate)

Output:

This div element is rotated 90deg by using CSS rotate method.

**3.) CSS scale**

The CSS scale() method is used to increase or decrease size of element on the basic parameter given in **scale()**.

The scale() takes two arguments in numbers one for width and one for height.   
Example - **transform: scale( width,height );**

.my-div {

transform: scale( 1.5,2);

-moz-transform: scale( 1.5,2);

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=2D-transform-scale)

Output:

scaled width 1.5× and height 2×.

**4.) CSS skew**

The CSS skew() method is used to skew the element on the **x** and **y** axis.

The skew() takes two arguments in degree one for **x** and one for **y**.   
Example - **transform: skew( Xdeg,Ydeg );**

.my-div {

transform: skew( 20deg, 0deg);

-moz-transform: scale( 20deg,0deg);

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=2D-transform-skew)

Output:

This div element is skewed 20deg in X and 0deg Y by using CSS skew method.

**5.) CSS matrix**

The CSS matrix method is a combination of all methods.

The 2D matrix method takes six arguments.   
Example - **transform: matrix( scaleX(),skewY(),skewX(),scaleY(),translateX(),translateX() );**

.my-div {

transform: matrix( 1.2,0, 1,1.5,100,50);

-moz-transform: matrix( 1.2,0, 1,1.5,100,50);

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=2D-transform-matrix)

Output:

This div element is using matrix method.

**CSS 3D TRANSFORMS**

[❮ Prev](https://www.tutorialstonight.com/css/css-2D-transformation.php) [Next ❯](https://www.tutorialstonight.com/css/css-transition.php)

Using CSS 3D transform you can perform 3D operations on HTML elements like scale, skew, rotate, translate etc.

The 3D translated elements don't affect surrounding elements and may overlap them. The translated element takes the same space as its default.

To apply 3D transformation CSS use transform property and values are given as required like scaling, rotation etc.

We are going to learn following 3D transform properties:

1. [CSS translate3d](https://www.tutorialstonight.com/css/css-3D-transformation.php#translate)
2. [CSS rotate3d](https://www.tutorialstonight.com/css/css-3D-transformation.php#rotate)
3. [CSS scale3d](https://www.tutorialstonight.com/css/css-3D-transformation.php#scale)

**1.) CSS translate3d**

Using CSS translate3d you can translate or move HTML elements along X, Y and Z-axis.

The CSS translate3d takes three arguments, each for one axis **X**, **Y** and **Z** respectively. The coordinate values could be negative as well as positive.

Example- **transform: translate3d( x, y, z)**

.my-div{

transform: translate3d( 50px,10px,20px);

-moz-transform: translate3d( 50px,10px,20px);

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=3D-transform-translate)

Output:

This div element is translated in 3D 50px in x-axis, 10px in Y-axis and 20px in Z-axis.

**2.) CSS rotate3d**

Using CSS rotate3d you can rotate HTML elements along X, Y and Z-axis.

The CSS rotate3d takes four arguments, each for one axis **X**, **Y**, **Z** and degree value respectively. The X, Y and Z value is multiplied with specified degree internally and this gives how much angle should be rotated for each axis.

Example- **transform: rotate3d( x, y, z, deg)**

.my-div{

transform:rotate3d( 1, 1, 1.2, 60deg);

-moz-transform: rotate3d( 1, 1, 1.2, 60deg);

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=3D-transform-rotate)

Output:

This div element is rotated in X and Y-axis by 60deg and Z-axis by 1.2×60deg=72deg.

**3.) CSS scale3d**

Using CSS scale3d you can scale HTML elements along X, Y and Z-axis.

The CSS scale3d changes the size of the element. It takes 3 arguments, each for **X**, **Y** and **Z** respectively.

For CSS rotate3d to work you need to use either rotate3d property or perspective property.

Example- **transform: scale3d( x, y, z)**

.my-div{

transform: scale3d( 1, 1.5, 2) rotate3d(1,1,0,30deg);

-moz-transform: scale3d( 1, 1.5, 2) rotate3d(1,1,0,30deg);

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=3D-transform-scale)

Output:

This div element is scaled in X by 1, in Y-axis by 1.5× and in Z-axis by 2×.

**CSS TRANSITION**

[❮ Prev](https://www.tutorialstonight.com/css/css-3D-transformation.php) [Next ❯](https://www.tutorialstonight.com/css/css-animation.php)

CSS transition is used to give a smooth effect in value change of CSS property.

CSS Transition

CSS transition controls the animation and calculates the intermediate state between two different property values. Instead of changing value instantaneously it slowly changes older to newer value.

These are transition properties:

1. [transition](https://www.tutorialstonight.com/css/css-transition.php" \l "transition)
2. [transition-delay](https://www.tutorialstonight.com/css/css-transition.php" \l "delay)
3. [transition-duration](https://www.tutorialstonight.com/css/css-transition.php" \l "duration)
4. [transition-property](https://www.tutorialstonight.com/css/css-transition.php" \l "property)
5. [transition-timing-function](https://www.tutorialstonight.com/css/css-transition.php" \l "timing-function)

**1.) CSS transition**

transition is shorthand property for all transition properties ( we will look for different properties further in the chapter).

For CSS transition to work you must add two things

1. The property you want to add effect on.
2. transition time other than 0.

Here is a working example that changes its color from red to blue in 2 seconds.

.box{

background-color: red;

height: 200px;

width: 200px;

}

.box:hover{

background-color: blue;

transition: 4s;

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=css-transition" \t "_blank)

Output:

Hover Me!

**2.) CSS transition delay**

transition-delay property controls the time of starting of the transition.

If transition effect is not mentioned then transition starts immediately as it triggered but when transition-delay is mention then effect starts taking place after mentioned time.

.box{

background-color: red;

height: 100px;

width: 200px;

}

.box:hover{

transition: 2s;

transition-delay: 1s;

background-color: blue;

width: 100%;

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=transition-delay" \t "_blank)

Output:

1 second delay

**3.) CSS transition duration**

transition-duration property defines the time in which value of any CSS property will change from value1 to value2.

If transition duration is not mentioned then transition effect is not visible as change in values takes place immediately.

.box{

background-color: red;

height: 100px;

width: 150px;

}

.box:hover{

transition-duration: 3s;

background-color: blue;

width: 100%;

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=transition-duration" \t "_blank)

Output:

Transition Duration 3 sec

**4.) CSS transition property**

transition-property defines the CSS property value you want transition effect for.

Suppose there are two properties that are changing its value then transition-property defines that property for which you want transition animation work for.

.box{

background-color: red;

height: 100px;

width: 150px;

}

.box:hover{

transition: width 3s;

width: 100%;

background-color: blue;

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=transition-property" \t "_blank)

Output:

width change takes 3sec but background-color changes immediately

**5.) CSS transition timing function**

transition-timing-function defines the speed by which transition effect takes place.

The transition-timing-function have following values:

* ease - This is default value. It starts slow, then fast and ends slowly.
* linear - When specified the transition effect takes place at same speed from start to end.
* ease-in - Transition takes a slow start.
* ease-out - Transition takes a slow end.
* ease-in-out -Transition takes a slow start and slow end.
* cubic-bezier(n,n,n,n) - Using this you can define your own bezier curve.

.box1 {

transition-timing-function: ease;

}

.box2 {

transition-timing-function: linear;

}

.box3 {

transition-timing-function: ease-in;

}

.box4 {

transition-timing-function: ease-out;

}

.box5 {

transition-timing-function: ease-in-out;

}

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=transition-function" \t "_blank)

**CSS ANIMATION**

[❮ Prev](https://www.tutorialstonight.com/css/css-transition.php) [Next ❯](https://www.tutorialstonight.com/css/css-variable.php)

CSS animation property allow us to creatin animation effect for change in element's style.

Lets first see an example and then further understand its key features.

@keyframes my-animation{

from { margin-left: 0;}

to { margin-left: 90%;

background-color: yellowgreen;

}

}

.box{

animation-name: my-animation;

animation-duration: 3s;

animation-iteration-count: infinite;

height: 50px;

width: 100px;

background-color: #006699;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=css-animation)

Output:

Animation

**Building blocks of CSS animation**

There are two building blocks of CSS animation:

1. @keyframes - It creates the platform for animation.

@keyframes animation-name{

.

.

.

}

CSS

 keyframes animation properties - Inside keyframes animation properties are specified. The animation properties are defined either in terms of to and from or in percentage form.

@keyframes animation-name{

from { property1: value1;}

to { property2: value2;}

}

CSS

@keyframes animation-name{

0% { property1: value1;}

50% { property2: value2;}

100% { property2: value3;}

}

CSS

**Steps to create animation**

Step 1:- Create @keyframes inside style tag.

Step 2:- Name @keyframes to be used further. Example: **@ketframes My\_animation { ... }**

Step 3: - Define CSS configuration inside keyframes for animation.

Step 4: - Mention animation-name with desired element you want animation for.  
Example: **div{ animation-name: My\_animation;}** .

Note: You must mention animation-duration property otherwise animation will not take place because a default value 0 is set to the animation-duration.

@keyframes my-animation{

0% { margin-left: 0; background-color: #006699;}

50% { margin-left: 90%; background-color: yellowgreen; border-radius: 50%;}

100% { margin-left: 0; background-color: #006699;}

}

.box{

animation-name: my-animation;

animation-duration: 4s;

animation-iteration-count: infinite;

height: 50px;

width: 100px;

background-color: #006699;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=css-animation2)

Output:

Animation

**CSS animation delay**

The animation-delay property specifies the delay time for animation to start.

@keyframes my-animation{

0% { margin-left: 0; background-color: #006699;}

50% { margin-left: 90%; background-color: yellowgreen; border-radius: 50%;}

100% { margin-left: 0; background-color: #006699;}

}

.box{

animation-name: my-animation;

animation-delay: 2s;

animation-duration: 4s;

animation-iteration-count: infinite;

height: 50px;

width: 100px;

background-color: #006699;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=animation-delay)

# CSS VARIABLE

[❮ Prev](https://www.tutorialstonight.com/css/css-animation.php) [Next ❯](https://www.tutorialstonight.com/css/css-pagination.php)

CSS variables are used to store a value that can be used frequently in the HTML document.

The variables are defined by a name preceded by double dash and value is assigned to the variable after a colon. Example: **--my-color: teal;**

:root{

--my-color: teal;

}

CSS

The variable is defined in a way mentioned above.

Now, to access the variable and to use it in the HTML document we use var() function. Example: **color: var(--my-color);**

p {

color: var(--my-color);

}

CSS

Working example:-

<!DOCTYPE html>

<html>

<head>

<style>

:root{ --my-color: teal; }

</style>

</head>

<body>

<h2>Learning about CSS variables.</p>

<p style="color: var(--my-color)">This paragraph is using the CSS variable.</p>

</body>

</html>

HTML

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=css-variable)

## CSS variable scope

Any variable that we define in CSS has a scope value. Beyond a variable's scope is not accessible.

We have two type of scope :

1. Global scope
2. Local scope

### 1.) Global scope

A variable which is defined globally can be accessed anywhere in the linked document.

A global variable is defined inside :root selector.

:root{

--my-background: orange;

--my-padding: 15px;

--my-margin: 10px;

}

.box{

background-color: var(--my-background);

padding: var(--my-padding);

margin: var(--my-margin);

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=variable-global)

### 2.) Local scope

A variable is defined locally by defining it within the code block of a specific selector.

A local variable can only be used within the selector it is defined.

.box{

--my-background: orange;

--my-padding: 15px;

--my-margin: 10px;

background-color: var(--my-background);

padding: var(--my-padding);

margin: var(--my-margin);

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=variable-local)

[❮ Prev](https://www.tutorialstonight.com/css/css-animation.php) [Next ❯](https://www.tutorialstonight.com/css/css-pagination.php)

**CSS PAGINATION**

[Previous](https://www.tutorialstonight.com/css/css-variable.php) [Next](https://www.tutorialstonight.com/css/css-flexbox.php)

CSS pagination is a way by which web users can easily jump anywhere in a large collection of related webpages. Pagination is generally in number form, which specifies nth webpage from starting.

In websites with a large number of webpages they sort the webpages in this manner.

**Simple pagination**

This is a simple pagination show next and previous buttons and a few pages in between.

.simple-pagination{

color: white;

background-color: #006699;

padding: 10px 15px;

border: 2px solid silver;

text-decoration: none; /\*to remove underline from numbers\*/

display: inline-block;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=pagination-simple)

Output:

[prev](https://www.tutorialstonight.com/css/css-pagination.php) [1](https://www.tutorialstonight.com/css/css-pagination.php) [2](https://www.tutorialstonight.com/css/css-pagination.php) [3](https://www.tutorialstonight.com/css/css-pagination.php) [4](https://www.tutorialstonight.com/css/css-pagination.php) [5](https://www.tutorialstonight.com/css/css-pagination.php) [6](https://www.tutorialstonight.com/css/css-pagination.php) [7](https://www.tutorialstonight.com/css/css-pagination.php) [next](https://www.tutorialstonight.com/css/css-pagination.php)

**Round pagination**

To create rounded pagination add border-radius to the pagination class and set value to 50%.

.round-pagination{

border-radius: 50%;

color: white;

background-color: #006699;

padding: 10px 15px;

border: 2px solid silver;

text-decoration: none; /\*to remove underline from numbers\*/

display: inline-block;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=pagination-round)

Output:

[1](https://www.tutorialstonight.com/css/css-pagination.php) [2](https://www.tutorialstonight.com/css/css-pagination.php) [3](https://www.tutorialstonight.com/css/css-pagination.php) [4](https://www.tutorialstonight.com/css/css-pagination.php) [5](https://www.tutorialstonight.com/css/css-pagination.php) [6](https://www.tutorialstonight.com/css/css-pagination.php) [7](https://www.tutorialstonight.com/css/css-pagination.php)

**Pagination with Hover effect**

Hover effect of pagination is initiated when you take the mouse pointer over the pagination buttons.

To define hover effect we need to add :hover CSS property for the pagination. Lets see in example below.

.hover-pagination{

color: white;

padding: 10px 15px;

border: 2px solid silver;

text-decoration: none;

display: inline-block;

}

.hover-pagination:hover{

color: white;

background-color: #006699;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=pagination-hover)

Output:

[prev](https://www.tutorialstonight.com/css/css-pagination.php) [1](https://www.tutorialstonight.com/css/css-pagination.php) [2](https://www.tutorialstonight.com/css/css-pagination.php) [3](https://www.tutorialstonight.com/css/css-pagination.php) [4](https://www.tutorialstonight.com/css/css-pagination.php) [5](https://www.tutorialstonight.com/css/css-pagination.php) [6](https://www.tutorialstonight.com/css/css-pagination.php) [7](https://www.tutorialstonight.com/css/css-pagination.php) [next](https://www.tutorialstonight.com/css/css-pagination.php)

**Active Pagination**

For quick knowledge for the user which page he is on currently, we make the page number representing that web page active on pagination.

To show that this pagination block is active we use sharp background-color for that section. Let's see in example.

.active{

background-color: #006699;

color: white;

}

.pagination{

padding: 10px 15px;

border: 2px solid silver;

text-decoration: none;

display: inline-block;

}

.pagination:hover:not(.active){

background-color: silver;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=pagination-active)

Output:

[prev](https://www.tutorialstonight.com/css/css-pagination.php) [1](https://www.tutorialstonight.com/css/css-pagination.php) [2](https://www.tutorialstonight.com/css/css-pagination.php) [3](https://www.tutorialstonight.com/css/css-pagination.php) [4](https://www.tutorialstonight.com/css/css-pagination.php) [5](https://www.tutorialstonight.com/css/css-pagination.php) [6](https://www.tutorialstonight.com/css/css-pagination.php) [7](https://www.tutorialstonight.com/css/css-pagination.php) [next](https://www.tutorialstonight.com/css/css-pagination.php)

**Centered Pagination**

To beautify a web page with pagination it's better if our pagination is in the center of the page.

To align the pagination in the [center](https://www.tutorialstonight.com/css/css-align.php) of the webpage we can use margin property.

.div-pagination{

margin: 0 auto;

text-align: center;

}

.active{

background-color: #006699;

color: white;

}

.pagination{

padding: 10px 15px;

border: 2px solid silver;

text-decoration: none;

display: inline-block;

}

.pagination:hover:not(.active){

background-color: silver;

}

CSS

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=pagination-center)

Output:

[prev](https://www.tutorialstonight.com/css/css-pagination.php) [1](https://www.tutorialstonight.com/css/css-pagination.php) [2](https://www.tutorialstonight.com/css/css-pagination.php) [3](https://www.tutorialstonight.com/css/css-pagination.php) [4](https://www.tutorialstonight.com/css/css-pagination.php) [5](https://www.tutorialstonight.com/css/css-pagination.php) [6](https://www.tutorialstonight.com/css/css-pagination.php) [7](https://www.tutorialstonight.com/css/css-pagination.php) [next](https://www.tutorialstonight.com/css/css-pagination.php)

# CSS FLEXBOX - THE ULTIMATE GUIDE

[❮ Prev](https://www.tutorialstonight.com/css/css-pagination.php) [Next ❯](https://www.tutorialstonight.com/css/css-media-queries.php)

In this CSS flexbox tutorial, you will learn everything about flexbox and what you can do with this.

1. [What is CSS flexbox?](https://www.tutorialstonight.com/css/css-flexbox.php#what-is-flexbox)
2. [Understanding Flexbox](https://www.tutorialstonight.com/css/css-flexbox.php#understanding-flexbox)
3. [How to use flexbox?](https://www.tutorialstonight.com/css/css-flexbox.php#how-to-use-flexbox)
4. [All CSS Flexbox Properties](https://www.tutorialstonight.com/css/css-flexbox.php#all-flexbox-properties)
5. [Flexbox CSS properties for container itself (Parent)](https://www.tutorialstonight.com/css/css-flexbox.php#flexbox-container-properties)
   1. [flex-direction](https://www.tutorialstonight.com/css/css-flexbox.php#direction)
   2. [flex wrap](https://www.tutorialstonight.com/css/css-flexbox.php#wrap)
   3. [flex-flow](https://www.tutorialstonight.com/css/css-flexbox.php#flow)
   4. [justify content](https://www.tutorialstonight.com/css/css-flexbox.php#justify-content)
   5. [align-items](https://www.tutorialstonight.com/css/css-flexbox.php#align-items)
   6. [align content](https://www.tutorialstonight.com/css/css-flexbox.php#align-content)
6. [Flexbox CSS properties for container's items (Child)](https://www.tutorialstonight.com/css/css-flexbox.php#flexbox-container-item-properties)
   1. [Flexbox order](https://www.tutorialstonight.com/css/css-flexbox.php#order)
   2. [Flex grow](https://www.tutorialstonight.com/css/css-flexbox.php#grow)
   3. [Flex basis](https://www.tutorialstonight.com/css/css-flexbox.php#basis)
   4. [Flex shrink](https://www.tutorialstonight.com/css/css-flexbox.php#shrink)
   5. [Flex](https://www.tutorialstonight.com/css/css-flexbox.php#flex)
   6. [align self](https://www.tutorialstonight.com/css/css-flexbox.php#align-self)

## What is CSS flexbox?

CSS flexbox is a one-dimensional **web layout method** that is used to arrange HTML containers in a row and column order.

Flexbox or CSS flexbox allows the [HTML](https://www.tutorialstonight.com/html/html-introduction.php" \t "_blank) elements to get automatically arranged depending on screen sizes.

The main idea behind CSS flexbox is to give the container ability to change its item's size automatically to best fit in available space.

Here are 3 images that show how HTML containers arranged using flexbox automatically fits themselves according to device width.

[▶ See full code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=css-flexbox-demo-example" \t "_blank)

CSS flexbox makes web developers life easy to design and arrange HTML elements in the webpage.

Before CSS flexbox it was quite difficult to fit and align-items on a webpage. The items were mostly laid out using [float](https://www.tutorialstonight.com/css/css-float.php" \t "_blank) and [position](https://www.tutorialstonight.com/css/css-position.php" \t "_blank) properties. Even though using just float and position properties we were unable to perform many tasks like:

* Aligning items vertically
* Making child element take up equal space even with unequal length of items

## Understanding CSS flexbox

**CSS flexbox is not a property its a module** ( a W3C recommendation as of Oct 2017 ). It involves a whole set of CSS properties, some of this property is for the container itself on which flexbox is applied and some are for its child elements.

To understand flexbox you have to think in terms of 2 axes -

* **the main axis**: The horizontal axis
* **the cross axis**: The verticle axis

The items of the container will either lie along the **main axis** or the **cross axis** and these items can be placed at a various place along this 2 axis.

So now we have an idea, that container will align its item in 2 direction horizontal and verticle and along with these 2 directions, it will be spread in various ways, like align in the start, in the end, in middle, or spread in other ways.

## How to use flexbox?

To start with flexbox we first need to define flexbox in our parent element so that other properties of flexbox could be used by the parent as well as a child element.

To make the container a flexible box define display: flex or display: inline-flex to the container:

* **display: flex** makes the container itself behave like a block element
* **display: inline-flex** makes container behave like inline element

.container {

display: flex; /\* or inline-flex \*/

}

CSS

[▶ Play Code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=css-flexbox" \t "_blank)

Output:

1

2

3

4

5

You can see in the output above adding display: flex property creates a flex container and by default, its item is aligned in the horizontal direction.

## All CSS Flexbox Properties

Since you know now how to create a flexbox in CSS, its time to know all the CSS properties associated with flexbox.

The flexbox properties can be divided into 2 parts:

* **CSS that is applied on the container itself (Parent)** - Flexbox properties which is applied on the container itself are as follows:
  1. [flex-direction](https://www.tutorialstonight.com/css/css-flexbox.php" \l "direction)
  2. [flex-wrap](https://www.tutorialstonight.com/css/css-flexbox.php" \l "wrap)
  3. [flex-flow](https://www.tutorialstonight.com/css/css-flexbox.php" \l "flow)
  4. [justify-content](https://www.tutorialstonight.com/css/css-flexbox.php" \l "justify-content)
  5. [align-items](https://www.tutorialstonight.com/css/css-flexbox.php" \l "align-items)
  6. [align-content](https://www.tutorialstonight.com/css/css-flexbox.php" \l "align-content)
* **CSS that is applied on items of the container (Child)** - Flexbox properties which is applied on the items of the container are as follows:
  1. [order](https://www.tutorialstonight.com/css/css-flexbox.php" \l "order)
  2. [flex-grow](https://www.tutorialstonight.com/css/css-flexbox.php" \l "flex-grow)
  3. [flex-shrink](https://www.tutorialstonight.com/css/css-flexbox.php" \l "flex-shrink)
  4. [flex-basis](https://www.tutorialstonight.com/css/css-flexbox.php" \l "flex-shrink)
  5. [flex](https://www.tutorialstonight.com/css/css-flexbox.php" \l "flex)
  6. [align-self](https://www.tutorialstonight.com/css/css-flexbox.php" \l "align-self)

Now let's see each property and what effect it creates with example.

## Flexbox CSS that is applied on container itself (Parent)

Let's look at all the flexbox property applied to the flexbox container.

### 1. CSS flex-direction

Flexbox module provides a property called flex-direction which specify what direction the children are laid out in.

flex-direction has 4 different values:

1. row: It aligns flexbox items from left to right. It is the default value of flex-direction.
2. row-reverse: It aligns flexbox items from the right end of the container to the left end.
3. column: It aligns flexbox items from top to bottom in the flexbox container.
4. column-reverse: It aligns flexbox items from bottom to top in the flexbox or container.

.container {

display: flex;

flex-direction: row; /\* or row-reverse | columns | column-reverse \*/

}

CSS

Let's see flex-direction property with each value with examples.

#### I. flex direction row

If you want your container elements to be in a row then use flex-direction: row property. It is also the default value of the flex-direction property, so you do not have to explicitly add it.

Possible mistake: Just applying flex-direction property on any container won't work you have use it on the container with flexbox (display: flex)

.container {

display: flex;

flex-direction: row; /\* Default value \*/

}

CSS

[▶ Play Code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=flex-direction-row" \t "_blank)

Output:

1

2

3

4

5

#### II. flex direction row reverse

To set container item in a row but in reverse direction use flex-direction: row-reverse property.

This property cause container items to start from the end of the container in reverse order.

.container {

display: flex;

flex-direction: row-reverse;

}

CSS

[▶ Play Code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=flex-direction-row-reverse" \t "_blank)

Output:

1

2

3

4

5

#### III. flex-direction column

The flex-direction: column aligns the flexbox items in columns i.e from top to bottom.

.container {

display: flex;

flex-direction: column;

}

CSS

[▶ Play Code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=flex-direction-column" \t "_blank)

Output:

1

2

3

#### IV. flex-direction column reverse

The flex-direction: column-reverse aligns the flexbox items in columns in reverse order i.e from bottom to top.

.container {

display: flex;

flex-direction: column-reverse;

}

CSS

[▶ Play Code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=flex-direction-column-reverse" \t "_blank)

Output:

1

2

3

### 2. CSS flex-wrap

flex-wrap property is used to define whether the flex items are aligned in a single row or the items can flow to multiple rows.

flex-wrap property set values of wrapping items. There are three values that could be set for flex-wrap property nowrap, wrap and wrap-reverse.

1. nowrap: It forces flex items to be in the same row. It is the default value for flex-wrap.
2. wrap: It let flex items to flow to the next row if there is no space to fit in the same row.
3. wrap-reverse: It specifies that flex items can flow to multiple rows but items will start from the bottom of the flexbox and after fill that it will flow to the upper row.

Let's see examples of each property value.

#### I. flex-wrap nowrap ( default )

The flex-wrap: nowrap property forces container items to stay in the same row. It items do not fit in the container then it is simply overflown.

.container {

display: flex;

flex-wrap: nowrap; /\* default value \*/

}

CSS

[▶ Play Code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=flex-wrap-nowrap" \t "_blank)

Output:

1

2

3

4

5

6

7

8

9

10

#### II. flex-wrap wrap

The flex-wrap: wrap property let the items to flow to next row if it does not fit in the container.

.container {

display: flex;

flex-wrap: wrap;

}

CSS

[▶ Play Code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=flex-wrap-wrap" \t "_blank)

Output:

1

2

3

4

5

6

7

8

9

10

#### III. flex-wrap wrap-reverse

The flex-wrap: wrap-reverse property let the items to flow to the next row but in reverse order, i.e item will start from the bottom of the container and will flow in an upward direction.

.container {

display: flex;

flex-wrap: wrap-reverse;

}

CSS

[▶ Play Code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=flex-wrap-wrap-reverse" \t "_blank)

Output:

1

2

3

4

5

6

7

8

9

10

## 3. flex-flow CSS

The flex-flow property is shorthand for flex-direction and flex-wrap.

Combining both flex-direction and flex-wrap property the flex-flow property defines the cross-axis flow of the container.

The default value of flex-flow is row nowrap.

.flow1 {

display: flex;

flex-flow: column wrap-reverse;

height: 400px; /\* limiting height to see effect \*/

}

.flow2 {

display: flex;

flex-flow: row wrap;

}

CSS

[▶ Play Code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=flex-flow" \t "_blank)

## 4. justify-content CSS

The justify-content property aligns flex items along the X-axis (horizontally) also known as **the main axis in the flexbox**. All flex line's items align horizontally according to justify-content value.

The justify-content property has 6 different values flex-start, flex-end, center, space-around, space-evenly and space-between.

1. flex-start: Aligns items horizontally to the start of the container (default value).
2. flex-end: Aligns items horizontally to the end of the container.
3. center: Aligns items horizontally to the center of the container.
4. space-around: Aligns items horizontally equally distributed.
5. space-evenly: Items are so aligned that space between container and item is the same as space between items.
6. space-between: Aligns items horizontally between the start and end of the container. The first row starts from left and the last row touches right.

Let's see examples of each value.

#### I. justify-content flex-start

This flex-start is the default value of the justify-content property, it aligns container items horizontally to the start of the container.

.container {

display: flex;

justify-content: flex-start;

}

CSS

[▶ Play Code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=flex-justify-content-flex-start" \t "_blank)

Output:

1

2

3

4

5

#### II. justify-content flex-end

This flex-end value of justify-content property aligns container items horizontally to the end of the container.

.container {

display: flex;

justify-content: flex-end;

}

CSS

[▶ Play Code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=flex-justify-content-flex-end" \t "_blank)

Output:

1

2

3

4

5

#### III. justify-content center

This center value of justify-content property aligns container items horizontally to the center of the container.

.container {

display: flex;

justify-content: center;

}

CSS

[▶ Play Code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=flex-justify-content-center" \t "_blank)

Output:

1

2

3

4

5

#### IV. justify-content space-around

This space-around value of justify-content property aligns container items horizontally to the end of the container.

.container {

display: flex;

justify-content: space-around;

}

CSS

[▶ Play Code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=flex-justify-content-space-around" \t "_blank)

Output:

Aligns items horizontally equally distributed.

1

2

3

4

#### V. justify-content space-evenly

This space-evenly value of justify-content aligns container items horizontally evenly in the container, space between the items is equal to space between items and container.

.container {

display: flex;

justify-content: space-evenly;

}

CSS

[▶ Play Code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=flex-justify-content-space-evenly" \t "_blank)

Output:

1

2

3

4

#### VI. justify-content space-between

This space-between value of justify-content property aligns container items horizontally between the start and endpoint of the container, both end item touches the container ends.

.container {

display: flex;

justify-content: space-between;

}

CSS

[▶ Play Code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=flex-justify-content-space-between" \t "_blank)

Output:

1

2

3

4

## 5. align-items CSS

The align-items property aligns the container items along Y-axis (Vertically), also known as **a cross-axis in flexbox**.

Note: The align-items is a single-line property if your container item wraps to the new line then it won't work for the new line. To align items as a whole use the **align-content** property.

The align-items property has 5 different values flex-start, flex-end, center, baseline and stretch.

1. stretch - It stretches the flex items along the container to fill the container but min-width and max-width are applied to items if mentioned. It is the default value of the align-item property.
2. flex-start - It aligns flex items to the top of the container.
3. flex-end - It aligns flex items to the bottom of the container.
4. center - It aligns flex items in the center of the container.
5. baseline - It aligns flex items in such a way that their baseline aligns.

Let's see an example of each value of this property.

#### I. align-items stretch

The flex-end value align-items property stretches the items along Y-axis and covers 100% height unless the min or max value is defined.

.container {

display: flex;

align-items: stretch; /\* Default value \*/

}

CSS

[▶ Play Code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=flex-align-item-stretch" \t "_blank)

Output:

1

2

3

4

5

#### II. align-items flex-start

The flex-start value align-items property aligns the items to the top of the container.

.container {

display: flex;

align-items: flex-start;

}

CSS

[▶ Play Code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=flex-align-item-flex-start" \t "_blank)

Output:

1

2

3

4

5

#### III. align-items flex-end

The flex-end value align-items property aligns the items to the bottom of the container. The items start from the bottom of the container.

.container {

display: flex;

align-items: flex-end;

}

CSS

[▶ Play Code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=flex-align-item-flex-end" \t "_blank)

Output:

1

2

3

4

5

#### IV. align-items center

The center value align-items property aligns the items to the center of the container along the Y-axis.

.container {

display: flex;

align-items: center;

}

CSS

[▶ Play Code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=flex-align-item-center" \t "_blank)

Output:

1

2

3

4

5

#### V. align-items baseline

The baseline value align-items property aligns the items along their baseline.

.container {

display: flex;

align-items: baseline;

}

CSS

[▶ Play Code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=flex-align-item-baseline" \t "_blank)

Output:

1

2

3

4

5

### 6. align content CSS

The align-content property works the same as the align-content property along Y-axis but the only difference in both is that instead of aligning flex items align-content aligns the flex container's line as a whole

**The align-content is multiline property**. Its effect is not visible when there is only one row in the flexbox.

The align-content property has 6 different values flex-start, flex-end, center, space-around, space-between and stretch.

1. flex-start: Aligns items to the start of the container packed together.
2. flex-end: Aligns items to the end of the container packed together.
3. center: Aligns items to the center of the container packed together.
4. space-around: Aligns items equally distributed along each row packed together.
5. space-between: Aligns items packed together between the start and end of the container. The first row starts from the top and the last row touches the bottom.
6. stretch: Aligns items packed together stretched along the cross axis of the container.

Let's see an example for each value of the align-item property.

#### I. align-content flex-start

flex rows aligned to flex-start which forces flex-items to start from top together in a group.

.container {

display: flex;

align-content: flex-start;

flex-wrap: wrap;

}

CSS

[▶ Play Code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=flex-align-content-flex-start" \t "_blank)

Output:

1

2

3

4

5

6

7

8

#### II. align-content flex-end

flex rows aligned to flex-end which forces flex-items to start from the bottom together in a group.

.container {

display: flex;

align-content: flex-end;

flex-wrap: wrap;

}

CSS

[▶ Play Code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=flex-align-content-flex-end" \t "_blank)

Output:

1

2

3

4

5

6

7

8

#### III. align-content center

flex rows align to the center of the container together in a group.

.container {

display: flex;

align-content: center;

flex-wrap: wrap;

}

CSS

[▶ Play Code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=flex-align-content-center" \t "_blank)

Output:

1

2

3

4

5

6

7

8

#### IV. align-content space-around

The space-around sets flex lines to align evenly in the container.

.container {

display: flex;

align-content: space-around;

flex-wrap: wrap;

}

CSS

[▶ Play Code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=flex-align-content-space-around" \t "_blank)

Output:

1

2

3

4

5

6

7

8

#### V. align-content space-between

flex rows aligned in space between the container first row touching the top and the last row touching the bottom.

.container {

display: flex;

align-content: space-between;

flex-wrap: wrap;

}

CSS

[▶ Play Code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=flex-align-content-space-between" \t "_blank)

Output:

1

2

3

4

5

6

7

8

#### VI. align-content stretch

flex rows items are stretched along the cross axis.

.container {

display: flex;

align-content: stretch;

flex-wrap: wrap;

}

CSS

[▶ Play Code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=flex-align-content-stretch" \t "_blank)

Output:

1

2

3

4

5

6

7

8

## Flexbox CSS that is applied on container's items (Child)

There are 6 different flexbox properties that are applied to the specific item of the flexbox container.

### 1. Flexbox order

The order CSS property of flexbox defines the order of an item in flexbox. Items of the container are sorted according to their **order value** in ascending order and then by their source code order.

The default value of the order is 0.

Order accepts only Integers (both positive and negative), decimal values are not allowed

#### flexbox order example

#target1 {

order: 0;

}

#target2 {

order: 1;

}

#target3 {

order: -1;

}

#target4 {

order: 5;

}

CSS

[▶ Play Code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=flexbox-order" \t "_blank)

### 2. flex-grow CSS

The flex-grow CSS property defines a **flex grow factor** of any specific item's main size, where the main size is either height or width of the container depends on the flex-direction property.

The flex-grow property defines how the remaining space in the container will be assigned to the item, where the remaining space is the size of the container which remain after all items fit in.

The flex-grow value is always positive or 0, the negative value is not supported.

#### flex grow example

Suppose you have 5 items of equal width 50px in a flex container with a width of 400px, and for simplicity suppose there is no margin between the items.

Remaining size: (400 - 50\*5) = 150px

Now, if we set flex-grow: 1 to 3rd item then this 150px will be given to it. And its total width will become 50px + 150px = 200px.

If we set flex-grow: 0.5 to 3rd item then this 75px will be given to it (50%).

If we set flex-grow: 2 to 3rd item and flex-grow: 1 to 4rt item then 2/3 of remaining (100px) will be given to 3rd element and 1/3 of remaining size will be given to 4th element(50px).

#target1 {

flex-grow: 2;

}

#target2 {

flex-grow: 1;

}

#target3 {

flex-grow: 0.5;

}

CSS

[▶ Play Code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=css-flex-grow" \t "_blank)

### 3. flex-basis

The flex-basis CSS property is used to set the size of the flex item.

In case both flex-basis and width (or height) are defined for an element then flex-basis has priority.

#target1 {

flex-basis: 150px;

}

CSS

[▶ Play Code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=flex-basis" \t "_blank)

### 4. flex-shrink

The flex-shrink CSS property is used to set the flex-shrink factor for a flex item.

It determines how much a flex item will shrink relative to the rest of the items in the container when the container is full.

#target1 {

flex-shrink: 5;

}

CSS

[▶ Play Code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=flex-shrink" \t "_blank)

### 5. flex

This is a shorthand property for flex-grow, flex-basis and flex-shrink.

/\* 1 value, unitless number: flex-grow \*/

flex: 1;

/\* 1 value, width/height: flex-basis \*/

flex: 15px;

flex: 10%;

/\* 2 values: flex-grow | flex-basis \*/

flex: 1 30px;

/\* 2 values: flex-grow | flex-shrink \*/

flex: 2 2;

/\* 3 values: flex-grow | flex-shrink | flex-basis \*/

flex: 2 2 10%;

CSS

### 6. align-self

The align-self property overwrites the align-items value for individual items. It aligns the item within the container in the Y-axis.

The align-self property has no effect on block-level items.

.container1 {

display: flex;

align-items: center;

}

#target1 {

align-self: self-end;

}

.container2 {

display: flex;

align-items: flex-start;

}

#target2 {

align-self: center;

}

CSS

[▶ Play Code](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=css-align-self" \t "_blank)

Points to remember:

1. CSS flexbox is a layout model which is used to position items in a container in 2D.
2. To start with flexbox you have to define your parent container as a flexbox by using display: flex | inline-flex.
3. Flexbox provides 6 difference CSS properties for the parent element and 6 difference CSS properties flexbox items.
4. To practice CSS flexbox in a fun way you can visit [flexbox froggy](https://flexboxfroggy.com/" \t "_blank), which provides a game that uses flexbox knowledge to play, you can practice flexbox here by playing the game.

Top of Form



# CSS Media Query - Make Responsive Websites

[❮ Prev](https://www.tutorialstonight.com/css/css-flexbox.php) [Home ❯](https://www.tutorialstonight.com/)

In this section, you will learn about the CSS media query. What is it used for, how you can use it and how to make a responsive website using it.

## What is a media query?

Media query is a technique introduced in CSS3 which let users define different CSS styles for different devices and different screen sizes.

It can modify the appearance and even behaviour of your webpage based on certain matched conditions.

It is useful when you want to design a webpage that auto-adjusts according to different devices like Desktop, laptop, pads, tablet, mobile etc.

Media query is mainly used to target a particular viewport range. It uses the @media rule with the condition to define a block of CSS which is applied to element when the condition is true. Here is an example we will discuss in detail later in this section.

/\* applied on screensize of 768px and above \*/

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

.box {

/\* apply your style \*/

}

}

CSS

Not only viewport width you can target a lot of things like screen resolution, device type, screen orientation, etc.

## Media query syntax

Now you have understood what a media query is, let's look at its syntax with examples.

A Media query has 4 different parts:

1. **At-rule (@media)** - Media query starts with the @media ruleset itself.
2. **Logical operator (not, only, and)** - Media query support logical operators like other programming languages like and, not, or (comma) and only.
3. **Media type (screen, printer, etc)** - It defines the type of media we are targeting like print, screen, speech or all.
4. **Media-feature (width, max-width, height, etc)** - Media feature target match features like width, height, aspect-ratio, orientation, etc.

A media query starts with the @media keyword followed by media-type and conditions separated by the logical operation.

@media not|only media-type and (mediafeature) {

/\* your CSS here... \*/

}

CSS

Here is a working example, in this example when the screen width gets greater than 768px than the background color of the webpage change to a light green color.

body {

background-color: lightpink;

}

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

body {

background-color: lightgreen;

}

}

CSS

[▶ Try it](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=media-query-example" \t "_blank)

## Media Query min width

The min-width media feature is used to set CSS properties for an element which works only when the minimum width of the media type (screen) is greater than the specified width.

In the example below, the background color and color of the webpage change when the screen width is greater than 500px.

Example 1

body {

background-color: black;

color: white;

}

/\* effective when minimum width is greater than 500px \*/

@media only screen and (min-width: 500px) {

body {

background-color: lightpink;

color: black;

}

}

CSS

[▶ Try it](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=media-query-min-width" \t "_blank)

Note: media queries must be last to define in style because defined styles are taken into consideration from top to down, if any media query is defined for any selector before defining normal CSS for that selector then the media query is ignored.

Here is another example where the column width increases to 100% when the viewport width gets smaller than 576px.

Example 2

.container {

display: flex;

flex-wrap: wrap;

border: 1px solid black;

}

.col {

width: 100%;

text-align: center;

outline: 1px solid gray;

height: 30px;

background-color: lightgreen;

}

@media only screen and (min-width:576px) {

.col {

width: 50%;

}

}

CSS

[▶ Try it](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=media-query-min-width-2" \t "_blank)

## Media Query max width

The max-width in the media query is used to set CSS properties which work only when the maximum width of the media type (screen) is smaller than or equal to the specified width.

body {

background-color: black;

color: white;

}

/\* effective when minimum width is greater than 500px \*/

@media only screen and (max-width: 500px) {

body {

background-color: lightpink;

color: black;

}

}

CSS

[▶ Try it](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=media-query-max-width" \t "_blank)

Note: The operator only is optional operator modern browser don't need it.

/\* this is also valid \*/

@media screen and (max-width: 500px) {

body {

background-color: lightpink;

color: black;

}

}

CSS

## Hiding an element on mobile using media query

There are some situations when you want to hide certain elements of a webpage on a mobile device or divide with a specific viewport width.

You can use [CSS display property](https://www.tutorialstonight.com/css/css-display.php" \t "_blank) and set it to none below breakpoint for mobile (here 320px is used).

div {

background-color: tomato;

color: white;

padding: 10px;

margin: 5px;

}

/\* effective when window width is smaller than 320px \*/

@media screen and (max-width: 320px) {

div {

display: none;

}

}

CSS

[▶ Try it](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=media-query-hide-and-show" \t "_blank)

## Combining multiple media feature

Media query let you combine multiple media feature (condition) as well as multiple media types separated by logical operators.

Here is an example which target device width with viewport width between 500px to 800px. When device width is between 500px to 800px then background is lightgreen else white.

@media (min-width:500px) and (max-width:800px) {

body {

background-color: lightgreen;

}

}

CSS

[▶ Try it](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=combining-media-query" \t "_blank)

## Inverting a query meaning

The not logical operator inverts the entire meaning of the media query. It will invert only the specified media query it is applied to.

If there are other comma-separated media query then it is not applied to all.

The not is evaluated last in the query.

@media not screen and (print) { ... }

CSS

The above query is evaluated like this:

@media not (screen and (print)) { ... }

CSS

Not like this:

~~@media (not screen) and (print) { ... }~~

~~CSS~~

Here is another example:

@media not screen and (color), print and (min-width:500px) { ... }

CSS

The above query is evaluated like as follows:

@media (not screen and (color)), print and (min-width:500px) { ... }

CSS

Not like as follows:

~~@media (not screen and (color), print and (min-width:500px)) { ... }~~

~~CSS~~

## Designing a responsive website using media query

The idea of the responsive website was generated from a media query.

Responsive websites are those websites that adjust elements size, alignment and orientation based on device width.

Let's design a webpage that is responsive to using media queries.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>CSS - media query responsive webiste</title>

<style>

.container {

display: flex;

flex-direction: column;

}

.header {

height: 80px;

background-color: #16d5ef;

}

.left-sidebar {

display: flex;

flex-direction: column;

width: 100%;

height: 200px;

margin-top: 10px;

background-color: #ff7e47;

}

.sidebar-column {

height: 30px;

margin: 10px;

background-color: white;

}

.main-element {

padding: 0;

}

.main-column {

background-color: #dfdfdf;

height: 112px;

margin-top: 10px;

}

.right-sidebar {

height: 80px;

margin-top: 10px;

background-color: #2cda72;

}

@media (min-width:600px) {

.left-sidebar {

height: 50px;

flex-direction: row;

}

.sidebar-column {

width: calc(25% - 20px);

}

}

@media (min-width:768px) {

.container {

flex-direction: row;

}

.left-sidebar {

width: 20%;

height: 600px;

flex-direction: column;

}

.main-element {

width: 65%;

padding: 0 10px;

}

.sidebar-column {

width: calc(100% - 20px);

margin: 10px;

}

.right-sidebar {

width: 15%;

height: 600px;

}

}

</style>

</head>

<body>

<h2>Responsive website using media query</h2>

<!-- Header -->

<div class="header"></div>

<div class="container">

<!-- Left sidebar -->

<div class="left-sidebar">

<div class="sidebar-column"></div>

<div class="sidebar-column"></div>

<div class="sidebar-column"></div>

<div class="sidebar-column"></div>

</div>

<!-- Main element -->

<div class="main-element">

<div class="main-column"></div>

<div class="main-column"></div>

<div class="main-column"></div>

<div class="main-column"></div>

<div class="main-column"></div>

</div>

<!-- Right sidebar -->

<div class="right-sidebar"></div>

</div>

</body>

</html>

HTML

[▶ Try it](https://www.tutorialstonight.com/online-html-editor.php?p=css&q=media-query-responsive-website" \t "_blank)

Output:

## Using Media Query in HTML

Media queries are not only applied in CSS but can be directly applied in HTML to load resources based on media type and media features.

### Loading different external CSS based on viewport size

<link> element used for loading external CSS file to the webpage can use media attribute to load CSS file optionally.

<!-- Loaded to all users -->

<link rel="stylesheet" href="all.css" media="all">

<!-- Loaded to screens 320px or more wide -->

<link rel="stylesheet" href="small.css" media="(min-width: 320px)">

<!-- Loaded to screens 768px or more wide -->

<link rel="stylesheet" href="medium.css" media="(min-width: 768px)">

<!-- Loaded to print medias -->

<link rel="stylesheet" href="print.css" media="print">

HTML

You can see in the above code, you can load different CSS files based on device type or viewport size. It improves the page targeting for the different device as CSS loaded is more specific to the device.

But, this is not always the case when a small device like mobile visits the page then it loads the only file that satisfies the condition of a media query, while when you visit a device with a bigger viewport then it loads all CSS files for a matching device.

Not only in CSS files but you can use it load different <source> in <picture> element.

<picture>

<!-- Load this image if the screen is at least 600px wide -->

<source srcset="logo.webp" media="(min-width: 600px)">

<!-- Load this image if the screen is at least 768px wide -->

<source srcset="logo.webp" media="(min-width: 768px)">

<!-- Load this image if nothing matches -->

<img src="logo.webp" alt="A logo">

</picture>

HTML

## Using Media Query in JavaScript

You can use media query in JavaScript also. You can create a media condition that targets a specific media type or media feature.

Use window.matchMedia() to define the condition.

// it is a media condition that target 768px or more wide viewport

const mediaQuery = window.matchMedia("(min-width: 768px)");

if (mediaQuery.matches) {

console.log("Query matched!");

}

JavaScript

## Conclusion

Nowadays it is necessary for your website to be responsive. And if you want to make responsive websites then media query is at the core. For responsive websites, the most famous choice is [bootstrap framework](https://www.tutorialstonight.com/bootstrap/bootstrap-introduction.php" \t "_blank), even bootstrap use media query at its code for responsiveness.

Points to remember:

1. Media query can conditionally apply CSS properties to certain elements of the webpage when the device type of viewport dimension matches.
2. It can load different resources only for a targeted device or targeted screen size. like in the <picture> element you can set multiple sources of an image that will load when a certain media condition satisfies.
3. It will load an [external CSS](https://www.tutorialstonight.com/css/css-implementation.php" \l "external-css" \t "_blank) file that is attached for a media query even when the query is false in <link> element but will not apply its property to the page until the condition is true.
4. Using window.matchMedia media query can also be used in JavaScript.

Top of Form

# How to center a div in CSS

As a **web developer**, one of the most common thing that you will most frequently need to do is to center a div element using [CSS](https://www.tutorialstonight.com/css/css-introduction.php" \t "_blank).

In this section, we will look at 15 different ways by which you can center a div horizontally and vertically on a webpage.

Table Of Contents

1. [Center horizontally](https://www.tutorialstonight.com/how-to-center-a-div-in-css.php" \l "center-horizontal)
   1. [Using align property](https://www.tutorialstonight.com/how-to-center-a-div-in-css.php" \l "using-align)
   2. [Using flexbox (justify-content)](https://www.tutorialstonight.com/how-to-center-a-div-in-css.php" \l "using-flexbox)
   3. [Position absolute and margin auto](https://www.tutorialstonight.com/how-to-center-a-div-in-css.php" \l "position-abolute-and-margin-auto)
   4. [Position absolute and translateX](https://www.tutorialstonight.com/how-to-center-a-div-in-css.php" \l "position-abolute-and-transform)
   5. [Using table](https://www.tutorialstonight.com/how-to-center-a-div-in-css.php" \l "using-table)
   6. [Using CSS table property](https://www.tutorialstonight.com/how-to-center-a-div-in-css.php" \l "using-display-table-cell)
2. [Center vertically](https://www.tutorialstonight.com/how-to-center-a-div-in-css.php" \l "center-vertical)
   1. [Using flexbox (align-items)](https://www.tutorialstonight.com/how-to-center-a-div-in-css.php" \l "using-flexbox-v)
   2. [Position absolute and margin auto](https://www.tutorialstonight.com/how-to-center-a-div-in-css.php" \l "position-abolute-and-margin-auto-v) (vertical center)
   3. [Position absolute and translateY](https://www.tutorialstonight.com/how-to-center-a-div-in-css.php" \l "position-abolute-and-transform-v) (vertical center)
   4. [Using table](https://www.tutorialstonight.com/how-to-center-a-div-in-css.php" \l "using-table-v) (vertical center)
   5. [Using CSS table property](https://www.tutorialstonight.com/how-to-center-a-div-in-css.php" \l "using-display-table-cell-v) (vertical center)

Let's see how you can align elements to center horizontally. We have discussed 5 different ways for it, of which margin auto and flexbox is the most commonly used method.

## 1. Center div using margin method

To center a div element horizontally you can simply use the [margin](https://www.tutorialstonight.com/css/css-margins.php" \t "_blank) property of CSS. It is the easiest way to align a div to center.

Give margin-left: auto and margin-right: auto to the element or you can use shorthand margin: 0 auto, this will automatically adjust the element to center of its parent element horizontally by leaving equal spaces to its left and right side.

Here is an example in which a child element is aligned to center using margin property.

.parent {

padding: 20px;

background: #7E89B4;

border: 2px solid #4B5681;

}

.child {

width: 200px;

color: white;

height: 100px;

text-align: center;

background: #353C5A;

font-family: sans-serif;

}

.center {

margin: 0 auto;

}

CSS

Here is the HTML code, the margin property is applied to the child element.

<div class="parent">

<div class="child center">Child</div>

</div>

HTML

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=article&q=center-div-margin" \t "_blank)

Note: To align any element to center horizontally you need to keep 2 things in mind:

1. The element must be a block label element. If the **display property** of an element is inline or inline-block then it will not center using align property.
2. The width of the element should be smaller than it's parent element. If the element is itself bigger than it's parent than will be unable to get equal spacing from it's left and right side and hence won't center.

## 2. Align center using flexbox

One of the best ways to align the div element to center horizontally is by using the [flexbox](https://www.tutorialstonight.com/css/css-flexbox.php" \t "_blank).

Using flexbox you can position any element in 2D (horizontal and vertical). Here we will center horizontally.

First, add display: flex property **to the parent of div element** to make it a flex container, then use the [justify-content](https://www.tutorialstonight.com/css/css-flexbox.php" \l "justify-content" \t "_blank) property to aligning center.

Now add justify-content: center to align the parent element to center div horizontally.

Remember the **flex property should be applied to the parent element**. Here is the CSS code for example.

.parent {

border: 2px solid #4B5681;

background: #7E89B4;

padding: 20px;

}

.child {

height: 100px;

width: 200px;

background: #353C5A;

font-family: sans-serif;

text-align: center;

color: white;

}

.center {

display: flex;

justify-content: center;

}

CSS

This is the HTML code.

<div class="parent center">

<div class="child">Child</div>

</div>

HTML

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=article&q=div-center-flexbox" \t "_blank)

## 3. Position absolute and margin auto

An element having an absolute position is like a floating element in the parent element. You can move it wherever you want.

To align the child element center horizontally you need to apply left: 0, right: 0 and margin-left: auto, margin-right: auto to the element (margin: 0 auto shorthand).

.parent {

position: relative;

border: 2px solid #4B5681;

background: #7E89B4;

height: 100px;

}

.child {

position: absolute;

left:0;

right: 0;

margin: 0 auto;

height: 100px;

width: 200px;

background: #353C5A;

font-family: sans-serif;

text-align: center;

color: white;

}

CSS

<div class="parent center">

<div class="child">Child</div>

</div>

HTML

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=article&q=center-horizontal-margin-auto-position-absolute" \t "_blank)

## 4. Center horizontally using position absolute and translateX

Using [position property](https://www.tutorialstonight.com/css/css-position.php" \t "_blank) and translateX you can align element horizontally as well as vertically to center. Here we will see horizontal alignment.

For this, the position of the parent element should be relative and the position of the child element should be absolute.

.parent {

position: relative;

}

.child {

position: absolute;

}

CSS

Now, shift the child element 50% to its left.

This left shift of 50% will make the child element to start from midway to the right side of the parent element.

You can see in the above picture we want to center the box horizontally but it starts from the middle of the parent element. So we can apply the [translateX](https://www.tutorialstonight.com/css/css-2D-transformation.php" \t "_blank) property and move the child element 50% in a negative direction which will align it in center. Now add transform: translateX(-50%) to finally center the child element.

.parent {

position: relative;

}

.child {

position: absolute;

left: 50%;

transform: translateX(-50%);

}

CSS

This is the HTML code.

<div class="parent">

<div class="child">Child</div>

</div>

HTML

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=article&q=div-center-using-position-absolute" \t "_blank)

## 5. Center using table

Using a [table](https://www.tutorialstonight.com/html/html-tables.php" \t "_blank) to design page layout is not used nowadays, but it's worth it to know that you can use this to center the element horizontally.

To center an element horizontally in the table use text-align: center to parent and display: inline-block to the child element.

To see the code in effect width of the child should be less than the parent element so we have set the parent width to 100vw for understanding purpose.

Here is HTML and CSS code as an example.

.parent {

text-align: center;

width: 100vw;

border: 2px solid #4B5681;

background: #7E89B4;

height: 100px;

}

.child {

display: inline-block;

height: 100px;

width: 200px;

background: #353C5A;

font-family: sans-serif;

color: white;

}

CSS

<table>

<tbody>

<tr>

<td class="parent">

<div class="child">Child</div>

</td>

</tr>

</tbody>

</table>

HTML

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=article&q=center-horizontally-using-table" \t "_blank)

## 6. Using display table-cell

CSS enables us to change ordinary elements into the effect of table elements. Using this feature and concept used in the above approach you can center the element horizontally.

Set display: table-cell and text-align: center to the parent element and display: inline-block to the child element.

.parent {

display: table-cell;

text-align: center;

width: 100vw;

border: 2px solid #4B5681;

background: #7E89B4;

height: 100px;

}

.child {

display: inline-block;

height: 100px;

width: 200px;

background: #353C5A;

font-family: sans-serif;

color: white;

}

CSS

<div class="parent">

<div class="child">Child</div>

</div>

HTML

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=article&q=center-horizontally-using-table-cell" \t "_blank)

Now we will see methods and modification of the above methods to align element in center vertically. For most cases you can refer above methods of horizontal alignment and add a few more property to center it vertically also, but anyway we will discuss in detail.

## 1. Vertical align using flexbox

Using flexbox you can align elements both in the horizontal and vertical direction. Here we will align element in the vertical direction.

First, you need to define display: flex to the parent element. Now the element inside this parent element becomes flex items.

To align child element vertically use align-items property and set its value as center.

.parent {

display: flex;

align-items: center;

border: 2px solid #4B5681;

background: #7E89B4;

height: 200px;

}

.child {

height: 100px;

width: 200px;

background: #353C5A;

font-family: sans-serif;

color: white;

}

CSS

<div class="parent">

<div class="child">Child</div>

</div>

HTML

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=article&q=center-vertical-flebox" \t "_blank)

## 2. Absolute position and margin auto

We already know the element having an absolute position is can be moved very easily within the webpage. Using this we can also align to the center.

To align the child element center vertically you need to apply top: 0, bottom: 0 and margin-top: auto, margin-bottom: auto to the element (margin: auto 0 shorthand).

.parent {

position: relative;

border: 2px solid #4B5681;

background: #7E89B4;

height: 200px;

}

.child {

position: absolute;

top: 0;

bottom: 0;

margin: auto 0;

height: 100px;

width: 200px;

background: #353C5A;

font-family: sans-serif;

text-align: center;

color: white;

}

CSS

<div class="parent center">

<div class="child">Child</div>

</div>

HTML

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=article&q=center-vertical-margin-auto-position-absolute" \t "_blank)

## 3. Using position absolute and translateY

In the same way, you centred the element horizontally using position absolute and translateX, you can use translateY to center the element vertically.

To do this set position: relative to the parent element and position: absolute to child element and top: 50%.

This will flow child element to start from verticle middle of the parent element, which is not center but you can use transform: translateY(-50%) to shift it up by 50% which will make it verticle align to center.

.parent {

position: relative;

}

.child {

position: absolute;

top: 50%;

transform: translateY(-50%);

}

CSS

<div class="parent">

<div class="child">Child</div>

</div>

HTML

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=article&q=verticle-align-using-position-absolute" \t "_blank)

## 4. Center vertically using table

As we have discussed above you can also use the table to create layout and center element (not recommended).

To center an element vertically in table use vertical-align: middle to parent and display: inline-block to the child element.

To see the code in effect height of the child should be less than the height parent element so we have set the parent height to 200px for understanding purpose.

.parent {

vertical-align: middle;

border: 2px solid #4B5681;

background: #7E89B4;

height: 200px;

}

.child {

display: inline-block;

height: 100px;

width: 200px;

background: #353C5A;

font-family: sans-serif;

color: white;

}

CSS

<table>

<tbody>

<tr>

<td class="parent">

<div class="child">Child</div>

</td>

</tr>

</tbody>

</table>

HTML

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=article&q=center-vertically-using-table" \t "_blank)

## 5. Vertical align using display table-cell

By using CSS by changing element behaviours as the table you can use the above method without using table elements.

Set display: table-cell and text-align: center to the parent element and display: inline-block to the child element.

.parent {

display: table-cell;

vertical-align: middle;

height: 200px;

border: 2px solid #4B5681;

background: #7E89B4;

}

.child {

display: inline-block;

height: 100px;

width: 200px;

background: #353C5A;

font-family: sans-serif;

color: white;

}

CSS

<div class="parent">

<div class="child">Child</div>

</div>

HTML

[▶ Run the code](https://www.tutorialstonight.com/online-html-editor.php?p=article&q=center-vertically-using-table-cell" \t "_blank)

## Conclusion

You can use the above methods to center elements vertically or horizontally. You can also use the table method to center elements vertically or horizontally.

Best way to ceter elements both horizontally and vertically is to use flexbox.

If you want to center elements vertically or horizontally using table elements, you can use the above methods or use display: table-cell and text-align: center to the parent element and display: inline-block to the child element.

Points to remember:

1. Flexbox is a modern way to layout element. It is recommended to use it to layout your elements in 2-dimension. Here is a [complete guide to](https://www.tutorialstonight.com/css/css-flexbox.php" \t "_blank) **[CSS flexbox](https://www.tutorialstonight.com/css/css-flexbox.php" \t "_blank)**.

Learn

[HTML5](https://www.tutorialstonight.com/html/html-introduction.php) [CSS3](https://www.tutorialstonight.com/css/css-introduction.php) [JavaScript](https://www.tutorialstonight.com/js/) [Bootstrap 4](https://www.tutorialstonight.com/bootstrap/bootstrap-introduction.php) [Python](https://www.tutorialstonight.com/python/python-tutorial.php)

# 5 ways how to center an image in CSS

In this article, we will look at 5 different ways on how to center an image in CSS or to center any non-block element.

Human beings are attracted to symmetry, even the basic definition that your mind conceives about beauty is symmetry.

So as a web developer you might want your webpage to follow symmetry and look beautiful. For this one task you have is to **center an image horizontally and/or vertically**.

We have discussed the following 5 different ways to center an image in CSS.

1. [Using margin auto](https://www.tutorialstonight.com/how-to-center-an-image-in-css.php" \l "center-image-using-margin)
2. [Using text-align center](https://www.tutorialstonight.com/how-to-center-an-image-in-css.php" \l "center-image-using-text-align)
3. [Using flexbox](https://www.tutorialstonight.com/how-to-center-an-image-in-css.php" \l "center-image-using-flexbox)
4. [Using absolute & transform properties](https://www.tutorialstonight.com/how-to-center-an-image-in-css.php" \l "center-image-using-absolute-and-transform)
5. [Using center element](https://www.tutorialstonight.com/how-to-center-an-image-in-css.php" \l "center-image-using-center-element) (deprecated)

## 1. Center an image using the margin property

Using [margin property](https://www.tutorialstonight.com/css/css-margins.php" \t "_blank) you can horizontally center any block-level HTML element. Set margin-left: auto and margin-right: auto or simply margin: 0 auto and your element will get aligned to center horizontally.

But an image is an **inline element by default** so you can not directly apply this to an image in HTML.

Follow the following steps:

1. First, set display: block to the image element to make the image a block-level element because margin auto will only work on block-level element.
2. Second, set the left and right margins of the image to "auto" by margin: 0 auto.
3. Finally, set the width of the image smaller than it's parent element because if the image is bigger than it's parent element then it doesn't worth centring it. Let's set width: 60% for the safe side.

.center {

display: block;

margin: 0 auto;

width: 60%;

}

CSS

Now, adding 'center' class to the 'img' element.

<img src="nebula.webp" class="center" alt="nebula image">

HTML

[▶ Try it](https://www.tutorialstonight.com/online-html-editor.php?p=article&q=center-image-using-margin" \t "_blank)

Output:

## 2. Center an image using text-align

Yes, you can align an image horizontally to center using the text-align property.

Since text-align property works on block level element so warp your image with a <div> which is block level element and set text-align: center to the <div> element.

Make sure the size of the viewport is smaller than the image width unless it won't work so you can set some width to the image.

<div style="text-align: center">

<img src="nebula.jpg" width="60%" alt="nebula image">

</div>

HTML

[▶ Try it](https://www.tutorialstonight.com/online-html-editor.php?p=article&q=center-image-using-text-align" \t "_blank)

Output:

## 3. Center an image using flexbox

Using the above 2 methods you can center an image only horizontally but using [CSS flexbox](https://www.tutorialstonight.com/css/css-flexbox.php" \t "_blank) you can center the image both horizontally and vertically.

To use flexbox for this purpose you first need to wrap the image in an <div> element and set the display property to "flex" to create a flex container.

Now, to center the image horizontally or vertically you need to use 2 flexbox property justify-content and align-items respectively.

### center image horizontally using flexbox

To center the image horizontally use the [justify-content property of CSS flexbox](https://www.tutorialstonight.com/css/css-flexbox.php" \l "justify-content" \t "_blank). The justify-content align the items of the flex container in the X-axis.

Set justify-content: center to the flex container.

Make sure the width of the image is smaller than the viewport. Here we set the width of the image to 60%.

.center {

display: flex;

justify-content: center;

}

CSS

Now set the 'center' class to the div element.

<div class="center">

<img src="nebula.jpg" width="60%" alt="nebula image">

</div>

HTML

[▶ Try it](https://www.tutorialstonight.com/online-html-editor.php?p=article&q=center-image-horizontally-using-flexbox" \t "_blank)

Output:

### center image vertically using flexbox

To center the image vertically use the [align-items property of CSS flexbox](https://www.tutorialstonight.com/css/css-flexbox.php" \l "justify-content" \t "_blank). The align-item property aligns the items of the flexbox in the Y-axis which will vertically align the image.

Set align-items: center to the <div> element.

Here we also set some height of element greater than the height of the image to visualize properly.

.center {

display: flex;

align-items: center;

height: 400px;

background: #e7e8e9;

}

CSS

Now set the 'center' class to div element.

<div class="center">

<img src="nebula.jpg" width="60%" alt="nebula image">

</div>

HTML

[▶ Try it](https://www.tutorialstonight.com/online-html-editor.php?p=article&q=center-image-vertically-using-flexbox" \t "_blank)

Output:

## 4. Center image using position and transform

Using position absolute and then applying transform property on it is a classic method to center elements in CSS.

Using this you can center images or any element both horizontally and vertically.

For this, to work there must be a parent element of the image with position relative.

### center image horizontally using position and transform

Step 1: Wrap the image element in a div element and set position: relative to it. Also, set some height greater than the image height for it to work properly.

Step 2: Now set position: absolute to the image which will move the image to start from (x, y): (0, 0) of its parent element.

Step 3: Due to absolute position the element within can float easily so set left: 50%. Now the element will start from (x, y) : (50%, 0).

Step 4: Finally translate the image 50% of its size in negative X direction using transform: translateX(-50%) this will align the image center horizontally.

Here is the complete code to align horizontally using position absolute and transform property.

.parent {

position: relative;

height: 350px;

background: #e7e8e9;

}

.center {

position: absolute;

left: 50%;

transform: translateX(-50%);

}

CSS

Now set the 'center' class to div element.

<div class="parent">

<img class="center" src="nebula.jpg" width="60%" alt="nebula image">

</div>

HTML

[▶ Try it](https://www.tutorialstonight.com/online-html-editor.php?p=article&q=center-image-horizontally-using-position" \t "_blank)

Output:

### center image vertically using position and transform

To align the image vertically you have to shift 50% from to and then translateY 50% in negative. As the parent element is already positioned, relative.

Step 1: Set position: absolute to the image. Image will float to (x, y) : (0, 0) of its parent element.

Step 2: Shift image 50% from top top: 50%. Now the element will start from (x, y) : (0, 50%).

Step 3: Finally translate the image 50% of its size in negative Y direction using transform: translateY(-50%) this will align the image center vertically.

Here is the complete code to align vertically using position absolute and transform property.

.parent {

position: relative;

height: 400px;

background: #e7e8e9;

}

.center {

position: absolute;

top: 50%;

transform: translateY(-50%);

}

CSS

Now set the 'center' class to div element.

<div class="parent">

<img class="center" src="nebula.jpg" width="60%" alt="nebula image">

</div>

HTML

[▶ Try it](https://www.tutorialstonight.com/online-html-editor.php?p=article&q=center-image-vertically-using-position" \t "_blank)

Output:

## 5. Center image using center element (deprecated)

Using <center> element is deprecated and not used in HTML5 but some browser still supports it.

To center the image horizontally wrap the image in <center> element.

Note: Using <center> element is not recommended.

<center><img class="center" src="assets/articles/nebula-by-nasa.jpg" width="60%" alt="nebula image"></center>

HTML

[▶ Try it](https://www.tutorialstonight.com/online-html-editor.php?p=article&q=center-image-using-center-element" \t "_blank)

Output:

### Conclusion

As a web-developer you constantly need to align images horizontally and vertically, we discussed here 5 different techniques to center an image using CSS.

Related Posts:

[How to center div in CSS](https://www.tutorialstonight.com/how-to-center-a-div-in-css.php" \t "_blank) [A complete guide to CSS flexbox](https://www.tutorialstonight.com/css/css-flexbox.php" \t "_blank)

Learn

[HTML5](https://www.tutorialstonight.com/html/html-introduction.php) [CSS3](https://www.tutorialstonight.com/css/css-introduction.php) [JavaScript](https://www.tutorialstonight.com/js/) [Bootstrap 4](https://www.tutorialstonight.com/bootstrap/bootstrap-introduction.php) [Python](https://www.tutorialstonight.com/python/python-tutorial.php)