**Smt. Chandaben Mohanbhai Patel Institute of Computer Applications,CHARUSAT**

**Cascading Style Sheets**

**CSS Background**

CSS background properties are used to define the background effects of an element.

## All CSS Background Properties

|  |  |
| --- | --- |
| **Property** | **Description** |
| [background](http://www.w3schools.com/cssref/css3_pr_background.asp) | Sets all the background properties in one declaration |
| [background-attachment](http://www.w3schools.com/cssref/pr_background-attachment.asp) | Sets whether a background image is fixed or scrolls with the rest of the page |
| [background-color](http://www.w3schools.com/cssref/pr_background-color.asp) | Sets the background color of an element |
| [background-image](http://www.w3schools.com/cssref/pr_background-image.asp) | Sets the background image for an element |
| [background-position](http://www.w3schools.com/cssref/pr_background-position.asp) | Sets the starting position of a background image |
| [background-repeat](http://www.w3schools.com/cssref/pr_background-repeat.asp) | Sets how a background image will be repeated |

## Background Color

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

The background color of a page is defined in the body selector:

## Example:

body {background-color:#b0c4de;}

In the example below, the h1, p, and div elements have different background colors:

## Example:

h1 {background-color:#6495ed;}  
p {background-color:#e0ffff;}  
div {background-color:#b0c4de;}

## Background Image

The background-image property specifies an image to use as the background of an element.

By default, the image is repeated so it covers the entire element.

The background image for a page can be set like this:

## Example

body {background-image:url('paper.gif');}

## Background - Repeat Horizontally or Vertically

By default, the background-image property repeats an image both horizontally and vertically.

Some images should be repeated only horizontally or vertically, or they will look strange, like this:

## Example

body  
{  
background-image:url('gradient2.png');  
}

If the image is repeated only horizontally (repeat-x), the background will look better:

## Example

body  
{  
background-image:url('gradient2.png');  
background-repeat:repeat-x;  
}

## Background - position and no-repeat

Showing the image only once is specified by the background-repeat property:

## Example

body  
{  
background-image:url('img\_tree.png');  
background-repeat:no-repeat;  
}

In the example above, the background image is shown in the same place as the text. We want to change the position of the image, so that it does not disturb the text too much.

## Property Values

|  |  |
| --- | --- |
| **Value** | **Description** |
| left top left center left bottom right top right center right bottom center top center center center bottom | If you only specify one keyword, the other value will be "center" |

The position of the image is specified by the background-position property:

## Example

body  
{  
background-image:url('img\_tree.png');  
background-repeat:no-repeat;  
background-position:right top;  
}

## Background - Shorthand property

As you can see from the examples above, there are many properties to consider when dealing with backgrounds.

To shorten the code, it is also possible to specify all the properties in one single property. This is called a shorthand property.

The shorthand property for background is simply "background":

## Example

body {background:#ffffff url('img\_tree.png') no-repeat right top;}

# CSS background-attachment Property

body  
{   
background-image:url('w3css.gif');  
background-repeat:no-repeat;  
background-attachment:fixed;  
}

Two values are there

1) fixed

2) Scroll

# CSS Text

## All CSS Text Properties

|  |  |
| --- | --- |
| **Property** | **Description** |
| [color](http://www.w3schools.com/cssref/pr_text_color.asp) | Sets the color of text |
| [text-align](http://www.w3schools.com/cssref/pr_text_text-align.asp) | Specifies the horizontal alignment of text |
| [text-decoration](http://www.w3schools.com/cssref/pr_text_text-decoration.asp) | Specifies the decoration added to text |
| [text-indent](http://www.w3schools.com/cssref/pr_text_text-indent.asp) | Specifies the indentation of the first line in a text-block |
| [text-shadow](http://www.w3schools.com/cssref/css3_pr_text-shadow.asp) | Specifies the shadow effect added to text |
| [text-transform](http://www.w3schools.com/cssref/pr_text_text-transform.asp) | Controls the capitalization of text |
| [word-spacing](http://www.w3schools.com/cssref/pr_text_word-spacing.asp) | Increases or decreases the space between words in a text |

## Text Color

The color property is used to set the color of the text.

The default color for a page is defined in the body selector.

## Example

body {color:blue;}  
h1 {color:#00ff00;}  
h2 {color:rgb(255,0,0);}

## Text Alignment

The text-align property is used to set the horizontal alignment of a text.

Text can be centered, or aligned to the left or right, or justified.

When text-align is set to "justify", each line is stretched so that every line has equal width, and the left and right margins are straight (like in magazines and newspapers).

## Example

h1 {text-align:center;}  
p.date {text-align:right;}  
p.main {text-align:justify;}

## Text Decoration

The text-decoration property is used to set or remove decorations from text.

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

## Example

a {text-decoration:none;}

It can also be used to decorate text:

## Example

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

## Text Transformation

The text-transform property is used to specify uppercase and lowercase letters in a text.

It can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word.

## Example

p.uppercase {text-transform:uppercase;}  
p.lowercase {text-transform:lowercase;}  
p.capitalize {text-transform:capitalize;}

## Text Indentation

The text-indent property is used to specify the indentation of the first line of a text.

## Example

p {text-indent:50px;}

# CSS word-spacing Property

## Example

Specify that the space between words in paragraphs should be 30 pixels:

p  
{   
word-spacing:30px;  
}

# CSS3 text-shadow Property

## Example

Basic text-shadow:

h1  
{  
text-shadow: 2px 2px 3px #ff0000;  
}

The text-shadow property is supported in all major browsers.

**Note:** The text-shadow property is not supported in Internet Explorer 9 and earlier versions.

## Definition and Usage

The text-shadow property applies shadow to text.

## Syntax

text-shadow: *h-shadow v-shadow blur color*;

# CSS Font

## All CSS Font Properties

|  |  |
| --- | --- |
| **Property** | **Description** |
| [font](http://www.w3schools.com/cssref/pr_font_font.asp) | Sets all the font properties in one declaration |
| [font-family](http://www.w3schools.com/cssref/pr_font_font-family.asp) | Specifies the font family for text |
| [font-size](http://www.w3schools.com/cssref/pr_font_font-size.asp) | Specifies the font size of text |
| [font-style](http://www.w3schools.com/cssref/pr_font_font-style.asp) | Specifies the font style for text |
| [font-variant](http://www.w3schools.com/cssref/pr_font_font-variant.asp) | Specifies whether or not a text should be displayed in a small-caps font |
| [font-weight](http://www.w3schools.com/cssref/pr_font_weight.asp) | Specifies the weight of a font |

## CSS Font Families

In CSS, there are two types of font family names:

* **generic family** - a group of font families with a similar look (like "Serif" or "Monospace")
* **font family** - a specific font family (like "Times New Roman" or "Arial")

|  |  |  |
| --- | --- | --- |
| **Generic family** | **Font family** | **Description** |
| Serif | Times New Roman Georgia | Serif fonts have small lines at the ends on some characters |
| Sans-serif | Arial Verdana | "Sans" means without - these fonts do not have the lines at the ends of characters |
| Monospace | Courier New Lucida Console | All monospace characters have the same width |

## Font Family

The font family of a text is set with the font-family property.

The font-family property should hold several font names as a "fallback" system. If the browser does not support the first font, it tries the next font.

Start with the font you want, and end with a generic family, to let the browser pick a similar font in the generic family, if no other fonts are available.

**Note**: If the name of a font family is more than one word, it must be in quotation marks, like: "Times New Roman".

More than one font family is specified in a comma-separated list:

## Example

p{font-family:"Times New Roman", Times, serif;}

## Font Style

The font-style property is mostly used to specify italic text.

This property has three values:

* normal - The text is shown normally
* italic - The text is shown in italics

## Example

p.normal {font-style:normal;}  
p.italic {font-style:italic;}

## Font Size

The font-size property sets the size of the text.

Being able to manage the text size is important in web design. However, you should not use font size adjustments to make paragraphs look like headings, or headings look like paragraphs.

Always use the proper HTML tags, like <h1> - <h6> for headings and <p> for paragraphs.

The font-size value can be an absolute, or relative size.

Absolute size:

* Sets the text to a specified size
* Does not allow a user to change the text size in all browsers (bad for accessibility reasons)
* Absolute size is useful when the physical size of the output is known

Relative size:

* Sets the size relative to surrounding elements
* Allows a user to change the text size in browsers

## Set Font Size with Pixels

Setting the text size with pixels gives you full control over the text size:

## Example

h1 {font-size:40px;}  
h2 {font-size:30px;}  
p {font-size:14px;}

# CSS font-variant Property

Set a paragraph to a small-caps font:

p.small  
{  
font-variant:small-caps;  
}

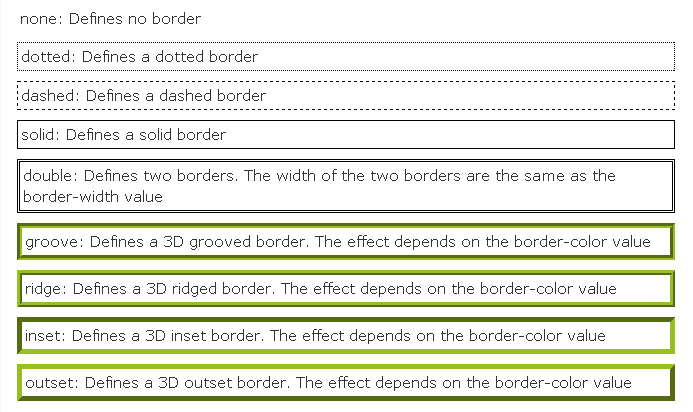
# CSS font-weight Property

Set different font weight for three paragraphs:

p.normal {font-weight:normal;}  
p.thick {font-weight:bold;}  
p.thicker {font-weight:900;}

**CSS Borders**

**Border-style values:**

****

**Example:**

**border-style:solid;**

**The border-style property can have from one to four values.**

* **border-style: dotted solid double dashed;**
  + top border is dotted border-top-style: dotted;
  + right border is solid
  + bottom border is double
  + left border is dashed
* **border-style: dotted solid double;**
  + top border is dotted
  + right and left borders are solid
  + bottom border is double
* **border-style: dotted solid;**
  + top and bottom borders are dotted
  + right and left borders are solid
* **border-style: dotted;**
  + all four borders are dotted

**Border Width:**

* The border-width property is used to set the width of the border.
* The width is set in pixels, or by using one of the three pre-defined values: thin, medium, or thick.
* Note: The "border-width" property does not work if it is used alone. Use the "border-style" property to set the borders first.
* **Example:**

border-width: 5px;

border-width: medium;

**Border Color:**

* The border-color property is used to set the color of the border. The color can be set by:
  + name - specify a color name, like "red"
  + RGB - specify a RGB value, like "rgb(255,0,0)"
  + Hex - specify a hex value, like "#ff0000“
* Note: The "border-color" property does not work if it is used alone. Use the "border-style" property to set the borders first.
* **Example:**

border-color: red;

**Example**:

* The border property is a shorthand for the following individual border properties:
  + border-width
  + border-style (required)
  + border-color
* border: 5px solid red;
* border-radius: value

**CSS Border Radius Properties:**

* The border-radius property is used to give round corners to a box like shape.

|  |  |
| --- | --- |
| Property | Description |
| border-top-left-radius | Specifies the shape of the border of the top-left corner |
| border-top-right-radius | Specifies the shape of the border of the top-right corner |
| border-bottom-left-radius | Specifies the shape of the border of the bottom-left corner |
| border-bottom-right-radius | Specifies the shape of the border of the bottom-right corner |
| border-radius | Specifies a shorthand property for setting all the border radius related properties |

**CSS border-radius Shorthand Example:**

Some examples to use the border-radius shorthand property are given as follows:

* border-radius: 5px 10px 5px 10px / 10px 5px 10px 5px;
* border-radius: 5px;
* border-radius: 5px 10px / 10px;

# CSS Selectors

# Different types of selectors that CSS support are,

# Type Selector(HTML Element )

# Class Selector(.)

# ID Selector(#)

# Grouping Selectors(separate by , )

# Pseudo classes(:)

# Type Selector (HTML Element):

# Type Selectors are very simple.

# They correspond with any HTML element type.

# Example: <HEAD> <style type="text/css"> b {font-family:arial; font-size:14px; color:red} </style> </HEAD> <BODY> <b>This is a customized headline style bold</b> </BODY>

# CLASS Selectors:

# Allows you to define same style for the different elements

# A class selector is a name preceded by a period (.)

# The general syntax for a Class selector is: .ClassSelector {Property:Value;} Example:

# <HEAD> <style type="text/css"> .headline {font-family:arial; font-size:14px; color:red} </style> </HEAD> <BODY> <b class="headline">This is a bold tag carrying the headline class</b> </br> <i class="headline">This is an italics tag carrying the headline class</i> </BODY>

# Combining class and type selectors:

# For two types of paragraphs in your document: one right-aligned paragraph, and one centre-aligned paragraph:

# p.right {text-align: right}

# p.center {text-align: center}

# You have to use the class attribute in your HTML document:

# <p class="right"> This paragraph will be right-aligned. </p>

# <p class="center"> This paragraph will be center-aligned. </p>

# Combining multiple classes

# Perhaps the most powerful aspect of class selectors is that multiple classes can be applied to one HTML element. For example,

# <p class="big indent"> .big { font-weight: bold; } .indent { padding-left: 2em; }

# ID Selectors:

# ID selectors are similar to class selectors. They can be used to select any HTML element that has an ID attribute, regardless of their position in the document tree.

# An ID selector is a name preceded by a hash character (#).

# 

# Should you use ID or class?

# Repeated use within a document

# Classes can be used as many times as needed within a document.

# IDs can only be applied once within a document.

# Combining class selectors

# You can use multiple classes to style an HTML element but

# you can only use one ID when styling an HTML element.

# Example,

# <p class="right color"> This paragraph will be right-aligned. </p>  .right {text-align: right}

# .color {color:green}

# IDs have higher specificity than classes

# If a class selector and ID selector were to be in conflict, the ID selector would be chosen.

# <h3 class="color" id="color1"> This paragraph will be center-aligned. </h3>

# .color {color:green}

# #color1 {color:red}

# Grouping Selectors:

# If multiple selectors have same style then you can combine those selector in one and each selector separate by comma (,).

# If you have elements with the same style definitions, like this:

# h1 {    text-align: center;     color: red; }

# h2 {    text-align: center;     color: red;} p {     text-align: center;    color: red;}

# you can group the selectors, to minimize the code.

# To group selectors, separate each selector with a comma.

# In the example below we have grouped the selectors from the code above:

# Example:

# h1, h2, p {     text-align: center;     color: red; }

# Pseudo-Classes:

# Pseudo Classes allow the designer the freedom to control how the element should appear under different conditions.

# It represents dynamic events to customize those styles.

# A pseudo-class is used to change in state of an element.

# For example, it can be used to:

# Style an element when a user mouse over it

# Style visited and unvisited links differently

# A pseudo-class starts with a colon (:).

# No whitespace may appear between a type selector or universal selector and the colon, nor can whitespace appear after the colon.

# Syntax:

# selector:pseudo-class {     property:value; }

# Dynamic Pseudo Classes:

# They include:

# :link :- specifies unvisited hyperlinks

# :visited :- specifies visited hyperlinks

# :hover :- Applies properties on mouse over.

# :focus :- Applies properties on focus (usually a form input field / if user has used their keyboard to navigate to a link).

# :active:- specifies user currently clicking

# Examples:

# a:link, a:visited { color: blue; } a:hover, a:active { color: red; }

# You can add class names and customized various links a.linkColor:link { color: #FF0066; }

# Example:

# <head>

# <style>

# a:hover { color: #FF00FF; }

# a.new:hover {color:red}

# </style>

# </head>

# <body>

# <p><b><a href=“#" >This is a link</a></b></p>

# <p><b><a href="#" class=“new”>This is a link</a></b></p>

# </body>

# :first-child, :last-child :

# p:first-child CSS pseudo class selects all elements that are the first child of some parent HTML element.

# P:nth-child(2) Selects every <p> element that is the second child of its parent

# P:last-child CSS pseudo class selects all elements that are the last child of some parent HTML element.

# P:nth-last-child(2) Selects every <p> element that is the second child of its parent, counting from the last child

# Example:

# <style>

# p:first-child { background-color: #ff00ff; }

# p:last-child { background-color: #00ff00; }

# p:nth-child(odd) { background: red; }

# p:nth-child(even) { background: blue; }

# </style>

# </head>

# <body>

# <h3>

# <p>I am a <i>strong</i> man. I am a <i>strong</i> man.</p>

# <p>I am a <i>strong</i> man. I am a <i>strong</i> man.</p>

# <p>I am a <i>strong</i> man. I am a <i>strong</i> man.</p>

# <p>I am a <i>strong</i> man. I am a <i>strong</i> man.</p>

# <p>I am a <i>strong</i> man. I am a <i>strong</i> man.</p>

# </h3>

# <h3>

# <p>I am a <i>strong</i> man. I am a <i>strong</i> man.</p>

# <p>I am a <i>strong</i> man. I am a <i>strong</i> man.</p>

# <p>I am a <i>strong</i> man. I am a <i>strong</i> man.</p>

# <p>I am a <i>strong</i> man. I am a <i>strong</i> man.</p>

# <p>I am a <i>strong</i> man. I am a <i>strong</i> man.</p>

# </h3>

# </body>

# CSS Links

Links can be styled in different ways.

## Styling Links

Links can be styled with any CSS property (e.g. color, font-family, background, etc.).

In addition, links can be styled differently depending on what **state** they are in.

**The four links states are:**

* **a:link - a normal, unvisited link**
* **a:visited - a link the user has visited**
* **a:hover - a link when the user mouse over it**
* **a:active - a link the moment it is clicked**

## Example

a:link {color:#FF0000;}      /\* unvisited link \*/  
a:visited {color:#00FF00;}  /\* visited link \*/  
a:hover {color:#FF00FF;}  /\* mouse over link \*/  
a:active {color:#0000FF;}  /\* selected link \*/

When setting the style for several link states, there are some order rules:

* a:hover MUST come after a:link and a:visited
* a:active MUST come after a:hover

## Common Link Styles

In the example above the link changes color depending on what state it is in.

Lets go through some of the other common ways to style links:

## Text Decoration

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

## Example

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 specifies the background color for links:

## Example

a:link {background-color:#B2FF99;}  
a:visited {background-color:#FFFF85;}  
a:hover {background-color:#FF704D;}  
a:active {background-color:#FF704D;}

**CSS Lists**

The CSS list properties allow you to:

* Set different list item markers for ordered lists
* Set different list item markers for unordered lists
* Set an image as the list item marker

## List

## All CSS List Properties

|  |  |
| --- | --- |
| **Property** | **Description** |
| [list-style](http://www.w3schools.com/cssref/pr_list-style.asp) | Sets all the properties for a list in one declaration |
| [list-style-image](http://www.w3schools.com/cssref/pr_list-style-image.asp) | Specifies an image as the list-item marker |
| [list-style-type](http://www.w3schools.com/cssref/pr_list-style-type.asp) | Specifies the type of list-item marker |

In HTML, there are two types of lists:

* unordered lists - the list items are marked with bullets
* ordered lists - the list items are marked with numbers or letters

With CSS, lists can be styled further, and images can be used as the list item marker.

## Different List Item Markers

The type of list item marker is specified with the list-style-type property:

## Example

ul.a {list-style-type: circle;}  
ul.b {list-style-type: square;}  
  
ol.c {list-style-type: upper-roman;}  
ol.d {list-style-type: lower-alpha;}

## An Image as The List Item Marker

To specify an image as the list item marker, use the list-style-image property:

## Example

ul  
{  
list-style-image: url('sqpurple.gif');  
}

When using the shorthand property, the order of the values is:

* list-style-type
* list-style-position (for a description, see the CSS properties table below)
* list-style-image

**Display Property:**

## Property Values

|  |  |
| --- | --- |
| **Value** | **Description** |
| inline | Default. Displays an element as an inline element (like <span>) |
| block | Displays an element as a block element (like <p>) |
| inline-block | Displays an element as an inline-level block container. The inside of this block is formatted as block-level box, and the element itself is formatted as an inline-level box |

**CSS position Property**

* **Definition and Usage**

The position property specifies the type of positioning method used for an element (static, relative, absolute or fixed).

**Property Values**

|  |  |
| --- | --- |
| **Value** | **Description** |
| absolute | The element is positioned relative to its first positioned (not static) ancestor element |
| fixed | The element is positioned relative to the browser window |
| relative | The element is positioned relative to its normal position, so "left:20" adds 20 pixels to the element's LEFT position |

Eg:-

h2  
{  
position:absolute;  
left:100px;  
top:150px;  
}

**CSS Units**

## Measurement Values

|  |  |
| --- | --- |
| **Unit** | **Description** |
| % | Percentage |
| in | inch |
| cm | centimeter |
| mm | millimeter |
| em | 1em is equal to the current font size. 2em means 2 times the size of the current font. E.g., if an element is displayed with a font of 12 pt, then '2em' is 24 pt. The 'em' is a very useful unit in CSS, since it can adapt automatically to the font that the reader uses |
| ex | one ex is the x-height of a font (x-height is usually about half the font-size) |
| pt | point (1 pt is the same as 1/72 inch) |
| pc | pica (1 pc is the same as 12 points) |
| px | pixels (a dot on the computer screen) |

**CSS Tables**

## Table Borders

To specify table borders in CSS, use the border property.

The example below specifies a black border for table, th, and td elements:

## Example

table, th, td  
{  
border: 1px solid black;  
}

## Collapse Borders

The border-collapse property sets whether the table borders are collapsed into a single border or separated:

## Example

table  
{  
border-collapse:collapse;  
}  
table,th, td  
{  
border: 1px solid black;  
}

## Table Width and Height

Width and height of a table is defined by the width and height properties.

The example below sets the width of the table to 100%, and the height of the th elements to 50px:

## Example

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

## Table Text Alignment

The text in a table is aligned with the text-align and vertical-align properties.

The text-align property sets the horizontal alignment, like left, right, or center:

## Example

td  
{  
text-align:right;  
}

The vertical-align property sets the vertical alignment, like top, bottom, or middle:

## Example

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

## Table Padding

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

## Example

td  
{  
padding:15px;  
}

## Table Color

The example below specifies the color of the borders, and the text and background color of th elements:

## Example

table, td, th  
{  
border:1px solid green;  
}  
th  
{  
background-color:green;  
color:white;  
}

**CSS3 Modules (Important Topic)**

CSS3 has been split into "modules". It contains the "old CSS specification" (which has been split into smaller pieces). In addition, new modules are added.

Some of the most important CSS3 modules are:

* **Box Model**
* **Backgrounds and Borders**
* **Text Effects**
* **Gradient**
* **2D Transformations**
* **Transitions**
* **Animations**

**CSS Box Model**

## The CSS Box Model

All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout.

The CSS box model is essentially a box that wraps around HTML elements, and it consists of: margins, borders, padding, and the actual content.

The box model allows us to place a border around elements and space elements in relation to other elements.

The image below illustrates the box model:



Explanation of the different parts:

* **Margin** - Clears an area around the border. The margin does not have a background color, it is completely transparent
* **Border** - A border that goes around the padding and content. The border is inherited from the color property of the box
* **Padding** - Clears an area around the content. The padding is affected by the background color of the box
* **Content** - The content of the box, where text and images appear

In order to set the width and height of an element correctly in all browsers, you need to know how the box model works.

**The total width of the element in the example below is 300px:**

width:250px;  
padding:10px;  
border:5px solid gray;  
margin:10px;

**The total width of an element should be calculated like this:**

Total element width = width + left padding + right padding + left border + right border + left margin + right margin

**The total height of an element should be calculated like this:**

Total element height = height + top padding + bottom padding + top border + bottom border + top margin + bottom margin

## CSS3 Borders

With CSS3, you can create rounded borders, add shadow to boxes, and use an image as a border - without using a design program, like Photoshop.

## CSS3 Border Properties

|  |  |
| --- | --- |
| **Property** | **Description** |
| [border-image](http://www.w3schools.com/cssref/css3_pr_border-image.asp) | A shorthand property for setting all the border-image-\* properties |
| [border-radius](http://www.w3schools.com/cssref/css3_pr_border-radius.asp) | A shorthand property for setting all the four border-\*-radius properties |
| [box-shadow](http://www.w3schools.com/cssref/css3_pr_box-shadow.asp) | Attaches one or more drop-shadows to the box |
| Border-style | Display border in different style |

## CSS3 Rounded Corners

Adding rounded corners in CSS2 was tricky. We had to use different images for each corner.

In CSS3, creating rounded corners is easy.

In CSS3, the border-radius property is used to create rounded corners:

## Example

Add rounded corners to a div element:

div  
{  
border:2px solid;  
border-radius:25px;  
}

The border-style property can have from one to four values

* **border-style:dotted solid double dashed;**
  + top border is dotted
  + right border is solid
  + bottom border is double
  + left border is dashed
* **border-style:dotted solid double;**
  + top border is dotted
  + right and left borders are solid
  + bottom border is double
* **border-style:dotted solid;**
  + top and bottom borders are dotted
  + right and left borders are solid
* **border-style:dotted;**
  + all four borders are dotted

## CSS3 Box Shadow

In CSS3, the box-shadow property is used to add shadow to boxes:

## Example

Add a box-shadow to a div element:

div  
{  
box-shadow: 10px 10px 5px #888888;  
}

## CSS3 Border Image

With the CSS3 border-image property you can use an image to create a border:

## Syntax

border-image: *source slice width repeat*;

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

border-image:url(border.png) 30 30 stretch;  
}

**CSS3 Backgrounds**

CSS3 contains several new background properties, which allow greater control of the background element.

## CSS3 Background Properties

|  |  |
| --- | --- |
| **Property** | **Description** |
| [background-size](http://www.w3schools.com/cssref/css3_pr_background-size.asp) | Specifies the size of the background images |

## CSS3 The background-size Property

The background-size property specifies the size of the background image.

Before CSS3, the background image size was determined by the actual size of the image. In CSS3 it is possible to specify the size of the background image, which allows us to re-use background images in different contexts.

You can specify the size in pixels or in percentages. If you specify the size as a percentage, the size is relative to the width and height of the parent element.

## Example 1

Resize a background image:

div  
{  
background:url(img\_flwr.gif);  
background-size:80px 60px;  
background-repeat:no-repeat;  
}

Stretch the background image to completely fill the content area:

div  
{  
background:url(img\_flwr.gif);  
background-size:100% 100%;  
background-repeat:no-repeat;  
}

|  |  |
| --- | --- |
| CSS3 Multiple Background Images |  |
| CSS3 allows you to use several background images for an element. |

## Example

Set two background images for the body element:

body  
{  
background:url(img\_tree.gif),url(img\_flwr.gif);  
background-size:100% 100%;  
background-repeat:no-repeat;  
}

## CSS3 Text Effects

CSS3 contains several new text features.

In this chapter you will learn about the following text properties:

* text-shadow
* word-wrap

## CSS3 Word Wrapping

If a word is too long to fit within an area, it expands outside:

This paragraph contains a very long word: thisisaveryveryveryveryveryverylongword. The long word will break and wrap to the next line.

In CSS3, the word-wrap property allows you to force the text to wrap - even if it means splitting it in the middle of a word:

This paragraph contains a very long word: thisisaveryveryveryveryveryverylongword. The long word will break and wrap to the next line.

The CSS code is as follows:

## Example

Allow long words to be able to break and wrap onto the next line:

p {word-wrap:break-word;}

CSS3 Fonts

## The CSS3 @font-face Rule

Before CSS3, web designers had to use fonts that were already installed on the user's computer.

With CSS3, web designers can use whatever font he/she likes.

When you have found/bought the font you wish to use, include the font file on your web server, and it will be automatically downloaded to the user when needed.

Your "own" fonts are defined in the CSS3 @font-face rule.

## Using The Font You Want

In the new @font-face rule you must first define a name for the font (e.g. myFirstFont), and then point to the font file.

|  |  |
| --- | --- |
| **Note** | **Tip:** Use lowercase letters for the font URL. Uppercase letters can give unexpected results in IE. |

To use the font for an HTML element, refer to the name of the font (myFirstFont) through the font-family property:

## Example

<style>   
@font-face  
{  
font-family: myFirstFont;  
src: url(sansation\_light.woff);  
}  
  
div  
{  
font-family:myFirstFont;  
}  
</style>

**CSS3 Gradients**

CSS3 gradients let you display smooth transitions between two or more specified colors.

Earlier, you had to use images for these effects. However, by using CSS3 gradients you can reduce download time and bandwidth usage. In addition, elements with gradients look better when zoomed, because the gradient is generated by the browser.

CSS3 defines two types of gradients:

* **Linear Gradients (goes down/up/left/right/diagonally)**
* **Radial Gradients (defined by their center)**

## CSS3 Linear Gradients

To create a linear gradient you must define at least two color stops. Color stops are the colors you want to render smooth transitions among. You can also set a starting point and a direction (or an angle) along with the gradient effect.

**Example of Linear Gradient:**



### Syntax

background: linear-gradient(direction, color-stop1, color-stop2, ...);

**Linear Gradient - Top to Bottom (this is default)**

The following example shows a linear gradient that starts at the top. It starts red, transitioning to blue:

## Example

A linear gradient from top to bottom:

#grad  
{  
background: linear-gradient(red, blue); /\* Standard syntax \*/  
}

**Linear Gradient - Left to Right**

The following example shows a linear gradient that starts from the left. It starts red, transitioning to blue:

## Example

A linear gradient from left to right:

#grad  
{  
background: linear-gradient(to right, red , blue);

}

**Linear Gradient - Diagonal**

You can make a gradient diagonally by specifying both the horizontal and vertical starting positions.

The following example shows a linear gradient that starts at top left (and goes to bottom right). It starts red, transitioning to blue:

## Example

A linear gradient that starts at top left (and goes to bottom right):

#grad  
{  
background: linear-gradient(to bottom right, red , blue);

}

## Using Angles

If you want more control over the direction of the gradient, you can define an angle, instead of the predefined directions (to bottom, to top, to right, to left, to bottom right, etc.).

### Syntax

background: linear-gradient(angle, color-stop1, color-stop2);

The angle is specified as an angle between a horizontal line and the gradient line, going counter-clockwise. In other words, 0deg creates a bottom to top gradient, while 90deg generates a left to right gradient.

The following example shows how to use angles on linear gradients:

## Example

A linear gradient with a specified angle:

#grad  
{  
background: linear-gradient(180deg, red, blue);

}

## Using Multiple Color Stops

The following example shows how to set multiple color stops:

## Example

A linear gradient from top to bottom with multiple color stops:

#grad  
{  
background: linear-gradient(red, green, blue); /\* Standard syntax \*/  
}

## Repeating a linear-gradient

The repeating-linear-gradient() function is used to repeat linear gradients:

## Example

A repeating linear gradient:

#grad  
{  
/\* Standard syntax \*/  
background: repeating-linear-gradient(red, yellow 10%, green 20%);  
}

## CSS3 Radial Gradients

A radial gradient is defined by its center.

To create a radial gradient you must also define at least two color stops. You can also specify the gradient's center, shape (circle or ellipse) as well as its size. By default, center is center, shape is ellipse, and size is farthest-corner.

**Example of Radial Gradient:**



### Syntax

background: radial-gradient(center, shape size, start-color, ..., last-color);

**Radial Gradient - Evenly Spaced Color Stops (this is default)**

## Example

A radial gradient with evenly spaced color stops:

#grad  
{  
background: radial-gradient(red, green, blue); /\* Standard syntax \*/  
}

**Radial Gradient - Differently Spaced Color Stops**

## Example

A radial gradient with differently spaced color stops:

#grad  
{  
background: radial-gradient(red 5%, green 15%, blue 60%); /\* Standard syntax \*/  
}

## Set Shape

The shape parameter defines the shape. It can take the value circle or ellipse. The default value is ellipse.

## Example

A radial gradient with the shape of a circle:

#grad  
{  
background: radial-gradient(circle, red, yellow, green); /\* Standard syntax \*/  
}

## Repeating a radial-gradient

The repeating-radial-gradient() function is used to repeat radial gradients:

## Example

A repeating radial gradient:

#grad  
{  
/\* Standard syntax \*/  
background: repeating-radial-gradient(red, yellow 10%, green 15%);  
}

## CSS3 Transforms

With CSS3 transform, we can move, scale, turn, spin, and stretch elements.

## How Does it Work?

A transform is an effect that lets an element change shape, size and position.

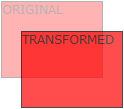
You can transform your elements using 2D or 3D transformation.

## 2D Transforms

In this chapter you will learn about the 2d transform methods:

* translate()
* rotate()
* scale()
* skew()

## The translate() Method



With the translate() method, the element moves from its current position, depending on the parameters given for the left (X-axis) and the top (Y-axis) position:

## Example

div  
{  
-webkit-transform: translate(50px,100px); /\* Safari and Chrome \*/  
}

## The rotate() Method

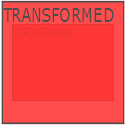


With the rotate() method, the element rotates clockwise at a given degree. Negative values are allowed and rotate the element counter-clockwise.

## Example

div  
{  
-webkit-transform: rotate(30deg); /\* Safari and Chrome \*/  
}

## The scale() Method

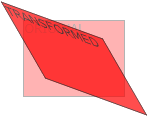


With the scale() method, the element increases or decreases the size, depending on the parameters given for the width (X-axis) and the height (Y-axis):

## Example

div  
{  
-webkit-transform: scale(2,4); /\* Safari and Chrome \*/  
}

## The skew() Method



With the skew() method, the element turns in a given angle, depending on the parameters given for the horizontal (X-axis) and the vertical (Y-axis) lines:

## Example

div  
{  
-webkit-transform: skew(30deg,20deg); /\* Safari and Chrome \*/  
}

## CSS3 Transitions

With CSS3, we can add an effect when changing from one style to another, without using Flash animations or JavaScripts.

## How does it work?

CSS3 transitions are effects that let an element gradually change from one style to another.

**To do this, you must specify two things:**

* **Specify the CSS property you want to add an effect to**
* **Specify the duration of the effect.**

## Example

Transition effect on the width property, duration: 2 seconds:

div  
{  
-webkit-transition: width 2s;

}

**Note:** If the duration is not specified, the transition will have no effect, because default value is 0.

The effect will start when the specified CSS property changes value. A typical CSS property change would be when a user mouse-over an element:

## Example

Specify :hover for <div> elements:

div:hover  
{  
width:300px;  
}

**Note:** When the cursor mouse out of the element, it gradually changes back to its original style.

## Multiple changes

To add a transitional effect for more than one style, add more properties, separated by commas:

## Example

Add effects on the width, height, and the transformation:

div  
{  
-webkit-transition: width 2s, height 2s, -webkit-transform 2s;

}

**Example of Multiple Effects**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<style>**

**div**

**{**

**width:100px;**

**height:100px;**

**background:red;**

**-webkit-transition:width 2s, height 2s, -webkit-transform 2s;**

**}**

**div:hover**

**{**

**width:200px;**

**height:200px;**

**-webkit-transform:rotate(180deg);**

**}**

**</style>**

**</head>**

**<body>**

**<p><b>Note:</b> This example does not work in Internet Explorer 9 and earlier versions.</p>**

**<div>Hover over me to see the transition effect!</div>**

**</body>**

**</html>**

## CSS3 Transition Properties

The following table lists all the transition properties:

|  |  |
| --- | --- |
| **Property** | **Description** |
| [transition](http://www.w3schools.com/cssref/css3_pr_transition.asp) | A shorthand property for setting the four transition properties into a single property |
| [transition-property](http://www.w3schools.com/cssref/css3_pr_transition-property.asp) | Specifies the name of the CSS property to which the transition is applied |
| [transition-duration](http://www.w3schools.com/cssref/css3_pr_transition-duration.asp) | Defines the length of time that a transition takes. Default 0 |
| [transition-timing-function](http://www.w3schools.com/cssref/css3_pr_transition-timing-function.asp) | Describes how the speed during a transition will be calculated. Default "ease" |
| [transition-delay](http://www.w3schools.com/cssref/css3_pr_transition-delay.asp) | Defines when the transition will start. Default 0 |

**CSS3 transition-property Property**

## Example

Hover over a div element, and change the width with a smooth transition effect:

div  
{  
-webkit-transition-property:width; /\* Safari and Chrome \*/  
  
}  
div:hover {width:300px;}

## Definition and Usage

The transition-property property specifies the name of the CSS property the transition effect is for (the transition effect will start when the specified CSS property changes).

**Tip:** A transition effect could typically occur when a user hover over an element.

**Note:** Always specify the transition-duration property, otherwise the duration is 0, and the transition will have no effect.

|  |  |
| --- | --- |
| **JavaScript syntax:** | *object*.style.transitionProperty="width,height" |

## Syntax

transition-property: none|all|*property*;

|  |  |
| --- | --- |
| **Value** | **Description** |
| none | No property will get a transition effect |
| all | All properties will get a transition effect |
| *property* | Defines a comma separated list of CSS property names the transition effect is for |

**CSS3 transition-duration Property**

Let the transition effect last for 5 seconds:

-webkit-transition-duration: 5s;

**CSS3 transition-timing-function Property**

A transition effect with the same speed from start to end:

-webkit-transition-timing-function: linear; /\* Safari and Chrome \*/

## Definition and Usage

The transition-timing-function property specifies the speed curve of the transition effect.

## Syntax

transition-timing-function: linear|ease|ease-in|ease-out|ease-in-out|cubic-bezier(*n*,*n*,*n*,*n*);

|  |  |
| --- | --- |
| **Value** | **Description** |
| linear | Specifies a transition effect with the same speed from start to end (equivalent to cubic-bezier(0,0,1,1)) |
| ease | Specifies a transition effect with a slow start, then fast, then end slowly (equivalent to cubic-bezier(0.25,0.1,0.25,1)) |
| ease-in | Specifies a transition effect with a slow start (equivalent to cubic-bezier(0.42,0,1,1)) |
| ease-out | Specifies a transition effect with a slow end (equivalent to cubic-bezier(0,0,0.58,1)) |
| ease-in-out | Specifies a transition effect with a slow start and end (equivalent to cubic-bezier(0.42,0,0.58,1)) |

**CSS3 transition-delay Property**

## Example

Wait 2 seconds before the transition effect starts:

transition-delay: 2s;

## Definition and Usage

The transition-delay property specifies when the transition effect will start.

The transition-delay value is defined in seconds (s) or milliseconds (ms).

## Syntax

transition-delay: *time*;

## Example

The same transition effects as above, using the shorthand transition property:

div  
{  
transition: width 1s linear 2s;  
}

## CSS3 Animations

With CSS3, we can create animations, which can replace animated images, Flash animations, and JavaScripts in existing web pages.

## CSS3 @keyframes Rule

To create animations in CSS3, you will have to learn about the @keyframes rule.

The @keyframes rule is where the animation is created. Specify a CSS style inside the @keyframes rule and the animation will gradually change from the current style to the new style.

@-webkit-keyframes myfirst /\* Safari and Chrome \*/  
{  
from {background: red;}  
to {background: yellow;}  
}

## CSS3 animation

When the animation is created in the @keyframe, bind it to a selector, otherwise the animation will have no effect.

Bind the animation to a selector by specifying at least these two CSS3 animation properties:

* Specify the name of the animation
* Specify the duration of the animation

## Example

Binding the "myfirst" animation to a div element, duration: 5 seconds:

div  
{  
-webkit-animation: myfirst 5s; /\* Safari and Chrome \*/  
}

## What are Animations in CSS3?

An animation is an effect that lets an element gradually change from one style to another.

You can change as many styles you want, as many times you want.

Specify when the change will happen in percent, or the keywords "from" and "to", which is the same as 0% and 100%.

0% is the beginning of the animation, 100% is when the animation is complete.

For best browser support, you should always define both the 0% and the 100% selectors.

## Example

Change the background color when the animation is 25%, 50%, and again when the animation is 100% complete:

@keyframes myfirst  
{  
0%   {background: red;}  
25%  {background: yellow;}  
50%  {background: blue;}  
100% {background: green;}  
}  
  
@-webkit-keyframes myfirst /\* Safari and Chrome \*/  
{  
0%   {background: red;}  
25%  {background: yellow;}  
50%  {background: blue;}  
100% {background: green;}  
}

## CSS3 Animation Properties

The following table lists the @keyframes rule and all the animation properties:

|  |  |
| --- | --- |
| **Property** | **Description** |
| [@keyframes](http://www.w3schools.com/cssref/css3_pr_animation-keyframes.asp) | Specifies the animation |
| [animation](http://www.w3schools.com/cssref/css3_pr_animation.asp) | A shorthand property for all the the animation properties, except the animation-play-state property |
| [animation-name](http://www.w3schools.com/cssref/css3_pr_animation-name.asp) | Specifies the name of the @keyframes animation |
| [animation-duration](http://www.w3schools.com/cssref/css3_pr_animation-duration.asp) | Specifies how many seconds or milliseconds an animation takes to complete one cycle. Default 0 |
| [animation-timing-function](http://www.w3schools.com/cssref/css3_pr_animation-timing-function.asp) | Describes how the animation will progress over one cycle of its duration. Default "ease" |
| [animation-delay](http://www.w3schools.com/cssref/css3_pr_animation-delay.asp) | Specifies when the animation will start. Default 0 |
| [animation-iteration-count](http://www.w3schools.com/cssref/css3_pr_animation-iteration-count.asp) | Specifies the number of times an animation is played. Default 1 |
| [animation-direction](http://www.w3schools.com/cssref/css3_pr_animation-direction.asp) | Specifies whether or not the animation should play in reverse on alternate cycles. Default "normal" |
| [animation-play-state](http://www.w3schools.com/cssref/css3_pr_animation-play-state.asp) | Specifies whether the animation is running or paused. Default "running" |

[**animation-direction**](http://www.w3schools.com/cssref/css3_pr_animation-direction.asp)**:**

## Definition and Usage

The animation-direction property defines whether or not the animation should play in reverse on alternate cycles.

## Syntax

animation-direction: *value*;

|  |  |
| --- | --- |
| **Value** | **Description** |
| normal | Default value. The animation should be played as normal |
| reverse | The animation should play in reverse direction |
| alternate | The animation will be played as normal every odd time (1,3,5,etc..) and in reverse direction every even time (2,4,6,etc...) |
| alternate-reverse | The animation will be played in reverse direction every odd time (1,3,5,etc..) and in a normal direction every even time (2,4,6,etc...) |

[**animation-play-state**](http://www.w3schools.com/cssref/css3_pr_animation-play-state.asp)**:**

## Definition and Usage

The animation-play-state property specifies whether the animation is running or paused.

## Syntax

animation-play-state: paused|running;

|  |  |
| --- | --- |
| **Value** | **Description** |
| paused | Specifies that the animation is paused |
| running | Specifies that the animation is running |

The two examples below sets all the animation properties:

## Example

Run an animation called myfirst, with all the animation properties set:

div  
{  
/\* Safari and Chrome: \*/  
-webkit-animation-name: myfirst;  
-webkit-animation-duration: 5s;  
-webkit-animation-timing-function: linear;  
-webkit-animation-delay: 2s;  
-webkit-animation-iteration-count: infinite;  
-webkit-animation-direction: alternate;  
-webkit-animation-play-state: running;  
}

## Example

The same animation as above, using the shorthand animation property:

div  
{  
animation: myfirst 5s linear 2s infinite alternate;  
/\* Safari and Chrome: \*/  
-webkit-animation: myfirst 5s linear 2s infinite alternate;  
}