CSS

**What You Can Do with CSS**

There are lot more things you can do with CSS.

* You can easily apply same style rules on multiple elements.
* You can control the presentation of multiple pages of a website with a single style sheet.
* You can present the same page differently on different devices.
* You can style dynamic states of elements such as hover, focus, etc. that isn't possible otherwise.
* You can change the position of an element on a web page without changing the markup.
* You can alter the display of existing HTML elements.
* You can transform elements like scale, rotate, skew, etc. in 2D or 3D space.
* You can create animations and transitions effects without using any JavaScript.
* You can create print friendly version of your web pages.

Cascading Style Sheets, fondly referred to as CSS, is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page.



There are three types of CSS which are given below:

* Inline CSS
* Internal or Embedded CSS
* External CSS

**Inline CSS:** Inline CSS contains the CSS property in the body section attached with element is known as inline CSS. This kind of style is specified within an HTML tag using the style attribute.

**Example:**

<!DOCTYPE html>

<html>

    <head>

        <title>Inline CSS</title>

    </head>

    <body>

        <p style = "color:#009900; font-size:50px;

                font-style:italic; text-align:center;">

Hello World

        </p>

    </body>

</html>

**Internal or Embedded CSS:** This can be used when a single HTML document must be styled uniquely. The CSS rule set should be within the HTML file in the head section i.e the CSS is embedded within the HTML file.  
**Example:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Example of CSS Embedded Style Sheet</title>

<style>

body { background-color: YellowGreen; }

p { color: #fff; }

</style>

</head>

<body>

<h1>This is a heading</h1>

<p>This is a paragraph of text.</p>

</body>

</html>

Example 2: with class and ID

|  |
| --- |
| <!DOCTYPE html>  <html>      <head>          <title>Internal CSS</title>          <style>              .main {                  text-align:center;              }              .GFG {                  color:#009900;                  font-size:50px;                  font-weight:bold;              }              .google {                  font-style:bold;                  font-size:20px;              }          </style>      </head>      <body>          <div class = "main">              <div class ="GFG">HEllo</div>                <div class =" google ">                  A computer science portal for geeks              </div>          </div>      </body>  </html> |

**External CSS:** External CSS contains separate CSS file which contains only style property with the help of tag attributes (For example class, id, heading, … etc). CSS property written in a separate file with .css extension and should be linked to the HTML document using **link** tag. This means that for each element, style can be set only once and that will be applied across web pages.

Example1:

### Linking External Style Sheets

Before linking, we need to create a style sheet first. Let's open your favorite code editor and create a new file. Now type the following CSS code inside this file and save it as "style.css".

body

{

background: lightyellow; font: 18px Arial, sans-serif;

}

h1

{

color: orange;

}

An external style sheet can be linked to an HTML document using the [<link>](https://www.tutorialrepublic.com/html-reference/html-link-tag.php) tag. The <link> tag goes inside the [<head>](https://www.tutorialrepublic.com/html-reference/html-head-tag.php) section, as you can see in the following example:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Example of CSS External Style Sheet</title>

<link rel="stylesheet" href="/examples/css/style.css">

</head>

<body>

<h1>This is a heading</h1>

<p>This is a paragraph of text.</p>

</body>

</html>

**Example 2:** The file given below contains CSS property. This file save with .css extension. For Ex: **first.css**

body {

background-color:powderblue;

}

.main {

text-align:center;

}

.GFG {

color:#009900;

font-size:50px;

font-weight:bold;

}

#google {

font-style:bold;

font-size:20px;

}

Below is the HTML file that is making use of the created external style sheet

* **link** tag is used to link the external style sheet with the html webpage.
* **href** attribute is used to specify the location of the external style sheet file.

|  |
| --- |
| <!DOCTYPE html>  <html>      <head>          <link rel="stylesheet" href="first.css"/>      </head>        <body>          <div class = "main">              <div class ="GFG">Hello world</div>              <div id ="google">                  A computer science portal for geeks              </div>          </div>      </body>  </html> |

# CSS Selectors

## What is Selector?

A CSS selector is a pattern to match the elements on a web page. The style rules associated with that selector will be applied to the elements that match the selector pattern.

Selectors are one of the most important aspects of CSS as they allow you to target specific elements on your web page in various ways so that they can be styled.

## Universal Selector

The universal selector, denoted by an asterisk (\*), matches every single element on the page.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Example of CSS universal selector</title>

<style>

\* {

margin: 0;

padding: 0;

}

</style>

</head>

<body>

<h1>This is heading</h1>

<p>This is a paragraph.</p>

</body>

</html>

Example2:

<!DOCTYPE html>

<html>

<head>

<title>\* selector</title>

<!-- CSS property for \* selector -->

<style>

\* {

background: green;

font-weight:bold;

margin-left:70px;

color:white;

}

</style>

</head>

<body>

<h1>HEllo</h1>

<h2>\*(Universal) Selector</h2>

<ul>

<li>Data Structure</li>

<li>Computer Network</li>

<li>Operating System</li>

</ul>

<ol>

<li>Java</li>

<li>Ruby</li>

<li>Pascal</li>

</ol>

</body>

</html>

## Id Selectors

The id selector is used to define style rules for a single or unique element.

The id selector is defined with a hash sign (#) immediately followed by the id value.

## Class Selectors

The class selectors can be used to select any HTML element that has a class attribute. All the elements having that class will be formatted according to the defined rule.

The class selector is defined with a period sign (.) immediately followed by the class value.

## Child Selectors

A child selector is used to select only those elements that are the direct children of some element.

A child selector is made up of two or more selectors separated by a greater than symbol (>). You can use this selector, for instance, to select the first level of list elements inside a nested list that has more than one level. Let's check out an example to understand how it works:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Example of CSS Child Selectors</title>

<style>

ul > li {

list-style: square;

}

ul > li ol {

list-style: none;

}

</style>

</head>

<body>

<ul>

<li><a href="#">Home</a></li>

<li><a href="#">About</a></li>

<li>

<a href="#">Services</a>

<ol>

<li><a href="#">Design</a></li>

<li><a href="#">Development</a></li>

</ol>

</li>

<li><a href="#">Contact</a></li>

</ul>

</body>

</html>

## Grouping Selectors

Often several selectors in a style sheet share the same style rules declarations. You can group them into a comma-separated list to minimize the code in your style sheet. It also prevents you from repeating the same style rules over and over again.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Example of CSS Grouping Selectors</title>

<style>

h1, h2, h3 {

font-weight: normal;

}

h1 {

font-size: 36px;

}

h2 {

font-size: 28px;

}

h3 {

font-size: 22px;

}

</style>

</head>

<body>

<h1>This is a heading of level 1</h1>

<h2>This is a heading of level 2</h2>

<h3>This is a heading of level 3</h3>

</body>

</html>

# CSS | Background

The CSS background properties are used to define the background effects for elements.

Css background properties are as follows :

1. Background-color
2. Background-image
3. Background-repeat
4. Background-attachment
5. Background-position
6. **Background color :**This property specifies the background color of an element.  
   Syntax :
7. body
8. {
9. background-color:color name

}

A color name can also be given as : “green”, a HEX value as “#5570f0”, an RGB value as “rgb(25, 255, 2)”.  
**Example :**

<style>

h1{

background-color: blue;

opacity: 0.3;

}

</style>

<body>

<h1>HEllo</h1>

</body>

**Background Image :**This property specify an image to use as the background of an element. By default, the image is repeated so it covers the entire element.  
Syntax:

body

{

background-image : link;

}

**Example:**

<style>

body{

background-image: url("paper.gif");

}

</style>

<body>

<h1>HEllo</h1>

</body>

**Background repeat :**By default the background image property repeats the image both horizontally and vertically.  
**To repeat an image horizontally:**  
**Syntax:**

body

{

background-image:link;

background-repeat: repeat-x/ repeat-y/ no-repeat/ repeat/ space/ round/ intial;

}

<!DOCTYPE html>

<html>

<head>

<style>

body {

background-image: url("gradient\_bg.png");

background-repeat: repeat-y;

}

</style>

</head>

<body>

<h1>Hello World!</h1>

<p>Here, a background image is repeated only horizontally!</p>

</body>

</html>

**Background-position :**This property is used to set the image to a particular position.  
**Syntax :**

body

{

background-repeat:no repeat;

background-position:left top;

}

## Property Values

left top  
left center  
left bottom  
right top  
right center  
right bottom  
center top  
center center  
center bottom

initial

inherit

x% y% e.g. 50% 50%

px px e.g. 10px 100px

## Background Attachment

The background-attachment property determines whether the background image is fixed with regard to the viewport or scrolls along with the containing block.

Property Values

|  |  |
| --- | --- |
| **Value** | **Description** |
| scroll | The background image will scroll with the page. This is default |
| Fixed | The background image will not scroll with the page |
| initial | Sets this property to its default value. [Read about *initial*](https://www.w3schools.com/cssref/css_initial.asp) |

<!DOCTYPE html>

<html>

<head>

<style>

body {

background-image: url("img\_tree.gif");

background-repeat: no-repeat;

background-attachment: scroll;

}

</style>

</head>

<body>

<p>The background-image scrolls with the page. Try to scroll down.</p>

<p>The background-image scrolls with the page. Try to scroll down.</p>

<p>If you do not see any scrollbars, try to resize the browser window.</p>

</body>

</html>

## The Background Shorthand Property

When using the background shorthand property the order of the property values should be.

background: *color* *image* *repeat* *attachment* *position*;

# CSS Text

The commonly used text properties are: text-align, text-decoration, text-transform, text-indent,

**Text Color**

The color of the text is defined by the CSS color property.

Color: red

**Text Alignment**

The [text-align](https://www.tutorialrepublic.com/css-reference/css-text-align-property.php) property is used to set the horizontal alignment of the text.

Text can be aligned in four ways: **to the left, right, centre or justified** (straight left and right margins).

Text:Align: left

**Text Decoration**

The [text-decoration](https://www.tutorialrepublic.com/css-reference/css-text-decoration-property.php) property is used to set or remove decorations from text.

This property typically accepts one of the following values: underline, overline, line-through, and none.

**Text Transformation**

The **Text-Transform** Property Is Used To Set The Cases For A Text.

Uppercase, capitalize, lowercase.

**TEXT DIRECTION**  
Text direction property is used to set the direction of the text.  
 **The direction can be set by using rtl : right to left .**

**Ltr:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Setting Text Color using CSS</title>

<style>

h1 {

color: #ff0000;

text-decoration:overline;

text-align: center;

direction:rtl;

}

p {

color: green;

}

</style>

</head>

<body>

<h1>This is a heading</h1>

<p>This is a simple paragraph of text.</p>

</body>

</html>

# CSS font Property

It is a shorthand property for setting the individual font properties i.e. [font-style](https://www.tutorialrepublic.com/css-reference/css-font-style-property.php), [font-variant](https://www.tutorialrepublic.com/css-reference/css-font-variant-property.php), [font-weight](https://www.tutorialrepublic.com/css-reference/css-font-weight-property.php), [font-size](https://www.tutorialrepublic.com/css-reference/css-font-size-property.php), [line-height](https://www.tutorialrepublic.com/css-reference/css-line-height-property.php) and [font-family](https://www.tutorialrepublic.com/css-reference/css-font-family-property.php).

font-style: normal | italic | oblique | [initial](https://www.tutorialrepublic.com/definitions.php#initial) | [inherit](https://www.tutorialrepublic.com/definitions.php#inherit)

**font size** in pixel values (e.g. 14px, 16px, etc.)

or in em where 1em = 20px

or in percentage.

Or keywords: keywords: xx-small, x-small, small, medium, large, x-large, xx-large.

|  |  |
| --- | --- |
| font-weight: | normal | bold | bolder | lighter | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | [initial](https://www.tutorialrepublic.com/definitions.php#initial) | [inherit](https://www.tutorialrepublic.com/definitions.php#inherit) |

|  |  |
| --- | --- |
| font-variant: | normal | small-caps | [initial](https://www.tutorialrepublic.com/definitions.php#initial) | [inherit](https://www.tutorialrepublic.com/definitions.php#inherit) |

# CSS Lists

ul {

list-style-type: square/ circle/disc;

}

ol {

list-style-type: upper-roman;

}

* circle
* decimal , eg :1,2,3,etc
* decimal-leading-zeroes , eg :01,02,03,04,etc
* lower-roman
* upper-roman
* lower-alpha, eg : a,b,c,etc
* upper-alpha, eg : A,B,C,etc
* square

List style position:

list-style-position: inside/ outside;

List style Image

list-style-image: url("images/bullet.png");

<!DOCTYPE>

<html>

<head>

<style>

ul.a

{

list-style-type:square;

}

ol.c

{

list-style-type:lower-alpha;

}

</style>

</head>

<body>

<h2>

GEEKSFORGEEKS

</h2>

<p>

Unordered lists

</p>

<ul class="a">

<li>one</li>

<li>two</li>

<li>three</li>

</ul>

<ul class="b">

<li>one</li>

<li>two</li>

<li>three</li>

</ul>

<p>

Ordered Lists

</p>

<ol class="c">

<li>one</li>

<li>two</li>

<li>three</li>

</ol>

<ol class="d">

<li>one</li>

<li>two</li>

<li>three</li>

</ol>

</body>

</html>

Shorthand –Property: list-style

|  |  |  |
| --- | --- | --- |
| :lt() | $("ul li:lt(3)") | Select all <li> elements at an index less than three within the matched set (i.e. selects 1st, 2nd, 3rd list elements), zero-based index. |
| :gt() | $("ul li:gt(3)") | Select all <li> elements at an index greater than three within the matched set (i.e. selects 4th, 5th, ... list elements), zero-based index. |

ul.a li:nth-child(odd) {

background: red;

}

CSS Padding

The CSS padding properties are used to generate space around an element's content, inside of any defined borders.

CSS has properties for specifying the padding for each side of an element:

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

<!DOCTYPE html>

<html>

<head>

<style>

div {

border: 1px solid black;

background-color: lightblue;

padding-top: 50px;

padding-right: 30px;

padding-bottom: 50px;

padding-left: 80px;

}

</style>

</head>

<body>

<h2>Using individual padding properties</h2>

<div>This div element has a top padding of 50px, a right padding of 30px, a bottom padding of 50px, and a left padding of 80px.</div>

</body>

</html>

div {

border-width: 1px ;

border-style:solid;

border-color:green;

border-radius:23px;

background-color: lightblue;

padding-top: 70px;

padding-right: 30px;

padding-bottom: 50px;

padding-left: 80px;

margin-left:50px;

margin:right:70px;

margin:50px 30px 40px 70px;;

}

**Styling Links with CSS**

A link has four different states — link, visited, active and hover. These four states of a link can be styled differently through using the following anchor pseudo-class selectors.

* **a:link** — define styles for normal or unvisited links.
* **a:visited** — define styles for links that the user has already visited.
* **a:hover** — define styles for a link when the user place the mouse pointer over it.
* **a:active** — define styles for links when they are being clicked.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Styling Different Link States using CSS</title>

<style>

/\* unvisited link \*/

a:link {

color: blue;

text-decoration: none;

border-bottom: 1px solid;

}

/\* visited link \*/

a:visited {

color: #purple;

}

/\* mouse over link \*/

a:hover {

color: #00ff00;

border-bottom: none;

}

/\* active link \*/

a:active {

color: red;

}

</style>

</head>

<body>

<p><a href="https://www.tutorialrepublic.com/" target="\_top">Visit Tutorial Republic</a></p>

</body>

</html>

### Removing the Default Underline from Links

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Remove Default Underline from Links using CSS</title>

<style>

a:link, a:visited {

text-decoration: none;

}

a:hover, a:active {

text-decoration: underline;

}

</style>

</head>

<body>

<p><a href="#">Place mouse pointer over me</a></p>

</body>

</html>

**Making Text Links Look Like Buttons**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Customize a Link as Button using CSS</title>

<style>

a:link, a:visited {

color: white;

background-color: cyan;

display: inline-block;

padding: 10px 20px;

border: 2px solid #099983;

text-decoration: none;

text-align: center;

font: 14px Arial, sans-serif;

}

a:hover, a:active {

background-color: purple;

border-color: blue;

}

</style>

</head>

<body>

<p><a href="#">CSS Link Button</a></p>

</body>

</html>

# CSS Tables

**Adding Borders to Tables**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Example of Setting Table Borders</title>

<style>

table, th, td {

border: 1px solid black;

}

</style>

</head>

<body>

<table>

<tr>

<th>ID</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

<tr>

<td>1</td>

<td>John</td>

<td>Carter</td>

<td>johncarter@mail.com</td>

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

<td>peterparker@mail.com</td>

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Rambo</td>

<td>johnrambo@mail.com</td>

</tr>

<tr>

<td>4</td>

<td>Harry</td>

<td>Potter</td>

<td>harrypotter@mail.com</td>

</tr>

</table>

</body>

</html>

**Collapsing Table Borders**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Example of Merging Separate Table Borders</title>

<style>

table {

border-collapse: collapse;

}

table, th, td {

border: 1px solid black;

}

</style>

</head>

<body>

<table>

<tr>

<th>ID</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

<tr>

<td>1</td>

<td>John</td>

<td>Carter</td>

<td>johncarter@mail.com</td>

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

<td>peterparker@mail.com</td>

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Rambo</td>

<td>johnrambo@mail.com</td>

</tr>

<tr>

<td>4</td>

<td>Harry</td>

<td>Potter</td>

<td>harrypotter@mail.com</td>

</tr>

</table>

</body>

</html>

**Adjusting Space inside Tables**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Example of Setting Table Cell Padding</title>

<style>

table {

border-collapse: collapse;

}

table, th, td {

border: 1px solid black;

}

th, td {

padding: 15px;

}

</style>

</head>

<body>

<table>

<tr>

<th>ID</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

<tr>

<td>1</td>

<td>John</td>

<td>Carter</td>

<td>johncarter@mail.com</td>

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

<td>peterparker@mail.com</td>

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Rambo</td>

<td>johnrambo@mail.com</td>

</tr>

<tr>

<td>4</td>

<td>Harry</td>

<td>Potter</td>

<td>harrypotter@mail.com</td>

</tr>

</table>

</body>

</html>

**Setting Table Width and Height**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Example of Setting Table Width and Height</title>

<style>

table {

width: 100%;

border-collapse: collapse;

}

th, td {

padding: 8px;

border: 1px solid #dee2e6;

}

th {

height: 40px;

text-align: left;

}

</style>

</head>

<body>

<table>

<thead>

<tr>

<th>Row</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

<td>clarkkent@mail.com</td>

</tr>

<tr>

<td>2</td>

<td>John</td>

<td>Carter</td>

<td>johncarter@mail.com</td>

</tr>

<tr>

<td>3</td>

<td>Peter</td>

<td>Parker</td>

<td>peterparker@mail.com</td>

</tr>

</tbody>

</table>

</body>

</html>

**Controlling the Table Layout**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Example of CSS table-layout property</title>

<style>

table {

width: 250px;

border-collapse: separate;

}

table, tr, th, td{

border: 1px solid #000000;

}

.auto {

table-layout: auto;

}

.fixed {

table-layout: fixed;

}

td{

width: 50%;

}

</style>

</head>

<body>

<table class="auto">

<caption>Example 1. Auto</caption>

<tr>

<th>Name</th>

<td>John Carter</td>

</tr>

<tr>

<th>Email</th>

<td>johncarter@mail.com</td>

</tr>

</table>

<br>

<table class="fixed">

<caption>Example 2. Fixed</caption>

<tr>

<th>Name</th>

<td>Peter Parker</td>

</tr>

<tr>

<th>Email</th>

<td>peterparker@mail.com</td>

</tr>

</table>

<p><strong>Note:</strong> You can see the width of table cell does not change to accommodate the content in fixed table-layout.</p>

</body>

</html>

**Aligning the Text Inside Table Cells**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Example of Horizontal Aligning Text in Table Cells</title>

<style>

table {

width: 100%;

border-collapse: collapse;

}

th, td {

padding: 8px;

border: 1px solid #dee2e6;

}

th {

text-align: left;

}

</style>

</head>

<body>

<table>

<thead>

<tr>

<th>Row</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

<td>clarkkent@mail.com</td>

</tr>

<tr>

<td>2</td>

<td>John</td>

<td>Carter</td>

<td>johncarter@mail.com</td>

</tr>

<tr>

<td>3</td>

<td>Peter</td>

<td>Parker</td>

<td>peterparker@mail.com</td>

</tr>

</tbody>

</table>

</body>

</html>

### Vertical Alignment of Cell Contents

Similarly, you can vertically align the content inside the <th> and <td> elements to top, bottom, or middle using the CSS vertical-align property. The default vertical alignment is middle.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Example of Vertical Aligning Text in Table Cells</title>

<style>

table {

width: 100%;

border-collapse: collapse;

}

th, td {

padding: 8px;

border: 1px solid #dee2e6;

}

th {

height: 40px;

vertical-align: bottom;

}

</style>

</head>

<body>

<table>

<thead>

<tr>

<th>Row</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

<td>clarkkent@mail.com</td>

</tr>

<tr>

<td>2</td>

<td>John</td>

<td>Carter</td>

<td>johncarter@mail.com</td>

</tr>

<tr>

<td>3</td>

<td>Peter</td>

<td>Parker</td>

<td>peterparker@mail.com</td>

</tr>

</tbody>

</table>

</body>

</html>

**Controlling the Position of Table Caption**

You can set the vertical position of a table caption using the CSS caption-side property.

The caption can be placed either at the top or bottom of the table. The default position is top.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Example of CSS caption-side property</t

itle>

<style>

table, td, th {

border: 1px solid gray;

}

caption {

caption-side: bottom;

}

</style>

</head>

<body>

<table>

<caption>Table 1.0 - User Details</caption>

<thead>

<tr>

<th>No.</th>

<th>Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>John Carter</td>

<td>johncarter@mail.com</td>

</tr>

<tr>

<td>2</td>

<td>Peter Parker</td>

<td>peterparker@mail.com</td>

</tr>

<tr>

<td>3</td>

<td>John Rambo</td>

<td>johnrambo@mail.com</td>

</tr>

</tbody>

</table>

<p><strong>Note:</strong> Internet Explorer 8 supports the caption-side property only if a <code>!DOCTYPE</code> is specified.</p>

</body>

</html>

**Creating Zebra-striped Tables**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Example of Creating Zebra-striped Tables</title>

<style>

table {

width: 100%;

font-family: arial, sans-serif;

border-collapse: collapse;

}

th, td {

padding: 8px;

text-align: left;

border-top: 1px solid #dee2e6;

}

tbody tr:nth-child(odd) {

background-color: #f2f2f2;

}

</style>

</head>

<body>

<table>

<thead>

<tr>

<th>Row</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

<td>clarkkent@mail.com</td>

</tr>

<tr>

<td>2</td>

<td>John</td>

<td>Carter</td>

<td>johncarter@mail.com</td>

</tr>

<tr>

<td>3</td>

<td>Peter</td>

<td>Parker</td>

<td>peterparker@mail.com</td>

</tr>

</tbody>

</table>

</body>

</html>

Hoverable Table

<!DOCTYPE html>

<html>

<head>

<style>

table {

border-collapse: collapse;

width: 100%;

}

th, td {

padding: 8px;

text-align: left;

border-bottom: 1px solid #ddd;

}

tr:hover {background-color:#f5f5f5;}

</style>

</head>

<body>

<h2>Hoverable Table</h2>

<p>Move the mouse over the table rows to see the effect.</p>

<table>

<tr>

<th>First Name</th>

<th>Last Name</th>

<th>Points</th>

</tr>

<tr>

<td>Peter</td>

<td>Griffin</td>

<td>$100</td>

</tr>

<tr>

<td>Lois</td>

<td>Griffin</td>

<td>$150</td>

</tr>

<tr>

<td>Joe</td>

<td>Swanson</td>

<td>$300</td>

</tr>

<tr>

<td>Cleveland</td>

<td>Brown</td>

<td>$250</td>

</tr>

</table>

</body>

</html>

**Hence border [properties:**

**Border is a short hand property.**

**As** border: 3px solid red;

border: <border-width> || <border-style> || <color>

* border-width: Specifies the thickness of the border.
  + <length>: A numeric value measured in px, em, rem, vh and vw units.
  + thin: The equivalent of 1px
  + medium: The equivalent of 3px
  + thick: The equivalent of 5px
* border-style: Specifies the type of line drawn around the element, including:
  + solid: A solid, continuous line.
  + none (default): No line is drawn.
  + hidden: A line is drawn, but not visible. this can be handy for adding a little extra width to an element without displaying a border.
  + dashed: A line that consists of dashes.
  + dotted: A line that consists of dots.
  + double: Two lines are drawn around the element.
  + groove: Adds a bevel based on the color value in a way that makes the element appear pressed into the document.
  + ridge: Similar to groove, but reverses the color values in a way that makes the element appear raised.
  + inset: Adds a split tone to the line that makes the element appear slightly depressed.
  + outset: Similar to inset, but reverses the colors in a way that makes the element appear slightly raised.
* color: Specifies the color of the border and accepts <rgb()>, <rgba()>, <code><hsl()>, <hsla()>, <hex-color>, <named-color>, currentcolor and <deprecated-system-color>

**Example:**

<style> body {

display: flex;

align-items: flex-start;

flex-wrap: wrap;

justify-content: space-between;

padding: 2em;

}

.box {

background: #eaeaea;

display: flex;

align-items: center;

justify-content: center;

flex-basis: 17%;

height: 200px;

margin-bottom: 1em;

}

.box-1 {

border: none;

}

.box-2 {

border: 5px hidden red;

}

.box-3 {

border: 5px solid orange;

}

.box-4 {

border: 5px dashed orange;

}

.box-5 {

border: 5px dotted orange;

}

.box-6 {

border: 5px double orange;

}

.box-7 {

border: 5px groove orange;

}

.box-8 {

border: 5px ridge orange;

}

.box-9 {

border: 5px inset orange;

}

.box-10 {

border: 5px outset orange;

}

</style>

<body>

<div class="box box-1">

none

</div>

<div class="box box-2">

hidden

</div>

<div class="box box-3">

solid

</div>

<div class="box box-4">

dashed

</div>

<div class="box box-5">

dotted

</div>

<div class="box box-6">

double

</div>

<div class="box box-7">

groove

</div>

<div class="box box-8">

ridge

</div>

<div class="box box-9">

inset

</div>

<div class="box box-10">

outset

</div>

</body>

**Other properties:**

* [border-collapse:](https://css-tricks.com/almanac/properties/b/border-collapse/) Specifies the spacing between borders on the <table> element.
* [border-image:](https://css-tricks.com/almanac/properties/b/border-image/) Allows the use of an image to draw the border instead of a solid color.
* [border-radius:](https://css-tricks.com/almanac/properties/b/border-radius/) Provides control for rounded corners.

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

border-image-source: url(border.png);

[border-image-slice](https://www.w3schools.com/cssref/css3_pr_border-image-slice.asp): How to slice the border image

#### border-image-repeat

##### Values

* stretch: the initial value. The border image is stretched as needed to fill the area.
* repeat: the image tiles to fill the area, dividing tiles if necessary.
* round: the image tiles to fill the area, and is rescaled if necessary to avoid dividing tiles.
* space: the image tiles to fill the area. If the area can’t be filled with whole tiles, space is inserted around the tiles for an even fit. Note: this value is not implemented by any browser yet.