THE FUTURE OF CSS
(or, as much as will fit in 40 minutes)
Zen Garden

The Road to Enlightenment
Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken CSS support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP and the major browser creators.

The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to be) time-honored techniques in new and invigorating fashion. Become one with the web.

So What is This About?
There is clearly a need for CSS to be taken seriously by graphic artists. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The code remains the same, the only thing that has changed is the external .css file.

Yes, really.

CSS allows complete and total control over the style of a hypertext document. The only way this can be illustrated in a way that gets people excited is by demonstrating what it can truly be, once the reins are placed in the hands of those able to create beauty from structure. To date, most examples of neat tricks and hacks have been demonstrated by structurists and coders. Designers have yet to make their mark. This needs to change.

Participation
Graphic artists only please. You are modifying this page, so strong CSS skills are necessary, but the example files are commented well enough that even CSS novices can use them as starting points. Please see the CSS Resource Guide for advanced tutorials and tips on working with CSS.

You may modify the style sheet in any way you wish, but not the HTML. This may seem daunting at first if you've never worked this way before, but follow the listed links to learn more, and use the sample files as a guide.

Download the sample html file and css file to work on a copy locally. Once you have completed your masterpiece (and please, don't submit half-finished work) upload your .css file to a web server under your control. Send us a link to the file and if we choose to use it, we will spider the associated images. Final submissions will be placed on our server.

Pagetop

archives:
next designs »
resources:
View This Design's CSS
CSS Resources
FAQ
Submit a Design
Translations
Browsers & Specs
Our plans to support modern browsers across Google Apps

(Cross-posted to the Google Enterprise Blog and the Gmail Blog)

For web applications to spring even farther ahead of traditional software, our teams need to make use of new capabilities available in modern browsers. For example, desktop notifications for Gmail and drag-and-drop file upload in Google Docs require advanced browsers that support HTML5. Older browsers just don’t have the chops to provide you with the same high-quality experience.

For this reason, soon Google Apps will only support modern browsers. Beginning August 1st, we’ll support the current and prior major release of Chrome, Firefox, Internet Explorer and Safari on a rolling basis. Each time a new version is released, we’ll begin supporting the update and stop supporting the third-oldest version.

As of August 1st, we will discontinue support for the following browsers and their predecessors: Firefox 3.5, Internet Explorer 7, and Safari 3. In these older browsers you may have trouble using certain features in Gmail, Google Calendar, Google Talk, Google Docs and Google Sites, and eventually these apps may stop working entirely.

So if it’s been a a while since your last update, we encourage you to get the latest version of your favorite browser. There are many to choose from:

- Chrome
- Firefox
- Internet Explorer
- Safari

As the world moves more to the web, these new browsers are more than just a modern convenience, they are a necessity for what the future holds.

Posted by Venkat Panchapakesan, Vice President of Engineering

by Google Docs at 11:00 am

Labels: Chrome
### CSS3 Word-wrap - Working Draft

Allows lines to be broken within words if an otherwise unbreakable string is too long to fit.

**Global user stats**:  
*Support: 95.1%*

<table>
<thead>
<tr>
<th>Show all versions</th>
<th>IE</th>
<th>Firefox</th>
<th>Chrome</th>
<th>Safari</th>
<th>Opera</th>
<th>iOS Safari</th>
<th>Opera Mini</th>
<th>Opera Mobile</th>
<th>Android Browser</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 versions back</td>
<td>6.0</td>
<td>4.0</td>
<td>12.0</td>
<td>3.2</td>
<td>10.6</td>
<td>3.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 versions back</td>
<td>7.0</td>
<td>5.0</td>
<td>13.0</td>
<td>4.0</td>
<td>11.0</td>
<td>4.0-4.1</td>
<td>10.0</td>
<td>2.1</td>
<td></td>
</tr>
<tr>
<td>Previous version</td>
<td>8.0</td>
<td>6.0</td>
<td>14.0</td>
<td>5.0</td>
<td>11.1</td>
<td>4.2-4.3</td>
<td>11.0</td>
<td>2.2</td>
<td></td>
</tr>
<tr>
<td>Current</td>
<td>9.0</td>
<td>7.0</td>
<td>15.0</td>
<td>5.1</td>
<td>11.5</td>
<td>5.0</td>
<td>11.1</td>
<td>2.3</td>
<td>4.0</td>
</tr>
<tr>
<td>Near future</td>
<td>10.0</td>
<td>9.0</td>
<td>17.0</td>
<td>6.0</td>
<td>12.0</td>
<td>5.0-6.0</td>
<td>11.1</td>
<td>4.0</td>
<td></td>
</tr>
<tr>
<td>Farther future</td>
<td>10.0</td>
<td>9.0</td>
<td>17.0</td>
<td>6.0</td>
<td>12.1</td>
<td>4.0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
This would be silly
Yet this is what web developers have been doing

http://www.flickr.com/photos/navdeepraj/504596529/
TV already solved this problem

http://www.flickr.com/photos/navdeepraj/504596529/
Warning
nth-child

ul li:nth-child(2n+1) {
  background: #ccc;
}

nth-child
nth-of-type

gallery img:nth-of-type(odd) {
  clear: left;
}


First/Last of Type

section p:first-of-type {
    margin-top: 0;
}
section p:last-of-type {
    margin-bottom: 0;
}
Forms - Enabled?

input:enabled {
    border: solid 1px #000;
}

input:disabled {
    border: solid 1px #CCC;
}
Forms - Required?

input:required:after {
  color: #F00;
  content: " *";
}


Forms - Optional?

input:optional:after {
  content: none;
}
input[type="email"]:valid {
  /* do nothing */
}
input[type="email"]:invalid {
  color: #F00;
}
div {
    border-radius: 5px;
}

div {
    border-radius: 5px;
}

Vendor Prefixes:  
Time to drop them?
div {
  -webkit-border-radius: 5px;
  -moz-border-radius: 5px;
  -ms-border-radius: 5px;
  -o-border-radius: 5px;
  border-radius: 5px;
}
div {
    border-radius: 5px;
}

color modes

h1 {
  color: #FFFFFF;
  color: rgb(255, 255, 255);
  color: rgba(255, 255, 255, 0.8);
  color: hsl(90, 50%, 100%);
}
div {
    background:
    url(shadow-top.gif) 0 0,
    url(shadow-bot.gif) 0 100%;
}
div {
  background: url(shadows.gif) 0 0;

  background: url(shadow-top.gif) 0 0, url(shadow-bot.gif) 0 100%;
}

div {
  background-clip: 180px 100px;
}

div {
  background-clip: 180px 100px;
}

div {
    background-size: 100px 100px;
}
```css
div {
  background-size: 100px 100px;
}
```
div {
  background-position: bottom right;
}
div {
  background-position: bottom right;
}


div {
    background-repeat: repeat;
}

```css
div {
    background-repeat: repeat;
}
```
div {
    background-repeat: space;
}

div {
  background-repeat: round;
}

```html
div {
  background-repeat: round;
}
```
```css
div {
  background-image: linear-gradient(top, #7db9e8 0%, #1e5799 100%);
}
```
div {
    background: #1e5799;
    background: -moz-linear-gradient(top, #1e5799 0%, #7db9e8 100%);
    background: -webkit-gradient(linear, left top, left bottom, color-stop(0%, #1e5799), color-stop(100%, #7db9e8));
    background: -webkit-linear-gradient(top, #1e5799 0%, #7db9e8 100%);
    background: -o-linear-gradient(top, #1e5799 0%, #7db9e8 100%);
    background: -ms-linear-gradient(top, #1e5799 0%, #7db9e8 100%);
    background: linear-gradient(top, #1e5799 0%, #7db9e8 100%);

    filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#1e5799', endColorstr='#7db9e8',GradientType=0 );
}

h1 {
  text-shadow: #000 0 1px 0;
}

3D Text

Use multiple text-shadows to create 3D text on any HTML element. No extra HTML, no extra headaches, just awesomesauce.

Works in the latest builds of Safari, Chrome, Firefox, and Opera.

```html
h1 {
  text-shadow: 0 1px 0 #ccc,
              0 2px 0 #c9c9c9,
              0 3px 0 #bbb,
              0 4px 0 #b9b9b9,
              0 5px 0 #aaa,
              0 6px 1px rgba(0,0,0,.1),
              0 0 5px rgba(0,0,0,.1),
              0 1px 3px rgba(0,0,0,.3),
              0 3px 5px rgba(0,0,0,.2),
              0 5px 10px rgba(0,0,0,.25),
              0 10px 10px rgba(0,0,0,.2),
              0 20px 20px rgba(0,0,0,.15);
}
```
don’t do this

h1 {
  text-shadow: #000 0 0 10px;
}
div {
  box-shadow: #000 0 5px 10px;
}

div {
  box-shadow: inset #000 0 5px 10px;
}
border-image
.box {
    border-width: 44px 46px 44px 47px;
    border-image: url(border.png) 44 46 44 47 repeat repeat;
}

Bacon Ipsum

div {
  padding: 20px;
}
div {
  width: 100%;
  padding: 20px;
}
div {
  -webkit-box-sizing: border-box;
  -moz-box-sizing: border-box;
  box-sizing: border-box;
  width: 100%;
  padding: 20px;
}
div {
  width: calc(100% - 40px);
  padding: 20px;
}
div {
    margin: calc(1em - 1px) calc(5em + 20px);
    background-position: calc(100% - 20px) calc(100% - 20px);
    left: calc(50% - 200px);
}

<table>
<thead>
<tr>
<th>Cities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beijing</td>
</tr>
<tr>
<td>Bogota</td>
</tr>
<tr>
<td>Buenos Aires</td>
</tr>
<tr>
<td>Cairo</td>
</tr>
<tr>
<td>Delhi</td>
</tr>
<tr>
<td>Dhaka</td>
</tr>
<tr>
<td>Guangzhou</td>
</tr>
<tr>
<td>Istanbul</td>
</tr>
<tr>
<td>Jakarta</td>
</tr>
<tr>
<td>Karachi</td>
</tr>
<tr>
<td>Kinshasa</td>
</tr>
<tr>
<td>Kolkata</td>
</tr>
<tr>
<td>Lagos</td>
</tr>
<tr>
<td>Lima</td>
</tr>
<tr>
<td>London</td>
</tr>
<tr>
<td>Los Angeles</td>
</tr>
<tr>
<td>Manila</td>
</tr>
<tr>
<td>Mexico City</td>
</tr>
<tr>
<td>Moscow</td>
</tr>
<tr>
<td>Mumbai</td>
</tr>
<tr>
<td>New York</td>
</tr>
<tr>
<td>Osaka</td>
</tr>
<tr>
<td>Paris</td>
</tr>
<tr>
<td>Rio de Janeiro</td>
</tr>
<tr>
<td>Cities</td>
</tr>
<tr>
<td>--------------</td>
</tr>
<tr>
<td>Beijing</td>
</tr>
<tr>
<td>Bogota</td>
</tr>
<tr>
<td>Buenos Aires</td>
</tr>
<tr>
<td>Cairo</td>
</tr>
<tr>
<td>Delhi</td>
</tr>
<tr>
<td>Dhaka</td>
</tr>
<tr>
<td>Guangzhou</td>
</tr>
<tr>
<td>Istanbul</td>
</tr>
<tr>
<td>Jakarta</td>
</tr>
<tr>
<td>Karachi</td>
</tr>
<tr>
<td>Kinshasa</td>
</tr>
<tr>
<td>Kolkata</td>
</tr>
<tr>
<td>Lagos</td>
</tr>
<tr>
<td>Lima</td>
</tr>
<tr>
<td>London</td>
</tr>
<tr>
<td>Los Angeles</td>
</tr>
<tr>
<td>Manila</td>
</tr>
<tr>
<td>Mexico City</td>
</tr>
<tr>
<td>Moscow</td>
</tr>
<tr>
<td>Mumbai</td>
</tr>
<tr>
<td>New York</td>
</tr>
<tr>
<td>Osaka</td>
</tr>
<tr>
<td>Paris</td>
</tr>
<tr>
<td>Rio de Janeiro</td>
</tr>
</tbody>
</table>

.city li {
  width: 80px;
  overflow: hidden;
  white-space: nowrap;
}
Cities

Beijing  Guangzhou  Lagos  Moscow
Bogota  Istanbul  Lima  Mumbai
Buenos Aires  Jakarta  London  New York
Cairo  Karachi  Los Angeles  Osaka
Delhi  Kinshasa  Manila  Paris
Dhaka  Kolkata  Mexico City  Rio de Janeiro

.city li {
  width: 80px;
  overflow: hidden;
  white-space: nowrap;
  text-overflow: ellipsis;
}
a {
  text-decoration: underline;
}

**text-decoration**
text-decoration

a {
  text-decoration: #00FFFF dotted underline;
}

a {
  text-decoration: #FF8800 wavy line-through;
}
a {
  text-rendering: optimizeLegibility;
}
h1 {
  font-variant-ligatures: common-ligatures;
}
h1 {
  font-variant-ligatures: discretionary-ligatures;
}
h1 {
    font-variant-alternates: historical-forms;
}
h1 {
  font-variant-ligatures: historical-ligatures;
}

\text{tz > k}
@font-feature-values Italicserif {
  @swash delicate 1, flowing 2;
}

d1 {
  font-family: Italicserif, serif;
  font-variant-alternates: swash(flowing);
}
div {
    transform: scale(2);
}


div {
  transform: scale(2);
}


```css
div {
    transform: translate(-20px, -20px);
}
```
div {
  transform: translate(-20px, -20px);
}
div {
    transform: skewX(90);
}
div {
  transform: skewX(90);
}

div {
    transform: rotate(90);
    transform-origin: center center;
}
div {
    transform: rotate(90);
    transform-origin: center center;
}

```css

div {
    transform: rotate(90);
    transform-origin: center center;
}
```
div {
  transform: rotate(90);
  transform-origin: center center;
}
div {
  transform:
    translate(-40px, -40px),
    rotate(10deg);
}
div {
    transform:
        translate(-40px, -40px),
        rotate(10deg);
}
Shadow DOM
div {
  -webkit-appearance: button;
  -webkit-appearance: checkbox;
  -webkit-appearance: menulist;
  -webkit-appearance: menulist-button;
  -webkit-appearance: push-button;
  -webkit-appearance: radio;
  -webkit-appearance: searchfield;
  -webkit-appearance: searchfield-cancel-button;
  -webkit-appearance: searchfield-results-button;
  -webkit-appearance: slider-horizontal;
  -webkit-appearance: textarea;
  -webkit-appearance: textfield;
}

div {
    -webkit-appearance: none;
}

Experimental Shadow DOM styling of <input> elements
<form class="box">
    <input placeholder="Your Name" required>
    <input type="email" placeholder="Your Email" required>
    <input type="submit" value="Send">
</form>
::-webkit-validation-bubble-message {
  font: 16px "proxima nova";
  color: #eee;
  background: #333;
  border: none;
  box-shadow: 0 2px 4px rgba(0,0,0,0.25);
  padding-right: 10px;
}

::-webkit-validation-bubble-arrow {
  background: #333;
  border: none;
  box-shadow: none;
}
Inset & Customized WebKit Scrollbar


WebKit Extras
div {
  background-color: #3d8ba4;
  background-image: url(bg.png);
  -webkit-background-composite: plus-darker;
}
Basic Heading

Outlined text

Thick outline

Hollow outlined text, solid shadow

.t1 {
  -webkit-text-stroke: 1px black;
}
.t2 {
  -webkit-text-stroke: 5px black;
}
.t3 {
  color: rgba(0,0,0,0);
  -webkit-text-stroke: 1px rgba(0,0,0,0.5);
  text-shadow: rgba(0,0,0,0.25) 3px 3px 2px;
}
DINERO
IS NOT SPANISH FOR DINNER

http://trentwalton.com/2011/05/19/mask-image-text/
But don't actually use this, okay?

div {
  -webkit-box-reflect: below 5px
  -webkit-gradient(
    linear, left top, left bottom,
    from(transparent),
    color-stop(0.5, transparent), to(white)
  );
}
Where Next?
Floats Suck
New layout controls?
How about 4?
Multi-column
Bacon Ipsum

Tri-tip chuck tongue biltong meatball corned beef, kielbasa shoulder andouille ball tip. Tenderloin short ribs beef strip steak brisket, turkey meatball shoulder.


Bacon Ipsum

Tri-tip chuck tongue biltong meatball corned beef, kielbasa shoulder andouille ball tip. Tenderloin short ribs beef strip steak brisket, turkey meatball shoulder.


Regions
#source {
  flow: "main-thread";
}
.region {
  content: from-flow("main-thread");
}

Tri-tip chuck tongue biltong meatball corned beef, kielbasa shoulder andouille ball tip. Tenderloin short ribs beef strip steak brisket, turkey meatball shoulder.


#source {
  flow: "main-thread";
}
.region {
  content: from-flow("main-thread");
}
.region:nth-child(1) { content-order: 3; }
.region:nth-child(2) { content-order: 2; }
.region:nth-child(3) { content-order: 1; }

<div id="source">
  ...content here ...
</div>
<div class="region"></div>
<div class="region"></div>
<div class="region"></div>
Bacon Ipsum

Tri-tip chuck tongue biltong meatball corned beef, kielbasa shoulder andouille ball tip. Tenderloin short ribs beef strip steak brisket, turkey meatball shoulder.


#source {
  flow: "main-thread";
}
.region {
  content: from-flow("main-thread");
}
.region:nth-child(1) {content-order: 3;}
.region:nth-child(2) {content-order: 2;}
.region:nth-child(3) {content-order: 1;}

<div id="source">
  ...content here ...
</div>
<div class="region"></div>
<div class="region"></div>
<div class="region"></div>
Template Layout & Grid Layout
Bacon Ipsum


.header {
  position: a;
}
.nav {
  position: b;
}
.content {
  position: c;
}
Bacon Ipsum


.parent {
  grid-columns: auto minmax(min-content, lfr);
  grid-rows: auto minmax(min-content, lfr);
}

header {
  grid-column: 1; grid-row: 1;
}
nav {
  grid-column: 1; grid-row: 2;
}
.content {
  grid-column: 2; grid-row: 2;
}
Bonus 5th Layout Method!
Flexible Box Layout
.parent {
  display: flexbox;
  flex-direction: lr;
  width: 800px;
}

.div1 {
  flex(0, 0, 100px);
}
.div2 {
  flex(5%, 5%, 200px);
}
.div3 {
  flex(5%, 5%, auto);
}
.parent {
  display: flexbox;
  flex-direction: lr;
  width: 800px;
}

.div1 {
  flex(0, 0, 100px);
}

.div2 {
  flex(5%, 5%, 200px);
}

.div3 {
  flex(5%, 5%, auto);
}
.parent {
  display: flexbox;
  flex-direction: lr;
  width: 800px;
}

.div1 {
  flex(0, 0, 100px);
}

.div2 {
  flex(5%, 5%, 200px);
}

.div3 {
  flex(5%, 5%, auto);
}
So... which one(s)?
div {
  background: filter(
    src(sandwich.png),
    saturate(0.5);
  );
}

(this syntax is likely totally wrong)
div {
  background: filter(
    src(sandwich.png),
    saturate(0.5);
  );
}

(this syntax is likely totally wrong)
CSS Shaders
Introducing CSS shaders: Cinematic effects for the web

by Vincent Hardy

Advances in HTML5 and CSS (for example transitions, animations, transforms, text shadows, box-shadows, gradients, SVG) have improved the graphical and interactive richness of HTML. SVG filter effects are now moving to Filter Effects 1.0 to become available in CSS and HTML, in addition to SVG, and will bring effects such as grayscale, sepia tone, or hue-rotate to all web content. More sophisticated effects, such as the chiseled effect shown in Figure 1, will also be possible.

![Filter Effects](http://www.adobe.com/devnet/html5/articles/css-shaders.html)

Figure 1. Filter Effects applied to SVG content, from the svg-wow.org website.

The Adobe CSS shaders proposal has been brought to the W3C FX task force. CSS shaders define a filter effects extensibility mechanism and provide rich, easily animated visual effects to all HTML5 content. They work particularly well with CSS animations and CSS transitions.

The following videos show examples of CSS shaders in action on HTML content.

![CSS Shaders](http://www.adobe.com/devnet/html5/articles/css-shaders.html)
CSS Variables
CSS Variables Module Level 1

Editor's Draft 30 October 2011

This version:
http://dev.w3.org/csswg/css-variables/

Editor's draft:
http://dev.w3.org/csswg/css-variables/

Editors:
Luke Macpherson, Google, Inc., macpherson@google.com
Tab Atkins Jr., Google, Inc.

Copyright © 2011 W3C® (MIT, ERCIM, Keio), All Rights Reserved. W3C liability, trademark and document use rules apply.

Abstract

CSS is a language for describing the rendering of structured documents (such as HTML and XML) on screen, on paper, in speech, etc. This module contains the features of CSS level 3 relating to variables. It includes and extends the functionality of CSS level 2 [CSS21], which builds on CSS level 1 [CSS1]. The main extensions compared to level 2 are the introduction of the variable as a new primitive value type that is accepted by all properties.

Status of this document

This is a public copy of the editors' draft. It is provided for discussion only and may change at any moment. Its publication here does not imply endorsement of its contents by W3C. Don’t cite this document other than as work in progress.

The (archived) public mailing list www-style@w3.org (see instructions) is preferred for discussion of this specification. When sending e-mail, please put the text "css-variables" in the subject, preferably like this: "[css-variables] ...summary of comment..."
CSS...
CSS... 4?
<h1>I’m not bold</h1>
<p>I probably will be bold</p>
<h2>And yet I won’t be</h2>
ol ol li, ul ul li, ol ul li, ul ol li, dl dl dt, dl dl dd {
    font-size: 100%;
}

:matches(ol, ul, dl)
:matches(ol, ul, dl)
:matches(li, dt, dd) {
    font-size: 100%;
}
<a href="/">Home</a>
<a href="/products/">Products</a>
<a href="/about/">About</a>
<a href="/contact/">Contact</a>
<a href="http://www.google.com/">Search</a>
<ul>
  <li>1</li>
  <li>2</li>
  <li>3</li>
</ul>

$ul > li:only-child {
  font-weight: bold;
}
Pseudo-Elements Out
Thanks!