



CONTENTS INCLUDE:

- › New Selectors in CSS3
- › Color Styles
- › Font and Text Styles
- › Background Styles
- › Box & UI Styles

CSS3: Basics

By: Jason Cranford Teague

Cascading Style Sheets (CSS) standard is a stylesheet language that allows web developers to add visual and temporal styles to Web pages using mark-up languages—typically HTML. CSS3 (or CSS Level 3) is the latest iteration of the CSS specification as defined by the World Wide Web Consortium's CSS Workgroup. It builds on features already defined in the CSS1, CSS2, and CSS2.1 specifications. CSS4 is also currently under development by the CSS Workgroup but its features are still experimental.

Because it is a progression of the previous standards rather than a replacement, there are few changes from the previous versions, mostly additions. There are a few cases, however, where older syntax is being depreciated in favor of new syntax. Those cases will be highlighted in this document.

Browser Support

CSS3 is a standard that is supported in part by all modern web browsers. However, web browsers have gradually implemented CSS3 over the last several years, so older browsers may not have properties implemented, have the properties fully implemented, or may be using a browser extension. Universal browser support should be assumed unless otherwise indicated.

Rendering Engines and Browser Extensions

The *rendering engine* (sometimes called layout engine or web browser engine) is a software component used to display and interpret HTML, CSS, JavaScript, or other code used to create web pages.

Extension	Rendering Engine	Browsers Include
-webkit-	Webkit	Webkit
-moz-	Mozilla	Firefox
-o-	Presto	Opera to v14
-ms-	Trident	Internet Explorer

Some of the features in CSS3 (and CSS4) are supported using *browser extensions*, which were used to test experimental features in rendering engines in advance of their inclusion in the full CSS specification. Although referred to as *browser extensions*, they are actually prefixes based on the specific company of origin or the rendering engine used by the browser, rather than the specific, name-brand browser.

Using Browser Extensions

To ensure maximum backwards - and cross-browser compatibility, always include the declaration using all possible browser extensions available for the specific property along with the CSS3 syntax.

```
.mybox {
  -webkit-transform: scale(2);
  -moz-transform: scale(2);
  -o-transform: scale(2);
  -ms-transform: scale(2);
  transform: scale(2);
}
```

It is a good idea to place the W3C version of the property last in order, so that it will override the older browser extension versions if recognized.

Hot Tip

Will I always need browser extensions?

Eventually, after the property is added to the CSS standard, the browser extension is no longer needed, but it could take several years before all legacy browsers have been updated. In the meantime, the website caniuse.com provides an easy to reference chart of supported properties and extensions.

NEW SELECTORS IN CSS3

Selectors are used to indicate the element to be styled and the conditions by which it should be styled. One of the key areas that CSS3 has added to the standard is new pseudo-classes and pseudo-elements, allowing you more precise control over web designs.

Pseudo-class	Pattern Matching
:root	Styled if the element is parent element of the document. This is synonymous with the <html> tag, but has higher specificity.
:empty	Styled if the element has no children.
:last-child	Styled if the element is the last child of another element.
:only-child	Styled if the element is the only child of the parent element.
:first-of-type	Styled if the element is the first of its selector type in the parent element.
:last-of-type	Styled if the element is the last of its selector type in the parent element.
:only-of-type	Styled if the element is the only child of its type in the parent.
:nth-child(#)	Styled if the element is the specific nth child (#) of the parent.
:nth-last-child(#)	Styled if the element is the specific nth child (#) of the parent from the bottom.
:nth-of-type(#)	Styled if the element is the specific nth child (#) of the specific selector type in the parent.
:nth-last-of-type(#)	Styled if the element is the specific nth child of the specific selector of the parent from the bottom.

New Pseudo-Classes

Pseudo-classes are used to reference specific elements on the page that are not specific HTML selectors.

Structural Pseudo-classes

Structural Pseudo-classes are used to reference specific elements based on their location in the DOM (Document Object Model).

UI Pseudo-classes

UI pseudo-classes are used to reference specific elements based on their use to input information by the user.

Pseudo-class	Pattern Matching
:enabled	Styled if the element can be activated (selected, clicked on, or accept text input) or accept focus
:disabled	Styled if the element has the value "disabled" and cannot be selected, clicked on, accept text input, or accept focus.
:check	Styled if the element is a radio, check box, or option form element that is set to an on state.
:indeterminate	Styled if the element is an input checkbox that has had its indeterminate set to true using JavaScript
:default	Styled if the element is the default amongst a group of similar elements. Not supported in Internet Explorer.
:valid	Styled if the element is an input or form that contains content that validates correctly according to the input type setting.
:invalid	Styled if the element is an input or form that contains content that does not validate correctly according to the input type settings.
:in-range	Styled if the element is an input with a value range set and a value within that range.
:out-of-range	Styled if the element is an input with a value range set and a value out of that range.
:required	Styled if the element is an input with the required attribute.
:optional	Styled if the element is an input without the required attribute.
:read-only	Styled if the element cannot be edited by the user. Not widely supported.
:read-write	Styled if the element can be edited by the user. Not widely supported.

Other Pseudo-Classes

Pseudo-class	Pattern Matching
:not(x)	Styled if the element is not "x" where x is a valid selector.
:target	Styled if the element is the target of a link with the specified URI anchor as part of its href.

New Pseudo-Elements

Pseudo-elements are used to reference elements based on their use within the document. Four new ones have been added in CSS3, but have not been widely implemented yet.

Selector	Description
::value	Styled if the element is the value in an input element. Not widely supported.
::choices	Styled if the element is a choice in a form element. Not widely supported.
::repeat-item	Styled if the element is a single item from a repeating sequence. Not widely supported.
::repeat-index	Styled if the element is the current item of a repeating sequence. Not widely supported.

Change in pseudo-element Syntax

The only major change to the actual syntax of CSS is with pseudo-elements, which now uses two colons (::) instead of a single colon.

Example using CSS2.1

```
p:first-letter { font-size: 2em; }
```

Example using CSS3:

```
p::first-letter { font-size: 2em; }
```

Hot Tip

Do I have to use the double colon syntax?

The older syntax is still recognized by all browsers, and will likely be for several years. For best backwards compatibility, it is recommended to use the old syntax or use the old and new syntax redundantly.

COLOR STYLES

There are new ways to define color values as well as gradients and to set the opacity of individual color fills.

HSL

Hue, saturation, and lightness are common ways to define a color value, and are now available to web designs.

Value	Description
Hue	Value from 0-360 indicating the value of the color's hue on a color wheel.
Saturation	Styled if the element is the target of a link with the specified URI anchor as part of its href.
Lightness	Value from 0% to 100% indicating the color tone from black (0%) to white (100%) with 50% being the regular color.

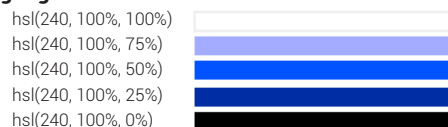
Changing Hue



Changing Saturation



Changing Lightness



HSL is not available in IE until version 9.

Hot Tip

Why HSL?

Using HSL to define colors makes it easier to darken or lighten them by simply changing the lightness value rather than having to recalculate the entire value. This is especially useful for defining link colors, which are often simply different tones of the same color hue.

Opacity

Opacity is used to make an entire element translucent. This includes all of the elements children.

Property	Values	Description
opacity	<number>	Specifies a value between 0 (transparent) and 1 (opaque) for the element's opacity.

50% opacity would be:

```
element1 { opacity: .5 }
```

Color Opacity with Alpha Values

The alpha of a color refers to its opacity as a value between 0 (transparent) to 1 (opaque). CSS3 adds the ability to control opacity of individual color fills by adding an alpha value to RGB and HSL color values. This is done by changing the syntax for the color declarations slightly to rgba() or hsla().

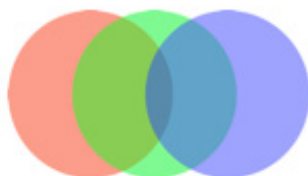
Solid Color:

```
rgb(255, 0, 0);    hsl(0, 100%, 50%);
rgb(0, 255, 0);   hsl(120, 100%, 50%);
rgb(0, 0, 255);   hsl(240, 100%, 50%);
```



Color set to 50% Opacity:

```
rgba(255, 0, 0, .5);    hsla(0, 100%, 50%, .5);
rgba(0, 255, 0, .5);   hsla(120, 100%, 50%, .5);
rgba(0, 0, 255, .5);   hsla(240, 100%, 50%, .5);
```



Alpha values are not available in Internet Explorer until version 9.

Hot Tip

Where can I use the alpha value?
The alpha value can be used anywhere color is declared, including color [text], background, border, and shadows.

Gradients

A gradient allows you to set two or more color stop values that will then fill with color, smoothly transitioning between. Gradients can only be applied to backgrounds.

Linear Gradient

Selector	Description
Angle	Value in degrees (45deg) or a keyword (top, right, bottom, and/or left) to set the angle of the gradient within the element
Color Stop	The color value and location of a particular color stop. Location is optional



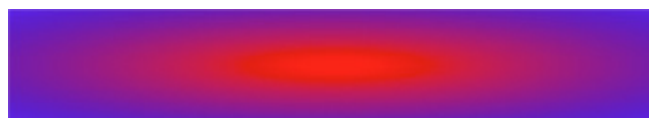
Currently, linear gradients still rely on browser extensions to ensure full cross-browser compatibility, but the syntax is identical.

```
background: -moz-linear-gradient(top, rgba(255,0,0,1) 0%,
  rgba(0,0,225,1) 100%);
background: -webkit-linear-gradient(top, rgba(255,0,0,1)
  0%,rgba(0,0,225,1) 100%);
background: -o-linear-gradient(top, rgba(255,0,0,1)
  0%,rgba(0,0,225,1) 100%);
background: -ms-linear-gradient(top, rgba(255,0,0,1)
  0%, rgba(0,0,225,1) 100%);
background: linear-gradient(top, rgba(255,0,0,1)
  0%, rgba(0,0,225,1) 100%);
```

Sample Radial Gradient

Value	Description
Shape/Size	Keywords: circle or ellipse.
Origin Position	The position of the center of the radial gradient using distance, percentage, or keywords (top, right, bottom, left, or center).
Color Stop	The color value and location of a particular color stop. Location is optional

```
background: radial-gradient(center, ellipse cover,
  rgba(255,0,0,1) 0%, rgba(0,0,225,1) 100%);
```



Currently, radial gradients rely on browser extensions to ensure full cross-browser compatibility, but the syntax is the same.

```
background: -moz-radial-gradient(center, ellipse cover,
  rgba(255,0,0,1) 0%, rgba(0,0,225,1) 100%);
background: -webkit-radial-gradient(center, ellipse cover,
  rgba(255,0,0,1) 0%, rgba(0,0,225,1) 100%);
background: -o-radial-gradient(center, ellipse cover,
  rgba(255,0,0,1) 0%, rgba(0,0,225,1) 100%);
background: -ms-radial-gradient(center, ellipse cover,
  rgba(255,0,0,1) 0%, rgba(0,0,225,1) 100%);
background: radial-gradient(center, ellipse cover,
  rgba(255,0,0,1) 0%, rgba(0,0,225,1) 100%);
```

Hot Tip

How do I calculate gradients?
Since gradients will need to use browser extensions to be fully cross-browser compatible, complex gradients can be tedious to calculate. Use Colorzilla's Ultimate CSS Gradient Generator tool (<http://www.colorzilla.com/gradient-editor/>) to design your gradients and output the full cross-browser code.

FONT AND TEXT STYLES

Font styles affect the appearance of individual letters, while text styles generally are used to affect an entire block of text.

Property	Values	Description
font-size-adjust	none auto <number>	Specifies how a font should be sized based on a ratio of its lowercase letters, rather than uppercase.
font-stretch	normal ultra-condensed extra-condensed semi-condensed semi-expanded expanded extra-expanded ultra-expanded	Specifies the version of the typeface to be used based on common type values for stretch.

Property	Values	Description
text-overflow	clip ellipsis <string>	One or two values that specify how overflow text should be treated in an inline element, either adding an ellipsis (") or a user defined string of characters.
text-shadow	[<color> <offset-x> <offset-y> <blur-radius>] #	Adds one or more shadows underneath all styled text with the specified color, offset, and blur.
text-align-last	auto start end left right center justify	Specifies the text alignment of the last line in a block of text. Not widely supported.
text-decoration-color	<color>	Specifies the color value for the text decoration. Not widely supported.
text-decoration-line	none [underline overline line-through blink]	Specifies the position of the text-decoration. One or more values can be specified, separated by spaces. Not widely supported.
text-decoration-skip	none [images spaces link all]	Specifies element types to skip over when adding a text decoration. Not widely supported.
text-decoration-style	solid double dotted dashed wavy	Specifies the style of the text decoration being used. Not widely supported.
word-wrap	normal break-word	Specifies whether text is allowed to break within words or not.

@font-face

Although technically a part of the CSS2 specification, @font-face is commonly associated with CSS3, where it has come into its own with the availability of webfonts that allow the syntax to work.

Property	Values	Description
font-family	"<text-string>"	Name for the font to use in your font-family declarations. This name can be anything you want, but make sure to put it in quotes.
src	<url>	Path for the font files location on your server and the format the font is using, typically: truetype, opentype, eot, svg, and woff.
font-variant	<font-variant>	Optional. Specifies whether the font is normal or small-caps.
font-stretch	<font-stretch>	Optional. Specifies whether the font is a particular stretch style, such as condensed or expanded.
font-weight	<weight>	Optional. Specifies whether the font is a particular weight, such as bold, bolder, lighter, or 100-900 .
font-style	<style>	Optional. Specifies whether the font is a particular italic or oblique.

The basic @font-face syntax looks like this:

```
@font-face {
  font-family: 'font name';
  src: url('fonts/fontfile.ttf') format('truetype');
  font-stretch: normal;
  font-style: italic;
  font-weight: bold;
}
```

Then use the font name in your CSS:

```
p { font-family: 'font name', times, serif; }
```

Cross-Browser Webfonts

Webfonts are simply font files that can be used by a particular browser to display type. However, different browsers use different file formats, so you will need different sources for each:

```
@font-face {
  font-family: 'font name';
  src: url('fonts/fontfile.eot?') format('embedded-opentype'),
       url('fonts/fontfile.woff') format('woff'),
       url('fonts/fontfile.ttf') format('truetype'),
       url('fonts/fontfile.svg#fontname') format('svg');
}
```

The browser will download the first file format it recognizes and use that.

Hot Tip

Note the question mark (?) after the .eot in that url. This is to help overcome a bug in older versions of Internet Explorer, which would have kept this file from loading.

Hot Tip

Webfont icons

You can use @font-face to load a font with icons instead of letters, numbers, and other symbols, and then use that instead of images to add icons in your interface For a demo, visit: <http://cdpn.io/acsej>

BACKGROUND STYLES

All elements have a background area that can be filled with a solid color, gradient or image. Although part of the CSS2 specification, with CSS3 you can now designate multiple background images on the same element, which has necessitated some slight changes to the properties syntax.

Property	Values	Description
background	[<bg-layer>] #	Specify multiple background colors or images on the same element in a comma-separated list.
background-attachment	[inherit scroll fixed local] #	Specifies how the background scrolls with the element. Local will only scroll with its content, and not with the parent element.
background-clip	[inherit border-box padding-box content-box] #	Specifies whether an element's background, either the color or image, extends underneath its border. This is important when the border is transparent.
background-image	[<url>] #	Specify the path to one or more background images on the same element in a comma-separated list.
background-origin	[inherit border-box padding-box content-box] #	Specifies the location of the top right corner of the background in relation to the element box.
background-position	<position> #	Specifies the initial position of the background in relation to its origin.
background-repeat	[inherit repeat-x repeat-y repeat space round no-repeat] #	Specifies how the background is repeated in the element box.
background-size	[<length> <percentage>] auto cover contain {1,2} #	Specifies a size for the background image. Two values will set width and height. cover fills the entire element, contain shows the entire image in the background.

Box-Shadow

Drop shadows are a common visual effect used to create a feeling of depth on a 2D screen.

Property	Values	Description
box-shadow	none [inset? && [<offset-x> <offset-y> <blur-radius>? <spread-radius>? <color>?]]#	Specify one or more "shadows" of any color in a comma-separated list, behind and/or inside (inset) the element's box. You can also control the horizontal and vertical offset the sharpness of the shadow (blur) and the distance of the solid area from the edge before the shadow fade begins (spread).

```
box-shadow: 2px 3px 5px rgba(0, 0, 0, .5), 0 0 20px rgba(023, 234, 120, .75) inset;
```



Hot Tip

`box-shadow` is only supported in IE 9+ and some older versions of Firefox, Safari, and Chrome may require the browser extension to work.

Border Image

CSS3 introduces the ability to use an image file to create customized border styles. To do this, you will first need to create a bit map image, saved in a web-compatible format (e.g., PNG, GIF, or JPEG).



The CSS slices the image into nine areas: the four corners, the four sides, and the center. The corners of the image will be used at the corners of the box, and each of the four sides are tiled horizontally or vertically



Then, use the properties below to set the image as the elements border.

Property	Syntax	Description
border-image-outset	[<length> <number>]{1,4}	One to four values that specify how far beyond the edge of the box the border image can extend without being clipped.
border-image-repeat	[stretch repeat round]{1,2}	One or two values to specify how the image border behaves horizontally and vertically.
border-image-slice	[<number> <percentage>]{1,4} && fill?	One to four values that specify the distance sliced into the image from its outer top, right, bottom, and left sides to create the image border. If fill is included, the central part of the border image is used as the element's background image.

Property	Syntax	Description
border-image-source	<url>	Specifies the path to the image file being used as the border.
border-image-width	[<length> <percentage> <number> auto]{1,4}	One to four values that specify the offset to use for dividing the border image in nine parts: the top-left corner, central top edge, top-right-corner, central right edge, bottom-right corner, central bottom edge, bottom-left corner, and central right edge.
border-image	<'border-image-source'> <'border-image-slice'> [/ <'border-image-width'>? / <'border-image-outset'>]? <'border-image-repeat'>	Shorthand to specify multiple values for the border image.

In the border-image shorthand, the URL for the image file, the width and the height that the border should slice into the image from its edges, and then how the border image should repeat.

```
border-image:url("flora.png") 30 30 repeat;
```



The border image property is still not fully supported as a CSS standard, but is available in Firefox, Safari, Chrome, and Opera with the appropriate browser extensions:

```
.floralborder {
border: 30px solid transparent;
-moz-border-image:url("flora.png") 30 30 repeat;
-webkit-border-image:url("flora.png") 30 30 repeat;
-o-border-image:url("flora.png") 30 30 repeat;
border-image:url("flora.png") 30 30 repeat;
}
```

Border Radius

Border radius allows for the inclusion of rounded corners. Although referred to as *border radius*, this property really rounds the corners of the box, whether there is a specific border on it or not. The radius can be a single value per corner for a circular radius or two values per corner for an elliptical radius. The border radius can be applied to all four corners or each individually as desired.



Property	Syntax	Description
border-radius	[<length> <percentage>]{1,4} [/ [<length> <percentage>]{1,4}]?	One to four values that specify the radius of the corners' curvature on each corner (top left, top right, bottom right, bottom left). Optionally, a slash (/) and one to four more values can be included to create elliptical curves on each of the four corners.

Property	Syntax	Description
border-top-left-radius, border-top-right-radius, border-bottom-right-radius, border-bottom-left-radius	[<length> <percentage>] [/ [<length> <percentage>]]?	A single value that specifies the radius for a specific corner.

```
border-radius: 20px;
```



```
border-radius: 20px 10px 30px 40px;
```



```
border-radius: 20px/10px;
```



```
border-radius: 20px 10px/10px 20px;
```



Hot Tip
Content in rounded corners is not cropped
 Border radius does not crop the content inside the box, so any image or text in a corner will still appear. However, it will crop the background — including background images — allowing you to create a variety of shapes, including circles and ovals.

BOX & UI STYLES

Overflow allows you to control how content is viewed if it does not fit in the area defined. However, the new **overflow-x** and **overflow-y** allow you to set how overflowing content is controlled independently horizontally and vertically, while the **resize** property allows the user to change the size of the element's box.

Property	Syntax	Description
overflow-y	visible hidden scroll auto	Specifies how vertically overflowing content in an element should be treated for scrolling.
overflow-x	visible hidden scroll auto	Specifies how horizontal overflowing content in an element should be treated for scrolling.
resize	none both horizontal vertical	Specifies whether an element can be resized by the user.

ABOUT THE AUTHOR



Jason (www.jasonspeaking.com) combines creative and technical know-how to help people communicate online. He has worked with businesses and organizations, including USA TODAY, Marriott International, AOL, Virgin, Bank of America, The Aspen Institute, and The Solar Energy Industry Association. Jason is well know for being able to explain complex technical concepts to non-technical audiences. His most recent book is CSS3 Visual Quickstart Guide available in finer book stores everywhere.

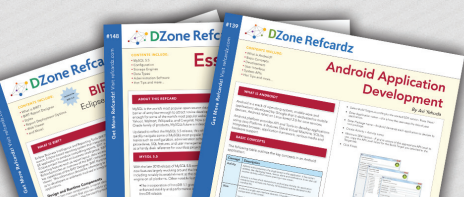
Jason blogs regularly for GeekDad (www.geekdad.com),

RECOMMENDED BOOK



With CSS3: Visual QuickStart Guide readers can start from the beginning to get a tour of the stylesheet language or look up specific tasks to learn just what they need to know. This task-based, visual reference guide uses step-by-step instructions, and plenty of screenshots to teach beginning and intermediate users CSS. Best-selling author Jason Cranford Teague takes readers step-by-step through today's CSS essentials and provides extensive coverage of CSS3 and CSS 2.1 techniques. The book outlines what can be done with CSS3 now and how the latest browsers have implemented many of the new features. Both beginning users, who want a thorough introduction to CSS, and more advanced users, who are looking for a convenient reference, will find what they need in straightforward language and through readily accessible examples.

[Buy Here](#)



Browse our collection of over 150 Free Cheat Sheets

Free PDF

Upcoming Refcardz

- JavaScript Debugging
- Regex
- Open Layers
- HTML5 IndexedDB



DZone communities deliver over 6 million pages each month to more than 3.3 million software developers, architects and decision makers. DZone offers something for everyone, including news, tutorials, cheat sheets, blogs, feature articles, source code and more. **"DZone is a developer's dream",** says PC Magazine.

DZone, Inc.
 150 Preston Executive Dr.
 Suite 201
 Cary, NC 27513
 888.678.0399
 919.678.0300
Refcardz Feedback Welcome
refcardz@dzone.com
Sponsorship Opportunities
sales@dzone.com

