**Universal Selector**

**The universal selector, written as \* i.e. asterisk or star symbol, matches every single element on the page. The universal selector may be omitted if other conditions exist on the target element.**

**The style rules inside the \* selector will be applied to every element in a document.**

## <!DOCTYPE html>

## <html>

## <head>

## <style>

## \*{

## text-align:center;

## color: red;

## font-family: verdana;

## font-size: 110%;

## }

## </style>

## </head>

## <body>

## <h1>Hello world!</h1>

## <h2>Hello EveryBody</h2>

## <p>CSS Example1</p>

## <p>CSS Example2</p>

## <p>CSS Example3</p>

## </body>

## </html>

## 

## Example2

**The universal selector may be omitted if other conditions exist on the target element.**

## <!DOCTYPE html>

## <html>

## <head>

## <style>

## \*{

## text-align:center;

## color: red;

## font-family: verdana;

## font-size: 110%;

## }

## .myclass{

## text-align: left;

## color: blue;

## font-size: 150%;

## }

## </style>

## </head>

## <body>

## <h1>Hello world!</h1>

## <h2>Hello EveryBody</h2>

## <p>CSS Example1</p>

## <p class="myclass">CSS Example2</p>

## <p class="myclass">CSS Example3</p>

## </body>

## </html>

## 

[Combinator selectors](https://www.w3schools.com/css/css_combinators.asp)

**There are four different combinators in CSS:**

* **descendant selector (space)**
* **child selector (>)**
* **adjacent sibling selector (+)**
* **general sibling selector (~)**

## The Descendant Selectors

Suppose you want to apply a style rule to a particular element only when it lies inside a particular element.

The descendant selector matches all elements that are descendants of a specified element.

The following example selects all <p> elements inside <div> elements:

Example1

<!DOCTYPE html>

<html>

<head>

<style>

div p {

background-color: yellow;

border:5px dotted cyan;

text-align:center;

color: red;

font-family: verdana;

font-size: 110%;

}

</style>

</head>

<body>

<h2>Descendant Selector Example</h2>

<p>Hello EveryBody. This is an example of Descendant Selector</p>

<div>

<p>Java Programming </p>

<p>Full Stack Web Development</p>

<p> C++ Programming</p>

</div>

<p>Paragraph 1</p>

<p>Paragraph 2</p>

<h2>End of Program</h2>

</body>

</html>

# 

As given in the following example, style rule will apply to <em> element only when it lies inside <ul> tag.

# <em> tag

## Description

The HTML **<em> tag** marks text that has stress emphasis which traditionally means that the text is displayed in italics by the browser. This tag is also commonly referred to as the <em> element.

**Syntax**

In HTML, the syntax for the **<em> tag** is:

<body>

<p><em>Stress emphasized text goes here</em> but not here</p>

</body>

### Sample Output

Stress emphasized text goes here but not here

**Example2**

<!DOCTYPE html>

<html>

<head>

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

<style type="text/css">

h1 em {

color: green;

}

</style>

</head>

<body>

<h1>Example of CSS <em> Descendant Selectors</em></h1>

<ul>

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

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

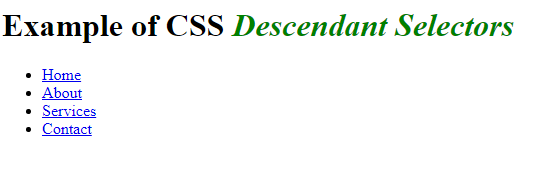
<li><a href="#">Services</a></li>

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

</ul>

</body>

</html>



**Example3**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>Example of CSS Descendant Selectors</title>**

**<style type="text/css">**

**h1 em {**

**color: green;**

**}**

**ul.menu {**

**padding: 0;**

**list-style: none;**

**}**

**ul.menu li{**

**display: inline;**

**}**

**ul.menu li a {**

**margin: 10px;**

**text-decoration: none;**

**}**

**</style>**

**</head>**

**<body>**

**<h1>This is a <em>heading</em></h1>**

**<ul class="menu">**

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

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

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

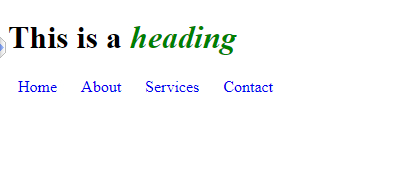
**<li><a href="#">Contact</a></li>**

**</ul>**

**</body>**

**</html>**

The style rules inside the selector ul.menu li a applied to only those [<a>](https://www.tutorialrepublic.com/html-tutorial/html-links.php) i.e. anchor elements that contained inside an [unordered list](https://www.tutorialrepublic.com/html-tutorial/html-lists.php) having the class .menu, and has no effect on other links inside the document. Similarly, the style rules inside the h1 em selector applied to only [<em>](https://www.tutorialrepublic.com/html-reference/html-em-tag.php) elements that contained inside heading [<h1>](https://www.tutorialrepublic.com/html-tutorial/html-headings.php).



h1 {  
  text-decoration: overline;  
}  
  
h2 {  
  text-decoration: line-through;  
}  
  
h3 {  
  text-decoration: underline;  
}  
  
h3 {  
  text-decoration: underline overline;  
}

## Child Selectors

A child selector can be 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 the greater than symbol (i.e. >).

**Example1**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<style>**

**body > p {**

**text-align:center;**

**color: red;**

**font-family: verdana;**

**font-size: 110%;**

**}**

**</style>**

**</head>**

**<body>**

**<h2>Descendant Selector Example</h2>**

**<p>Hello EveryBody. This is an example of Descendant Selector</p>**

**<div>**

**<p>Java Programming </p>**

**<p>Full Stack Web Development</p>**

**<p> C++ Programming</p>**

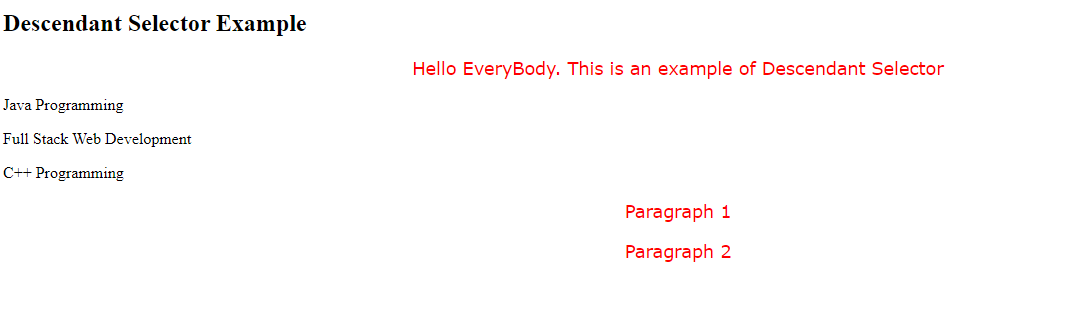
**</div>**

**<p>Paragraph 1</p>**

**<p>Paragraph 2</p>**

**</body>**

**</html>**



**Example2**

You can use these selectors for example, to select the first level of list elements inside a nested list that has more than one level.

**<!DOCTYPE html>**

**<html>**

**<head>**

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

**<style type="text/css">**

**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>**



Adjacent Sibling Selectors

The adjacent sibling selectors can be used to select sibling elements. This selector has the syntax like: E1 + E2, where E2 is the target of the selector.

The selector h1 + p in the example below will select the [<p>](https://www.tutorialrepublic.com/html-tutorial/html-paragraphs.php) elements only if both the [<h1>](https://www.tutorialrepublic.com/html-tutorial/html-headings.php) and <p>elements share the same parent in the document tree and <h1> is immediately precedes the <p>element. That means only those paragraphs that come immediately after each <h1> heading will have the associated style rules.

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>Example of CSS Adjacent Sibling Selectors</title>**

**<style type="text/css">**

**h1 + p {**

**color: blue;**

**font-size: 18px;**

**}**

**ul.san + p {**

**color: #f0f;**

**text-indent: 30px;**

**}**

**</style>**

**</head>**

**<body>**

**<h1>This is a heading</h1>**

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

**<h1>This is a heading</h1>**

**<p>This is another paragraph.</p>**

**<p>This is another paragraph But not effected.</p>**

**<ul class="san">**

**<li>Java </li>**

**<li>Web Design</li>**

**<li>Compiler</li>**

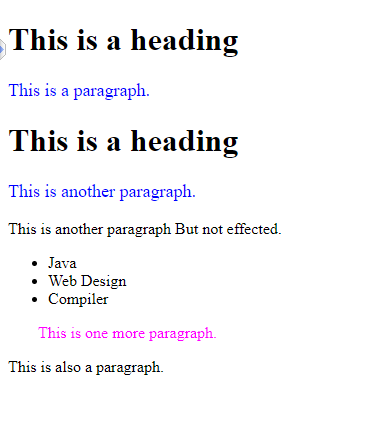
**</ul>**

**<p>This is one more paragraph.</p>**

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

**</body>**

**</html>**



**Example2**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>Example of CSS Adjacent Sibling Selectors</title>**

**<style type="text/css">**

**div + p {**

**text-align:center;**

**color: red;**

**font-family: verdana;**

**font-size: 110%;**

**}**

**</style>**

**</head>**

**<body>**

**<p>This is a paragraph 1</p>**

**<div>**

**<p>This is another paragraph But not effected.</p>**

**</div>**

**<p>This is paragraph 2</p>**

**<p>This is a paragraph 3</p>**

**</body>**

**</html>**



## General Sibling Selectors

**A general sibling selector is made up of two simple selectors separated by the tilde (∼) character. It can be written like: E1 ∼ E2, where E2 is the target of the selector.**

**The selector h1 ∼ p in the example below will select all the <p> elements that preceded by the <h1> element, where all the elements share the same parent in the document tree.**

The general sibling selector selects all elements that are next siblings of a specified element.

<!DOCTYPE html>

<html>

<head>

<title>Example of CSS General Sibling Selectors</title>

**<style type="text/css">**

**h1 ~ p {**

**color: blue;**

**font-size: 18px;**

**}**

**ul.san ~ p {**

**color: #f0f;**

**text-indent: 30px;**

**}**

**</style>**

</head>

<body>

<h1>This is a heading</h1>

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

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

<ul class="san">

<li>Java</li>

<li>WebDesign</li>

<li>Compiler</li>

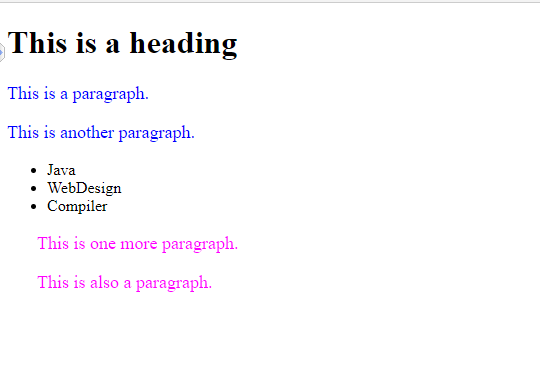
</ul>

<p>This is one more paragraph.</p>

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

</body>

</html>



## The Attribute Selectors

You can also apply styles to HTML elements with particular attributes.

It is possible to style HTML elements that have specific attributes or attribute values.

## CSS [attribute] Selector

The [attribute] selector is used to select elements with a specified attribute.

Example1

<!DOCTYPE html>

<html>

<head>

<style>

a[title]{

color:red;

}

a[target] {

text-decoration:underline;

}

</style>

</head>

<body>

<a title="Course Detail" href="/css\_links" target="\_blank">Click for Course Detail

</a>

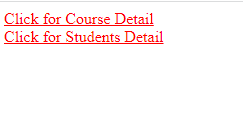
<br>

<a title="Student Detail" href="/css\_links" target="\_blank">Click for Students Detail

</a>

</body>

</html>



## CSS [attribute="value"] Selector

The [attribute="value"] selector is used to select elements with a specified attribute and value.

The style rule below will match all the input elements having a type attribute with a value of *text*

<!DOCTYPE html>

<html>

<head>

<style>

a[target="\_blank"] {

background-color: cyan;

}

input[type="text"] {

color: red;

background:yellow;

}

</style>

</head>

<body>

<a href="http://www.university.com" target="\_blank">university.com</a>

<br>

<a href="http://www.wikipedia.org" target="\_top">wikipedia.org</a>

<form>

<label>Enter your name

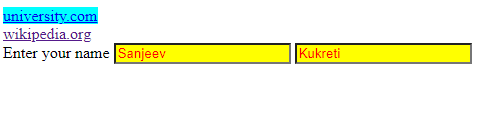
<input type="text" name="fname">

<input type="text" name="lname"></label>

</form>

</body>

</html>



## CSS [attribute~="value"] Selector

The [attribute~="value"] selector is used to select elements with an attribute value containing a specified word.

You can use the ~= operator to make an attribute selector matches any element whose attribute value is a list of *space-separated* values (like class="alert warning") , one of which is exactly equal to the specified value:

<!DOCTYPE html>

<html>

<head>

<style>

[class~="cp1"]

{

color: #fff;

background: red;

}

</style>

</head>

<body>

<p class="cp1">Paragraph 1</p>

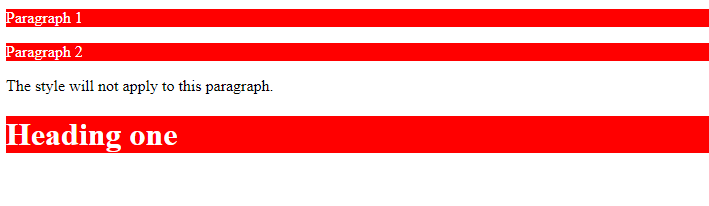
<p class="cp1 cp2">Paragraph 2</p>

<p class="cp3 mycp1">The style will not apply to this paragraph.</p>

<h1 class="mycp1 cp1">Heading one</h1>

</body>

</html>



## CSS [attribute|="value"] Selector

The [attribute|="value"] selector is used to select elements with the specified attribute, whose value can be exactly the specified value, or the specified value followed by a hyphen (-).

**Note:** The value has to be a whole word, either alone, like class="top", or followed by a hyphen( - ), like class="top-text".

<!DOCTYPE html>

<html>

<head>

<style>

[class|="top"] {

background: red;

}

</style>

</head>

<body>

<h1 class="top-header">Welcome</h1>

<p class="top-text">Hello world!</p>

<p class="topcontent">Are you learning CSS?</p>

<p class="topvalue">Are you learning CSS?</p>

<h1 class="top-header">Welcome</h1>

<h1 class="top header">Hello</h1>

<h1 class="top">Everybody</h1>

</body>

</html>



## CSS [attribute^="value"] Selector

The [attribute^="value"] selector is used to select elements with the specified attribute, whose value starts with the specified value.

The following example selects all elements with a class attribute value that starts with "top":

**Note:** The value does not have to be a whole word!

<!DOCTYPE html>

<html>

<head>

<style>

[class^="top"] {

background: red;

}

</style>

</head>

<body>

<h1 class="top-header">Welcome</h1>

<p class="top text">Hello world!</p>

<p class="topcontent">Are you learning CSS?</p>

<p class="value top">CSS Example</p>

<h1 class="value-top header">Welcome</h1>

<h1 class="top">Everybody</h1>

</body>

</html>



## CSS [attribute$="value"] Selector

The [attribute$="value"] selector is used to select elements whose attribute value ends with a specified value.

The following example selects all elements with a class attribute value that ends with "top"

<!DOCTYPE html>

<html>

<head>

<style>

[class$="top"] {

background: yellow;

}

</style>

</head>

<body>

<h1 class="headertop">Welcome</h1>

<p class="top">Hello world!</p>

<p class="topcontent">Are you learning CSS?</p>

<p class="value top">CSS Example</p>

<h1 class="value-top header">Welcome</h1>

<h1 class="valuetopcontent">Everybody</h1>

</body>

</html>



## CSS [attribute\*="value"] Selector

You can use the \*= operator to make an attribute selector matches all elements whose attribute value contains a specified value.

<!DOCTYPE html>

<html>

<head>

<style>

[class\*="top"] {

background: yellow;

}

</style>

</head>

<body>

<h1 class="headertop">Welcome</h1>

<p class="top">Hello world!</p>

<p class="topcontent">Are you learning CSS?</p>

<p class="value top">CSS Example</p>

<h1 class="value-bottom">Welcome</h1>

<h1 class="valuetopcontent">Everybody</h1>

</body>

</html>

****

## Styling Forms with Attribute Selectors

The attribute selectors are particularly useful for styling forms without [class](https://www.tutorialrepublic.com/html-reference/html-common-attributes.php) or [id](https://www.tutorialrepublic.com/html-reference/html-common-attributes.php):

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Example of CSS Attribute Selector</title>

<style>

input[type="text"], input[type="password"] {

width: 150px;

display: block;

margin-bottom: 10px;

background: yellow;

}

input[type="submit"] {

padding: 2px 10px;

border: 1px solid #804040;

background: #ff8040;

}

</style>

</head>

<body>

<form>

<label>Username: <input type="text"></label>

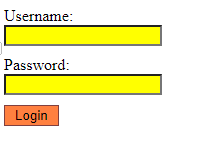
<label>Password: <input type="password"></label>

<input type="submit" value="Login">

</form>

</body>

</html>



# CSS Pseudo-classes

**What is Pseudo-class**

The CSS pseudo-classes allow you to style the dynamic states of an element such as hover, active and focus state, as well as elements that are existing in the document tree but can't be targeted via the use of other selectors without adding any IDs or classes to them, for example, targeting the first or last child elements.

A pseudo-class starts with a colon (:). Its syntax can be given with:

selector:pseudo-class { property: value; }

What are Pseudo-classes?

A pseudo-class is used to define a special state of an element.

For example, it can be used to:

* Style an element when a user mouses over it
* Style visited and unvisited links differently
* Style an element when it gets focus

## Syntax

The syntax of pseudo-classes:

selector:pseudo-class

 {  
  property: value;  
}

## All CSS Pseudo Classes

|  |  |  |
| --- | --- | --- |
| **Selector** | **Example** | **Example description** |
| [:active](https://www.w3schools.com/cssref/sel_active.asp) | a:active | Selects the active link |
| [:checked](https://www.w3schools.com/cssref/sel_checked.asp) | input:checked | Selects every checked <input> element |
| [:disabled](https://www.w3schools.com/cssref/sel_disabled.asp) | input:disabled | Selects every disabled <input> element |
| [:empty](https://www.w3schools.com/cssref/sel_empty.asp) | p:empty | Selects every <p> element that has no children |
| [:enabled](https://www.w3schools.com/cssref/sel_enabled.asp) | input:enabled | Selects every enabled <input> element |
| [:first-child](https://www.w3schools.com/cssref/sel_firstchild.asp) | p:first-child | Selects every <p> elements that is the first child of its parent |
| [:first-of-type](https://www.w3schools.com/cssref/sel_first-of-type.asp) | p:first-of-type | Selects every <p> element that is the first <p> element of its parent |
| [:focus](https://www.w3schools.com/cssref/sel_focus.asp) | input:focus | Selects the <input> element that has focus |
| [:hover](https://www.w3schools.com/cssref/sel_hover.asp) | a:hover | Selects links on mouse over |
| [:in-range](https://www.w3schools.com/cssref/sel_in-range.asp) | input:in-range | Selects <input> elements with a value within a specified range |
| [:invalid](https://www.w3schools.com/cssref/sel_invalid.asp) | input:invalid | Selects all <input> elements with an invalid value |
| [:lang(*language*)](https://www.w3schools.com/cssref/sel_lang.asp) | p:lang(it) | Selects every <p> element with a lang attribute value starting with "it" |
| [:last-child](https://www.w3schools.com/cssref/sel_last-child.asp) | p:last-child | Selects every <p> elements that is the last child of its parent |
| [:last-of-type](https://www.w3schools.com/cssref/sel_last-of-type.asp) | p:last-of-type | Selects every <p> element that is the last <p> element of its parent |
| [:link](https://www.w3schools.com/cssref/sel_link.asp) | a:link | Selects all unvisited links |
| [:not(selector)](https://www.w3schools.com/cssref/sel_not.asp) | :not(p) | Selects every element that is not a <p> element |
| [:nth-child(n)](https://www.w3schools.com/cssref/sel_nth-child.asp) | p:nth-child(2) | Selects every <p> element that is the second child of its parent |
| [:nth-last-child(n)](https://www.w3schools.com/cssref/sel_nth-last-child.asp) | p:nth-last-child(2) | Selects every <p> element that is the second child of its parent, counting from the last child |
| [:nth-last-of-type(n)](https://www.w3schools.com/cssref/sel_nth-last-of-type.asp) | p:nth-last-of-type(2) | Selects every <p> element that is the second <p> element of its parent, counting from the last child |
| [:nth-of-type(n)](https://www.w3schools.com/cssref/sel_nth-of-type.asp) | p:nth-of-type(2) | Selects every <p> element that is the second <p> element of its parent |
| [:only-of-type](https://www.w3schools.com/cssref/sel_only-of-type.asp) | p:only-of-type | Selects every <p> element that is the only <p> element of its parent |
| [:only-child](https://www.w3schools.com/cssref/sel_only-child.asp) | p:only-child | Selects every <p> element that is the only child of its parent |
| [:optional](https://www.w3schools.com/cssref/sel_optional.asp) | input:optional | Selects <input> elements with no "required" attribute |
| [:out-of-range](https://www.w3schools.com/cssref/sel_out-of-range.asp) | input:out-of-range | Selects <input> elements with a value outside a specified range |
| [:read-only](https://www.w3schools.com/cssref/sel_read-only.asp) | input:read-only | Selects <input> elements with a "readonly" attribute specified |
| [:read-write](https://www.w3schools.com/cssref/sel_read-write.asp) | input:read-write | Selects <input> elements with no "readonly" attribute |
| [:required](https://www.w3schools.com/cssref/sel_required.asp) | input:required | Selects <input> elements with a "required" attribute specified |
| [:root](https://www.w3schools.com/cssref/sel_root.asp) | root | Selects the document's root element |
| [:target](https://www.w3schools.com/cssref/sel_target.asp) | #news:target | Selects the current active #news element (clicked on a URL containing that anchor name) |
| [:valid](https://www.w3schools.com/cssref/sel_valid.asp) | input:valid | Selects all <input> elements with a valid value |
| [:visited](https://www.w3schools.com/cssref/sel_visited.asp) | a:visited | Selects all visited links |

# CSS Links

**Styling Links with CSS**

A link has four different states — link, visited, active and hover. These four states of a link or hyperlink can be styled differently through CSS properties using the [pseudo-classes](https://www.tutorialrepublic.com/css-tutorial/css-pseudo-classes.php) of anchor element, depending on what state they are in.

Here're the CSS pseudo-classes associated with HTML [<a>](https://www.tutorialrepublic.com/css-tutorial/css-links.php) tag that you can use to define the different styles for the different states of a hyperlink.

**a:link** — Set styles for a normal or unvisited links that has no   
 mouse pointer over it.

**a:visited** — Set styles for a link the user has visited but has no   
 mouse pointer on it.

**a:hover** — Set styles for a link when the user place the mouse   
 pointer over it.

**a:active** — Set styles for a link that is in the process of being   
 clicked.

**Example**

<!DOCTYPE html>

<html>

<head>

<style type="text/css">

a:link {

color: red;

text-decoration: none;

}

a:visited {

color: green;

}

a:hover {

color: blue;

text-decoration: underline;

}

a:active {

color: yellow;

}

</style>

</head>

<body>

<p><a href="#">Information About CSS</a></p>

<p><a href="#">Information About Selectors</a></p>

</body>

</html>

Text Decoration

The text-decoration property is mostly used to remove underlines from links:

a:link {  
 text-decoration: none;  
}  
  
a:visited {  
 text-decoration: none;  
}  
  
a:hover {  
  text-decoration: underline;  
}  
  
a:active {  
  text-decoration: underline;  
}

Background Color

The background-color property can be used to specify a background color for links:

a:link {  
  background-color: yellow;  
}  
  
a:visited {  
 background-color: cyan;  
}  
  
a:hover {  
  background-color: lightgreen;  
}  
  
a:active {  
  background-color: hotpink;  
}

## Pseudo-classes and HTML Classes

Pseudo-classes can be combined with HTML classes:

When you hover over the link in the example, it will change color:

<!DOCTYPE html>

<html>

<head>

<style>

a.cent:hover {

color: #ff0000;

font-size: 22px;

}

</style>

</head>

<body>

<p><a class="cent" href="#">CSS Syntax</a></p>

<p><a href="#">CSS Tutorial</a></p>

<p><a class="cent" href="#">CSS Selectors</a></p>

</body>

</html>



**Example3**

<!DOCTYPE html>

<html>

<head>

<style>

.bt1

{

background-color: aqua;

color: black;

padding:10px 20px;

}

.bt1:hover

{

background-color:yellow;

}

</style>

</head>

<body>

<button class="bt1">Button</button>

<p> When you Mouse over button, background color will change to yellow color</p>

</body>

</html>

## Hover on <div>

An example of using the :hover pseudo-class on a <div> element:

**Example**

<!DOCTYPE html>

<html>

<head>

<style>

div {

background-color: green;

color: white;

padding: 25px;

text-align: center;

}

div:hover {

background-color: blue;

}

</style>

</head>

<body>

<div>Mouse over the div element to change its background color:</div>

</body>

</html>

## Simple Tooltip Hover

Hover over a <div> element to show a <p> element (like a tooltip):

**Example**

<!DOCTYPE html>

<html>

<head>

<style>

p {

display: none;

background-color: yellow;

padding: 20px;

}

div:hover p {

display: block;

}

</style>

</head>

<body>

<div>Hover over this div element to show the p element

<p>Hello EveryBody</p>

</div>

</body>

</html>

# CSS :focus Selector

The :focus selector is used to select the element that has focus.

**Tip:** The :focus selector is allowed on elements that accept keyboard events or other user inputs.

## CSS Syntax

:focus {  
  css declarations;  
}

Example

<!DOCTYPE html>

<html>

<head>

<style>

textarea:focus {

color: aqua;

width:300px;

height:auto;

background-color:black;

padding:20px 30px;

}

</style>

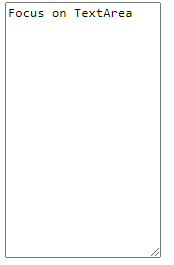
</head>

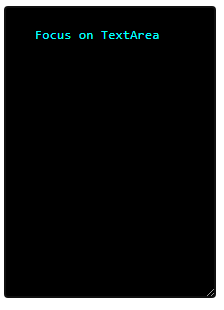
<body>

<textarea style="width:150px;height:250px">Focus on TextArea </textarea>

</body>

</html>





Link Buttons

This example demonstrates a more advanced example where we combine several CSS properties to display links as boxes/buttons:

a:link, a:visited {  
  background-color: blue;  
  color: white;  
  padding: 14px 25px;  
  text-align: center;   
  text-decoration: none;  
  display: inline-block;  
}  
  
a:hover, a:active {  
  background-color: red;  
}

<style>

a:link, a:visited {

background-color: white;

color: black;

border: 2px solid green;

padding: 10px 20px;

text-align: center;

text-decoration: none;

display: inline-block;

}

a:hover, a:active {

background-color: green;

color: white;

}

</style>

<!DOCTYPE html>

<html>

<head>

<style>

a.one:link {color:#ff0000;}

a.one:visited {color:#0000ff;}

a.one:hover {color:#ffcc00;}

a.two:link {color:#ff0000;}

a.two:visited {color:#0000ff;}

a.two:hover {font-size:150%;}

a.three:link {color:#ff0000;}

a.three:visited {color:#0000ff;}

a.three:hover {background:#66ff66;}

a.four:link {color:#ff0000;}

a.four:visited {color:#0000ff;}

a.four:hover {font-family:monospace;}

a.five:link {color:#ff0000;text-decoration:none;}

a.five:visited {color:#0000ff;text-decoration:none;}

a.five:hover {text-decoration:underline;}

</style>

</head>

<body>

<p>Mouse over the links and watch them change layout:</p>

<p><b><a class="one" href="default.asp" target="\_blank">This link changes color</a></b></p>

<p><b><a class="two" href="default.asp" target="\_blank">This link changes font-size</a></b></p>

<p><b><a class="three" href="default.asp" target="\_blank">This link changes background-color</a></b></p>

<p><b><a class="four" href="default.asp" target="\_blank">This link changes font-family</a></b></p>

<p><b><a class="five" href="default.asp" target="\_blank">This link changes text-decoration</a></b></p>

</body>

</html>

# CSS Tables

**Styling Tables with CSS**

## Adding Borders to Tables

The CSS [border](https://www.tutorialrepublic.com/css-reference/css-border-property.php) property is the best way to define the borders for the tables.

table, th, td {

border: 2px solid black;

}

**Collapsing Table Borders**

 collapse the separate table borders into one by using the CSS border-collapse property.

table {

border-collapse: collapse;

}

table, th, td {

border: 1px solid black;

}

<style type="text/css">

table {

border-collapse: collapse;

}

table, th, td {

border: 3px solid black;

}

</style>

Striped Tables

For zebra-striped tables, use the nth-child() selector and add a background-color to all even (or odd) table rows:

<!DOCTYPE html>

<html>

<head>

**<style>**

**table {**

**border-collapse: collapse;**

**width: 100%;**

**}**

**th, td {**

**text-align: left;**

**padding: 8px;**

**}**

**tr:nth-child(even) {background-color: red;}**

**</style>**

</head>

<body>

<h2>Striped Table</h2>

<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>

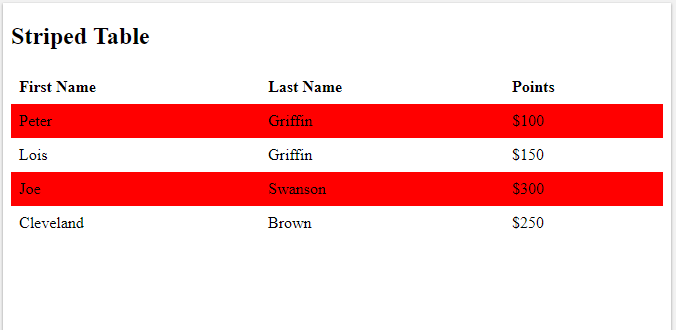
<td>$250</td>

</tr>

</table>

</body>

</html>



<!DOCTYPE html>

<html>

<head>

**<style>**

**table {**

**border-collapse: collapse;**

**width: 100%;**

**}**

**th, td {**

**text-align: left;**

**padding: 8px;**

**}**

**tr:nth-child(even){background-color: red}**

**tr:nth-child(odd){background-color: yellow}**

**th {**

**background-color: green;**

**color: white;**

**}**

**</style>**

</head>

<body>

<h2>Colored Table Header</h2>

<table>

<tr>

<th>Firstname</th>

<th>Lastname</th>

<th>Savings</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>

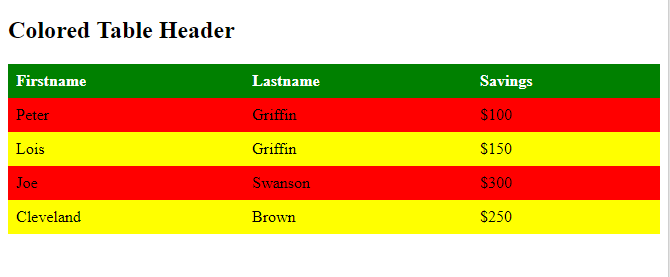


Table Width and Height

table {  
  width: 100%;  
}  
  
th {  
  height: 50px;  
}

Horizontal Alignment

th {  
  text-align: left;  
}

Vertical Alignment

td {  
  height: 50px;  
  vertical-align: bottom;  
}

Table Padding

To control the space between the border and the content in a table, use the padding property on <td> and <th> elements:

th, td {  
  padding: 15px;  
  text-align: left;  
}

Hoverable Table

Use the :hover selector on <tr> to highlight table rows on mouse over:

tr:hover {background-color: #f5f5f5;}

## CSS Pseudo Elements

What are Pseudo-Elements?

A CSS pseudo-element is used to style specified parts of an element.

For example, it can be used to:

* Style the first letter, or line, of an element
* Insert content before, or after, the content of an element

## Syntax of pseudo-element

## selector::pseudo-element{   property1:value1;   …   …   property-n:value-n;}

|  |  |  |
| --- | --- | --- |
| **Selector** | **Example** | **Example description** |
| [::after](https://www.w3schools.com/cssref/sel_after.asp) | p::after | Insert content after every <p> element |
| [::before](https://www.w3schools.com/cssref/sel_before.asp) | p::before | Insert content before every <p> element |
| [::first-letter](https://www.w3schools.com/cssref/sel_firstletter.asp) | p::first-letter | Selects the first letter of every <p> element |
| [::first-line](https://www.w3schools.com/cssref/sel_firstline.asp) | p::first-line | Selects the first line of every <p> element |
| [::marker](https://www.w3schools.com/cssref/sel_marker.asp) | ::marker | Selects the markers of list items |
| [::selection](https://www.w3schools.com/cssref/sel_selection.asp) | p::selection | Selects the portion of an element that is selected by a user |

# CSS ::first-letter Selector

The ::first-letter selector is used to add a style to the first letter of the specified selector.

## CSS Syntax

::first-letter {  
  css declarations;  
}

**Note:** The following properties can be used with ::first-letter:

* font properties
* color properties
* background properties
* margin properties
* padding properties
* border properties
* text-decoration
* vertical-align (only if float is 'none')
* text-transform
* line-height
* float
* clear

<!DOCTYPE html>

<html>

<head>

<style>

p::first-letter {

font-family:Algerian;

font-size:3em;

color:DodgerBlue;

}

</style>

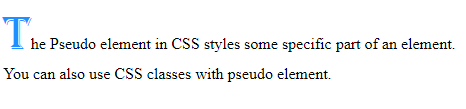
</head>

<body>

<p>The Pseudo element in CSS styles some specific part of an element. <br>You can also use CSS classes with pseudo element.</p>

</body>

</html>



## The ::first-line Pseudo-element

The ::first-line pseudo-element is used to add a special style to the first line of a text.

The following example formats the first line of the text in all <p> elements:

**Note:** The ::first-line pseudo-element can only be applied to block-level elements.

The following properties apply to the ::first-line pseudo-element:

* font properties
* color properties
* background properties
* word-spacing
* letter-spacing
* text-decoration
* vertical-align
* text-transform
* line-height
* clear

<!DOCTYPE html>

<html>

<head>

<style>

p::first-line {

font-family:Algerian;

font-size:1em;

color:DodgerBlue;

}

</style>

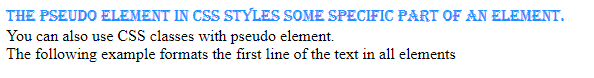
</head>

<body>

<p>The Pseudo element in CSS styles some specific part of an element. <br>You can also use CSS classes with pseudo element.<br>The following example formats the first line of the text in all elements</p>

</body>

</html>



## Multiple Pseudo-elements

Several pseudo-elements can also be combined.

**Example**

<style>

p::first-letter {

color: #ff0000;

font-size: xx-large;

}

p::first-line {

color: #0000ff;

font-variant: small-caps;

}

</style>

## CSS - The ::before Pseudo-element

The ::before pseudo-element can be used to insert some content before the content of an element.

Example

<!DOCTYPE html>

<html>

<head>

<style>

li::before {

content: " @# ";

}

</style>

</head>

<body>

<h4>Example for CSS ::before property</h4>

<ul type="none">

<li>Java</li>

<li>HTML</li>

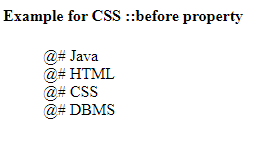
<li>CSS</li>

<li>DBMS</li>

</ul>

</body>

</html>



## CSS - The ::after Pseudo-element

The ::after pseudo-element can be used to insert some content after the content of an element.

<!DOCTYPE html>

<html>

<head>

<style>

li::after {

content: ">>";

}

li::before {

content: "<<";

}

</style>

</head>

<body>

<h4>Example for CSS Before and After Property</h4>

<ul type="none">

<li>Java</li>

<li>HTML</li>

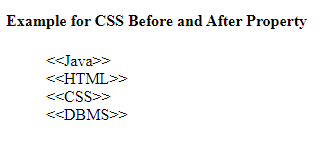
<li>CSS</li>

<li>DBMS</li>

</ul>

</body>

</html>



## CSS - The ::marker Pseudo-element

The ::marker pseudo-element selects the markers of list items.

<!DOCTYPE html>

<html>

<head>

<style>

::marker {

color: red;

font-size: 23px;

}

</style>

</head>

<body>

<ul>

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ul>

<ol>

<li>First</li>

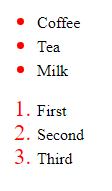
<li>Second</li>

<li>Third</li>

</ol>

</body>

</html>



## CSS - The ::selection Pseudo-element

The ::selection pseudo-element matches the portion of an element that is selected by a user.

The following CSS properties can be applied to ::selection: color, background, cursor, and outline.

The following example makes the selected text red on a yellow background:

<!DOCTYPE html>

<html>

<head>

<style>

::selection {

color: red;

background: yellow;

}

</style>

</head>

<body>

<h1>Select some text on this page:</h1>

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

<div>This is some text in a div element.</div>

</body>

</html>

