

CSS2 Properties

Contents

- Background
 - Border
 - Outline
 - Color
 - Text
 - Font
 - Margin & Padding
 - List
 - Box Size
 - Visual formatting
 - Table
-
-
-

Background

background-color

The background-color property sets the background color of an element.

Values

- `<color>`
The color can be specified as a hexadecimal RGB value, a regular RGB value, or by using one of the pre-defined color keywords.
- `transparent`
If specified, the background color is transparent. This is default.
- `inherit`
Takes the same specified value as the property for the element's parent

Example

[style.css]

```
body{
  background-color: #666;
}
h1{
  background-color: #ddf;
}
p{
  background-color: #ccc;
}
```

[index.html]

```
<body>
  <h1>background example</h1>
  <p>This is a paragraph</p>
</body>
```

background-image

The background-image property sets the background image of an element

Example

[style.css]

```
body {  
    background-image: url("images/bg.png") ;  
}
```

[index.html]

```
<body>  
    <p>This is a paragraph</p>  
</body>
```

background-repeat

The background-repeat property specifies whether the image is repeated (tiled), and how

Values

- `repeat`
The image is repeated both horizontally and vertically.
- `repeat-x`
The image is repeated horizontally only.
- `repeat-y`
The image is repeated vertically only.

- `no-repeat`
The image is not repeated: only one copy of the image is drawn.
- `inherit`
Takes the same specified value as the property for the element's parent

Example

[style.css]

```
body {  
    background-image: url("images/bg.png");  
    background-repeat: repeat-x;  
}
```

[index.html]

```
<body>  
    <p>This is a paragraph</p>  
</body>
```

background-attachment

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

Values

- `scroll`
If specified, the background image scrolls with the element.

- `fixed`
If specified, the background image does not scrolls with the element.
- `inherit`
Takes the same specified value as the property for the element's parent.

Example A

[style.css]

```
body {  
  background-image: url("images/bg_big.png");  
  background-attachment: scroll;  
  background-repeat: no-repeat;  
}
```

[index.html]

```
<body>  
  <p>This is a paragraph</p>  
</body>
```

background-position

The background-position property specifies its initial position.

Values

- `<percentage>`
- `<length>`
- `top`
Equivalent to 0% for the vertical position.

- `right`
Equivalent to 100% for the horizontal position.
- `bottom`
Equivalent to 100% for the vertical position.
- `left`
Equivalent to 0% for the horizontal position.
- `center`
Equivalent to '50%' for the horizontal position if it is not otherwise given, or '50%' for the vertical position if it is.
- `inherit`
Takes the same specified value as the property for the element's parent

Example A

[style.css]

```
body {  
  background-image: url("images/bg.png");  
  background-position: bottom right;  
  background-repeat: no-repeat;  
}
```

[index.html]

```
<body>  
  <p>This is a paragraph</p>  
</body>
```

background

The background property is a shorthand property for setting the individual background properties (i.e., 'background-color', 'background-image', 'background-repeat', 'background-attachment' and 'background-position') at the same place in the style sheet.

Example

[style.css]

```
body {  
    background: url("bg.png") right bottom no-repeat  
fixed;  
}
```

[index.html]

```
<body>
  <p>This is a paragraph</p>
</body>
```

border

The border property is a shorthand property for setting the same width, color, and style for all four borders of a box

Example

[style.css]

```
h1 {
  border: 1px solid red;
}
```

[index.html]

```
<body>
  <h1>CSS border property example</h1>
</body>
```


- border-width
- border-style
- border-color

border-width

- border-top-width
- border-right-width
- border-bottom-width
- border-left-width
- border-color
 - border-top-color
 - border-right-color
 - border-bottom-color
 - border-left-color
- border-style
 - border-top-style
 - border-right-style
 - border-bottom-style
 - border-left-style
- border
 - border-top
 - border-right
 - border-bottom
 - border-left

Outline

outline-width

The outline-width property sets the width of outline of a box.

Values

- `thin`
A thin border.
- `medium`
A medium border.
- `thick`
A thick border.
- `<length>`
- `inherit`
Takes the same specified value as the property for the element's parent.

Example

[style.css]

```
h1 {  
  outline-width: 1px;  
  outline-style: solid;  
}
```

[index.html]

```
<body>  
  <h1>CSS outline property example</h1>  
</body>
```

outline-style

The outline-style property specifies the line style of the outline of a box. This property accepts the same values as 'border-style', except that 'hidden' is not a legal outline style.

Values

- `none`
No border.
- `dotted`
The border is a series of dots.
- `dashed`
The border is a series of short line segments.
- `solid`
The border is a single line segment.
- `double`
The border is two solid lines. The sum of the two lines and the space between them equals the value of 'border-width'.
- `groove`
The border looks as though it were carved into the canvas.
- `ridge`
The opposite of 'groove': the border looks as though it were coming out of the canvas.

- `inset`
The border makes the box look as though it were embedded in the canvas.
- `outset`
The opposite of 'inset': the border makes the box look as though it were coming out of the canvas.
- `inherit`
Takes the same specified value as the property for the element's parent.

Example A

[style.css]

```
h1 {  
  outline-width: 1px;  
  outline-style: solid;  
}
```

[index.html]

```
<body>  
  <h1>CSS outline property example</h1>  
</body>
```

outline-color

The outline-color property sets the color of the outline of a box.

Example

[style.css]

```
h1 {  
  outline-width: 1px;  
  outline-style: solid;  
  outline-color: red;
```

```
}
```

[index.html]

```
<body>
  <h1>CSS outline property example</h1>
</body>
```

Outline

The outline property is a shorthand property, and sets all three of 'outline-style', 'outline-width', and 'outline-color'.

Example A

[style.css]

```
h1 {
  outline: 1px solid red;
}
```

[index.html]

```
<body>
  <h1>CSS outline property example</h1>
</body>
```

Color

The color property describes the foreground color of an element's text content.

Values

- `<color value>`

The color can be specified as

- a [hexadecimal RGB value](#): #faf or #ffaaff

- a [RGB value](#): `rgb(255, 160, 255)` or `rgb(100%, 62.5%, 100%)`
Each value is from 0 to 255, or from 0% to 100%.
- a [RGBA value](#): `rgba(255, 160, 255, 1)` or `rgba(100%, 62.5%, 100%, 1)`
This variant includes an “alpha” component to allow specification of the opacity of a color. Values are in the range 0.0 (fully transparent) to 1.0 (fully opaque).
- a [HSL value](#): `hsl(0, 100%, 50%)`
A triple (hue, saturation, lightness). hue is an angle in degrees. saturation and lightness are percentages (0-100%).
- a [HSLA value](#): `hsla(0, 100%, 50%, 1)`
This variant includes an “alpha” component to allow specification of the opacity of a color. Values are in the range 0.0 (fully transparent) to 1.0 (fully opaque)

Example

[style.css]

```
body {  
  color: #fff;  
  background-color: #666;  
}
```

[index.html]

```
<body>  
  <h1>CSS color property example</h1>  
</body>
```

opacity

The opacity property specifies the opacity of an element(transparency level).

Example

[style.css]

```
body {
  color: #fff;
  background-color: #666;
}
h1{
  background-color: red;
}
#op{
  opacity: 1.0;
}
#tr{
  opacity: 0.4;
}
```

[index.html]

```
<body>
  <h1 id="op">CSS color property example(fully
opaque)</h1>
  <h1 id="tr">CSS color property
example(transparent)</h1>
</body>
```

Text

text-align

The text-align property describes how inline-level content of a block container is aligned.

Values	left right center justify inherit
---------------	---

Example

[style.css]

```
p {  
  border: 1px solid black;  
  text-align: center;  
}
```

[index.html]

```
<body>  
  <p>This is a paragraph</p>  
</body>
```


text-decoration

The text-decoration property describes decorations that are added to the text of an element using the element's color.

Values	none [underline overline line-through blink] inherit
---------------	---

Example

[style.css]

```
p#no{ text-decoration: none; }
p#un{ text-decoration: underline; }
p#ov{ text-decoration: overline; }
p#th{ text-decoration: line-through; }
p#bl{ text-decoration: blink; }
```

[index.html]

```
<body>
  <p id="no">This is a paragraph (none)</p>
  <p id="un">This is a paragraph (underline)</p>
  <p id="ov">This is a paragraph (overline)</p>
  <p id="th">This is a paragraph (line-through)</p>
  <p id="bl">This is a paragraph (blink)</p>
</body>
```

text-indent

The text-indent property specifies the indentation of the first line of text in a block container.

Values	<length> <percentage> inherit
---------------	-----------------------------------

Example

[style.css]

```
p {  
  width: 300px;  
  border: 1px solid black;  
  text-indent: 1em;  
}
```

[index.html]

```
<body>  
  <p>CSS is a Style Sheet Language for formatting the  
document content  
  (written in HTML or other markup language). You  
might be interested  
  by the CSS Working Group wiki too.</p>  
</body>
```

text-transform

The text-transform property controls capitalization effects of an element's text.

Values	capitalize uppercase lowercase none inherit
---------------	---

Example A

[style.css]

```
p#no{ text-transform: none; }  
p#cap{ text-transform: capitalize; }  
p#up{ text-transform: uppercase; }  
p#low{ text-transform: lowercase; }
```

[index.html]

```
<body>  
  <p id="no">This is a paragraph(none)</p>  
  <p id="cap">This is a paragraph(capitalize)</p>  
  <p id="up">This is a paragraph(uppercase)</p>  
  <p id="low">This is a paragraph(lowercase)</p>  
</body>
```

line-height

On a block container element whose content is composed of inline-level elements, 'line-height' specifies the minimal height of line boxes within the element. On a non-replaced inline element, 'line-height' specifies the height that is used in the calculation of the line box height.

Values	normal <number> <length> <percentage> inherit
---------------	---

Example A

[style.css]

```
p {
  width: 300px;
  border: 1px solid black;
}
#two {
  line-height: 200%;
}
```

[index.html]

```
<body>
  <h2>line-height: normal</h2>
  <p>CSS is a Style Sheet Language for formatting the
    document content
      (written in HTML or other markup language). You
might be interested
      by the CSS Working Group wiki too.</p>
  <h2>line-height: 200%</h2>
  <p id="two">CSS is a Style Sheet Language for
    formatting the document content
      (written in HTML or other markup language). You
might be interested
      by the CSS Working Group wiki too.</p>
```

```
</body>
```

vertical-align

The vertical-align property affects the vertical positioning inside a line box of the boxes generated by an inline-level element.

Values	baseline sub super top text-top middle bottom text-bottom <percentage> <length> inherit
---------------	---

Example A

[style.css]

```
img {  
    vertical-align: middle;  
}
```

[index.html]

```
<body>  
  <p>This is a   
  Web site.</p>  
</body>
```

letter-spacing

The letter-spacing property specifies spacing behavior between text characters.

Values	normal <length> inherit
---------------	-----------------------------

Example A

[style.css]

```
p#no{ letter-spacing: normal; }  
p#px{ letter-spacing: 2px; }
```

[index.html]

```
<body>  
  <p id="no">This is a paragraph(normal)</p>  
  <p id="px">This is a paragraph(2px)</p>  
</body>
```

word-spacing

The word-spacing property specifies spacing behavior between words.

Values	normal <length> inherit
---------------	-----------------------------

Example A

[style.css]

```
p#no{ word-spacing: normal; }  
p#px{ word-spacing: 2px; }
```

[index.html]

```
<body>  
  <p id="no">This is a paragraph(normal)</p>  
  <p id="px">This is a paragraph(2px)</p>  
</body>
```

white-space

The white-space property declares how white space inside the element is handled

Values	normal pre nowrap pre-wrap pre-line inherit
--------	---

Example

[style.css]

```
p#no{ white-space: normal; }
p#pre{ white-space: pre; }
```

[index.html]

```
<body>
  <p id="no">
    This is a paragraph.
    This is a paragraph.
    This is a paragraph.
  </p>

  <p id="pre">
    This is a paragraph.
    This is a paragraph.
    This is a paragraph.
  </p>
</body>
```

direction

The direction property specifies the base writing direction of blocks and the direction of embeddings and overrides for the Unicode bidirectional algorithm

Values	ltr rtl inherit
--------	---------------------

Font

font-family

The font-family property value is a prioritized list of font family names and/or generic family names.

Values [<family-name> <generic-family>]
--

Values

- `<family-name>`
The name of a font family of choice.
- `<generic-family>`
The following generic families are defined:
 - serif
 - sans-serif
 - cursive
 - fantasy
 - monospace

Example

[style.css]

```
#se {
  font-family: serif;
}
#sa {
  font-family: Gill, Helvetica, sans-serif;
}
```

[index.html]

```
<body>
  <p id="se">This is a paragraph(serif)</p>
  <p id="sa">This is a paragraph(sans-serif)</p></body>
```

font-style

The font-style property selects between normal, italic and oblique faces within a font family.

Values normal italic oblique inherit

Example

[style.css]

```
p#no{ font-style: normal; }
p#ita{ font-style: italic; }
p#obl{ font-style: oblique; }
```

[index.html]

```
<body>
  <p id="no">This is a paragraph(normal)</p>
  <p id="ita">This is a paragraph(italic)</p>
  <p id="obl">This is a paragraph(oblique)</p>
</body>
```

font-variant

The font-variant property selects the weight of the font.

Values normal small-caps inherit

Example

[style.css]

```
p#no{ font-variant: normal; }
p#sm{ font-variant: small-caps; }
```

[index.html]

```
<body>
  <p id="no">This is a paragraph(normal)</p>
```



```
<p id="sm">This is a paragraph (small-caps)</p></body>
```

font-weight

The font-weight property selects the weight of the font.

Values	normal bold bolder lighter 100 200 300 400 500 600 700 800 900 inherit
--------	--

Example

[style.css]

```
p#no{ font-weight: normal; }
p#b{ font-weight: bold; }
```

[index.html]

```
<body>
  <p id="no">This is a paragraph (normal)</p>
  <p id="b">This is a paragraph (bold)</p>
</body>
```

font-size

The font-size property sets size of the text.

Values	<absolute-size> <relative-size> <length> <percentage> inherit
--------	---

Example

[style.css]

```
p#xxs{ font-size: xx-small; }
p#xs{ font-size: x-small; }
p#s{ font-size: small; }
p#m{ font-size: medium; }
p#l{ font-size: large; }
p#xl{ font-size: x-large; }
```

```
p#xxl{ font-size: xx-large; }
```

[index.html]

```
<body>
  <p id="xss">This is a paragraph(xx-small)</p>
  <p id="xs">This is a paragraph(x-small)</p>
  <p id="s">This is a paragraph(small)</p>
  <p id="m">This is a paragraph(medium)</p>
  <p id="l">This is a paragraph(large)</p>
  <p id="xl">This is a paragraph(x-large)</p>
  <p id="xxl">This is a paragraph(xx-large)</p>
</body>
```

font

The font property is a shorthand property for setting the individual font properties (i.e., 'font-style', 'font-variant', 'font-weight', 'font-size', 'line-height' and 'font-family') at the same place in the style sheet

Example

[style.css]

```
h1{
  font: bold 1.2em Helvetica, Arial, sans-serif;
}
p{
  font: italic 30px Times, Garamond;
}
```

[index.html]

```
<body>
  <h1>font property example</h1>
  <p>This is a paragraph.</p>
```

</body>

Margin & Padding

margin

The margin property is a shorthand property for setting 'margin-top', 'margin-right', 'margin-bottom', and 'margin-left' at the same place in the style sheet

Values

- `<margin-width>{1,4}`
 - `margin: 10px 20px;`
top and bottom paddings are 10px, right and left paddings are 20px.
 - `margin: 10px 20px 30px;`
top padding is 10px, right and left paddings are 20px, bottom padding is 30px.
 - `margin: 10px 20px 30px 40px;`
top padding is 10px, right padding is 20px, bottom padding is 30px, left padding is 40px.

Also individual properties

- [margin-top](#)
- [margin-right](#)
- [margin-bottom](#)
- [margin-left](#)

Example

[style.css]

```
#body{
  background-color: #666;
}
```

```
p{
  background-color: #ccc;
}
#mar{
  margin: 50px 10px;
}
```

[index.html]

```
<body>
  <p>This is a paragraph</p>
  <p id="mar">This is a paragraph</p>
  <p>This is a paragraph</p>
  <p>This is a paragraph</p>
</body>
```

padding

The padding property is a shorthand property for setting 'padding-top', 'padding-right', 'padding-bottom', and 'padding-left' at the same place in the style sheet

Values

- `<padding-width>{1,4}`
 - padding: 10px 20px;
top and bottom paddings are 10px, right and left paddings are 20px.
 - padding: 10px 20px 30px;
top padding is 10px, right and left paddings are 20px, bottom padding is 30px.
 - padding: 10px 20px 30px 40px;
top padding is 10px, right padding is 20px, bottom padding is 30px, left padding is 40px

Example A

[style.css]

```
#body{
  background-color: #666;
}
p{
  background-color: #ccc;
}
#pad{
  padding: 50px 10px;
}
```

[index.html]

```
<body>
  <p>This is a paragraph</p>
  <p id="pad">This is a paragraph</p>
  <p>This is a paragraph</p>
  <p>This is a paragraph</p>
</body>
```

List

list-style-type

The list-style-type property specifies appearance of the list item marker if 'list-style-image' has the value 'none' or if the image pointed to by the URI cannot be displayed.

Description

Values	disc circle square decimal decimal-leading-zero lower-roman upper-roman lower-greek lower-latin upper-latin armenian georgian lower-alpha upper-alpha none inherit
--------	--

Example

[style.css]

```
ul{  
  list-style-type: square;  
}
```

[index.html]

```
<body>  
  <ul>  
    <li>list-item 1</li>  
    <li>list-item 2</li>  
    <li>list-item 3</li>  
  </ul>  
</body>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
  div
```

```
  {
```

```
    width:100px;
```

```
    height:200px;
```

```
    border:2px solid blue;
```

```
    word-wrap:break-word;
```

```
    position:absolute;
```

```
    top:50px;
```

```
    left:50px;
```

```
  }
```

```
div
```

```
{
```

```
transition: width 2s;
```

```
-moz-transition: width 5s; /* Firefox 4 */
-webkit-transition: width 2s; /* Safari and Chrome */
-o-transition: width 2s; /* Opera */
}

div:hover
{
    width:50px;
}

</style>
</head>
<body>
    <div>
        Marlabs

SoftwareSoftwareSoftwareSoftwareSoftwareSoftware
    </div>
</body>
</html>
```

list-style-image

The list-style-image property sets the image that will be used as the list item marker.

Example

[style.css]


```
ul{
  list-style-image: uri(images/list.gif);
}
```

[index.html]

```
<body>
  <ul>
    <li>list-item 1</li>
    <li>list-item 2</li>
    <li>list-item 3</li>
  </ul>
</body>
```

list-style-position

The list-style-position property specifies the position of the marker box with respect to the principal block box.

Description

Values	inside outside inherit
---------------	----------------------------

Example

[style.css]

```
#in{
  list-style-type: circle;
  list-style-position: inside;
}
#out{
  list-style-type: square;
  list-style-position: outside;
}
```

[index.html]

```
<body>
  <ul id="in">
```

```
<li>list-item 1</li>
<li>list-item 2</li>
<li>list-item 3</li>
</ul>
<ul id="out">
  <li>list-item 1</li>
  <li>list-item 2</li>
  <li>list-item 3</li>
</ul>
</body>
```

list-style

The list-style property is a shorthand notation for setting the three properties 'list-style-type', 'list-style-image', and 'list-style-position' at the same place in the style sheet.

Example

[style.css]

```
ul{
  list-style: url("images/list.gif") inside;
}
```

[index.html]

```
<body>
  <ul>
    <li>list-item 1</li>
    <li>list-item 2</li>
    <li>list-item 3</li>
  </ul>
</body>
```

Box Size

- [width](#)
- [min-width](#)
- [max-width](#)

- [height](#)
 - [min-height](#)
 - [max-height](#)
-

display

The display property specifies the type of box used for an element.

Values	inline block list-item run-in inline-block table inline-table table-row-group table-header-group table-footer-group table-row table-column-group table-column table-cell table-caption none inherit
--------	---

Example

[style.css]

```
p {
  display: inline;
}
```

[index.html]

```
<body>
  <p>This is a paragraph 1.</p>
  <p>This is a paragraph 2.</p>
  <p>This is a paragraph 3.</p>
</body>
```

position

The position property specifies the positioning algorithms for an elements.

Description

Values	static relative absolute fixed inherit
---------------	--

Example

[style.css]

```
h1{
  position: absolute;
  top: 100px;
  left: 50px;
}
```

[index.html]

```
<body>
  <h1>position property</h1>
  <p>This is a paragraph.</p>
</body>
```

float

The float property specifies whether a box should float to the left, right, or not at all.

Description

Values	left right none inherit
---------------	-------------------------------

Example

[style.css]

```
p{
  width: 100px;
  height: 100px;
  float: left;
}
p#red{
  background-color: red;
}
p#blue{
  background-color: blue;
}
```

[index.html]

```
<body>
  <p id="red">This is a paragraph</p>
  <p id="blue">This is a paragraph</p>
</body>
```

clear

The clear property indicates which sides of an element's box(es) may not be adjacent to an earlier floating box.

Description

Values	none left right both inherit
---------------	--------------------------------------

Example

[style.css]

```
p{
  width: 100px;
  height: 100px;
  float: left;
```

```
}
p#red{
  background-color: red;
}
p#blue{
  background-color: blue;
}
p#green{
  background-color: green;
  clear: left;
}
```

[index.html]

```
<body>
  <p id="red">This is a paragraph</p>
  <p id="blue">This is a paragraph</p>
  <p id="green">This is a paragraph</p>
</body>
```

z-index

The z-index property specifies the stack order of an element.

Values	auto <integer> inherit
---------------	----------------------------

Example

[style.css]

```
h1{
  z-index: 1;
}
p{
  width: 250px;
  height: 100px;
  position: absolute;
  top: 50px;
  left: 30px;
  z-index: 2;
}
```

```
background-color: #999;  
}
```

[index.html]

```
<body>  
  <h1></h1>  
  <p>This is a paragraph. z-index: 2;</p>  
</body>
```


visibility

The visibility property specifies whether the boxes generated by an element are rendered.

Values	visible hidden collapse inherit
---------------	---------------------------------------

Example

[style.css]

```
p#two{
  visibility: hidden;
}
```

[index.html]

```
<body>
  <p>This is a paragraph 1.</p>
  <p id="two">This is a paragraph 2.</p>
  <p>This is a paragraph 3.</p>
</body>
```

cursor

The cursor property specifies the type of cursor to be displayed for the pointing device.

Description

Values	[[<uri> ,]* [auto crosshair default pointer move e-resize ne-resize nw-resize n-resize se-resize sw-resize s-resize w-resize text wait help progress]] inherit
---------------	--

Example

[style.css]

```
p#au{ cursor: auto; }
```

```
p#cro{ cursor: crosshair; }
p#def{ cursor: default; }
p#poi{ cursor: pointer; }
p#mo{ cursor: move; }
p#ne{ cursor: ne-resize; }
p#sw{ cursor: sw-resize; }
p#tex{ cursor: text; }
p#wai{ cursor: wait; }
p#he{ cursor: help; }
p#pro{ cursor: progress; }
```

[index.html]

```
<body>
  <p id="au">auto</p>
  <p id="cro">crosshair</p>
  <p id="def">default</p>
  <p id="poi">pointer</p>
  <p id="mo">move</p>
  <p id="ne">ne-resize</p>
  <p id="sw">sw-resize</p>
  <p id="tex">text</p>
  <p id="wai">wait</p>
  <p id="he">help</p>
  <p id="pro">progress</p>
</body>
```

Table

caption-side

The caption-side property specifies the position of the caption box with respect to the table box.

Values	top bottom inherit
---------------	------------------------

Example

[style.css]

```
table{
  border-width: 1px;
  border-style: solid;
  border-collapse: collapse;
  width: 400px;
}
tr, th, td{
  border-width:1px;
  border-style: solid;
}
caption{
  caption-side: bottom;
}
```

[index.html]

```
<body>
  <table>
    <caption>Figure 1-1. CSS property</caption>
    <tr>
      <th>Values</th>
      <td>top | bottom | inherit</td>
    </tr>
    <tr>
      <th>Initial value</th>
      <td>top</td>
    </tr>
    <tr>
      <th>Applies to</th>
      <td>table-caption elements</td>
    </tr>
    <tr>
      <th>Inherited</th>
      <td>Yes</td>
    </tr>
  </table>
</body>
```