DESIGNING THE HI-DPI WEB
HELLO MY NAME IS
DAVID DEMAREE
@dudemaree on your favorite social network
WHAT IS “HI-DPI”
Hi-DPI device characteristics

A high density display
Typically > 160 pixels per inch

Scaled UI graphics
Scaling factor

Ratio of screen pixels to CSS pixels on a single dimension

**CSS**

- height: 2px
- width: 2px

**Standard (1x)**

**Android ‘hdpi’ (1.5x)**

**‘Retina’/‘xhdpi’ (2x)**
There is no such thing as half a screen pixel, therefore artwork is blended/scaled on Android.

CSS

- height: 2px
- width: 2px
Designing the Hi-DPI Web
Designing the Hi-DPI Web
Retina “disasters”
Fine detail + dark/black background = fuzzy rendering at 2x
SELECT COUNTRY:

UNITED STATES
UNITED KINGDOM
GERMANY
JAPAN 日本
CHINA 中国

Rendered using Cufon (HTML5 Canvas)
Why you should support hi-DPI devices
Future friendly
Future friendly
Fit & finish
Future friendly
Fit & finish
Clearer text/graphics aid usability
Future friendly
Fit & finish
Clearer text/graphics aid usability
Why you might not support hi-DPI devices
Bandwidth
Bandwidth

Legacy browser/device support
Bandwidth

Legacy browser/device support

Your site looks good enough at 1x
HI-DPI RULES OF THUMB
HI-DPI RULES OF THUMB

CSS & web fonts
where possible
HI-DPI RULES OF THUMB

CSS & web fonts where possible

SVG or icon fonts where appropriate
HI-DPI RULES OF THUMB

CSS & web fonts
where possible

SVG or icon fonts
where appropriate

Responsive images
as needed
CSS3
<table>
<thead>
<tr>
<th>Feature</th>
<th>9+</th>
<th>✓</th>
<th>✓</th>
<th>✓</th>
<th>✓</th>
<th>✓</th>
<th>✓</th>
<th>✓</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSS3 colors (rgba, hsla)</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Border radius</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Multiple backgrounds</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Box shadow</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>2D/3D transforms</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Text shadow</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Gradients</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Border images</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Text stroke</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Masks</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Manage & Refill Prescriptions

- Refill from account
- Refill with Rx number
- View order status
- Get a flu shot
Manage & Refill Prescriptions

- Refill from account
- Refill with Rx number
- View order status
- Get a flu shot

Full CSS3 support (Chrome)

Manage & Refill Prescriptions

- Refill from account
- Refill with Rx number
- View order status
- Get a flu shot

No/limited support (IE 7-8)
Vendor prefixes

You hate ‘em, right?

```css
.navigation-bar {
  background-color: #04ae4f;
  background-image: -webkit-gradient(linear, left top, left bottom,
                             color-stop(0%, #04ae4f), color-stop(100%, #048ac2));
  background-image: -webkit-linear-gradient(top, #04ae4f, #048ac2);
  background-image: -moz-linear-gradient(top, #04ae4f, #048ac2);
  background-image: -ms-linear-gradient(top, #04ae4f, #048ac2);
  background-image: -o-linear-gradient(top, #04ae4f, #048ac2);
  background-image: linear-gradient(top, #04ae4f, #048ac2);
  color: white;
  border-bottom: 1px solid #ccc;
  display: table;
  width: 100%;
}
```
Use Sass
Getting Started with Sass

by DAVID DEMAREE

Published in: CSS, Scripting

CSS’ simplicity has always been one of its defining, most welcome features. CSS style sheets are just long lists of rules, each consisting of a selector and some styles to apply to the elements with the same selector. As websites got bigger and become more complex, and target a wider range of devices and screen sizes, this simplicity—so welcome as we first started to move away from...
SASS + COMPASS

```css
#linear-gradient {
  @include background-image(linear-gradient(left top, white, #dddddd));
}
```
SASS + COMPASS

```scss
#linear-gradient {
    @include background-image(linear-gradient(left top, white, #dddddd));
}
```

COMPILES TO CSS

```css
#linear-gradient {
    background-image: -webkit-gradient(linear, 0% 0%, 100% 100%, color-stop(0%, #ffffff), color-stop(100%, #dddddd));
    background-image: -webkit-linear-gradient(left top, #ffffff, #dddddd);
    background-image: -moz-linear-gradient(left top, #ffffff, #dddddd);
    background-image: -o-linear-gradient(left top, #ffffff, #dddddd);
    background-image: linear-gradient(left top, #ffffff, #dddddd);
}
```
WEB FONTS
Web font advantages

Real, native browser text
Scales to any size
Robust cross-browser support
RUBYCONF 2012
HTTP://RUBYCONF.ORG/

*** TICKET PURCHASE ***

========================================
2012-09-19 15:03:12 -0600

========================================

CUSTOMER INFORMATION

DAVID DEMAREE
DEMAREE@ADOBE.COM
ADOBE SYSTEMS
1842 N HUMBOLDT BLVD
60647

PAYMENT INFORMATION

DAVID DEMAREE
MASTERCARD

TICKET INFORMATION

SECOND PLACE (SECOND PLACE)
CONFIRMATION # $350.00
DAVID DEMAREE
ADOBE SYSTEMS
MEN'S XL

TOTAL $350.00

----------------------------------------
YOUR CREDIT CARD STATEMENT WILL READ
"BUSYCONF, LLC"
----------------------------------------
Thank you!
Here's your receipt.

BusyConf.com
Simple receipt page made delightful through creative use of web fonts
Sites we like
SPELLTOWER
A GAME OF WORDS

FIND & ELIMINATE WORDS

4 PUZZLING MODES

LONGER WORDS, LARGER REWARDS

TACKLE TILES & KEEP YOUR TOWER LOW
LBVD / LIGHTS, CAMERA, HAM! / A BIG BANK JOB / WILL WORK FOR FREE

IDEAS THAT WORK / Luck, unfortunately, does not count as financial planning.

HOWARD LUCK GOSSAGE
THE "HEY HONEY" TEST /
Every hit series has a blooper reel.

THE ALL-AMERICAN HAM

EMPOWERED BIRDS
DEVIL'S MUSIC / Who says they don't make 'em like they used to?

How do you sell phones? Allow us to demonstrate. / GARDEN GROWTH

Sometimes advertising is the last thing we do.
The Coop is Chicago’s favorite coworking spot. We have desks for rent by the day, week or month.

*We’re affordable, fun and gosh darnit people like us. Join now!*

NEW ON THE BLOG:
Lunchin’ & Learnin’
Challenges
Challenges

Serving multiple font formats
Challenges

Serving multiple font formats
Cross-platform rendering
Font loading & fallbacks
Licensing
Hundreds of commercial typefaces

Licensing + CDN hosting included

Unlimited websites +
500,000 pageviews for $49/yr

Also included with
Adobe Creative Cloud membership
Adobe Edge Web Fonts

New free font service powered by Typekit technology

Hundreds of free/open-source fonts, including **Cooper Black**

No signup required

Integrated into Edge Tools & Services
ICON FONTS
David Demaree
ddemaree

ADOBE SYSTEMS
Chicago, U.S.A.
ddemaree@adobe.com
http://demaree.me/
Joined on Feb 26, 2008

GiftBox
Simple gift tracker for iOS
Last updated 8 days ago

Objective-C ★ 1 0

i18n
Internationalization (i18n) library for Ruby
Last updated 3 months ago

Ruby ★ 1 101

annotations
Extracts and displays source code annotations, e.g. "# TODO: Something"
Last updated 4 months ago

Ruby ★ 2 1

BubbleWrap
Cocoa wrappers and helpers for RubyMotion (Ruby for iOS) - Making Cocoa APIs more Ruby like, one API at a time. Fork away and send your pull requests.
Last updated 4 months ago

Ruby ★ 1 106
David Demaree
ddemaree

Repositories

GiftBox
Simple gift tracker for iOS
Last updated 8 days ago

i18n
forked from svenfuchs/i18n
Internationalization (i18n) library for Ruby
Last updated 3 months ago

annotations
Extracts and displays source code annotations, e.g. "# TODO: Something"
Last updated 4 months ago

BubbleWrap
forked from rubymotion/BubbleWrap
Cocoa wrappers and helpers for RubyMotion (Ruby for iOS) - Making Cocoa APIs more Ruby like, one API at a time. Fork away and send your pull requests.
Last updated 4 months ago

Objective-C
1
0

Ruby
1
101

2
1

Followers: 22
Stargazers: 49
Following: 13

Tweets

Locations
Adobe Systems
Chicago, U.S.A.
cdemaree@adobe.com
http://demaree.me/

Joined on Feb 26, 2008

Edit Your Profile

Adornments

Unlock

Star

Watch

Share

Create

Fork

Comment

Edit

Delete

Favorite

Copy

Create

Fork

Comment

Edit
No expenses found
Try another category, perhaps?

```
!-- Look, ma, no extra markup -->
<div class="blank-slate">
  <h2>No expenses found</h2>
  <p>Try another category, perhaps?</p>
</div>

.blank-slate:before {
  color: #999;
  content: '\1F4E5'; /* Unicode code point */
  display: block;
  font-family: 'SSStandard'; /* Icon font */
  font-size: 300%;
}
Delete this expense
Single color only

font-family: ‘SSSocial’
content: ‘flickr’
Single color only

font-family: ‘SSSocial’
content: ‘flickr’
Flickr’s actual logo is rendered in 2 colors, which icon fonts don’t support.
Accessibility

Generated content is invisible to search engines, but not to assistive devices

Inbox
Accessibility
Generated content is invisible to search engines, but not to assistive devices

This will be read by VoiceOver on iOS as:

“Inbox Tray, Inbox.”
ALL ICONS BELOW ARE THE PICTOS FONT. THERE ARE NO IMAGES.

Using the Pictos Font and CSS3 you can make some awesome & lightweight toolbars. Try clicking the icons. View in Webkit for best results. Zero images used.

ADVANTAGE

Load Speed & File size: Since each icon is a sprite, the load times will be three times faster. Not only that, but a single sprite will load all the images for each icon at once.

Scalability: Image sprites can be scaled up or down without losing quality.

Design on the Fly: Change your designs without changing or updating the code. This makes it easy to experiment.

Browser Support: One of the best advantages is that CSS sprites work in all the latest browsers.

Complex lighting, shadowing and gradient effects made possible by CSS3.
“Capital D, link”
Symbolsets are semantic symbol fonts. They work in modern browsers and anywhere OpenType features are supported.
Symbolsets are semantic symbol fonts. They work in modern browsers and anywhere OpenType features are supported.
SVG
SVG browser support

✓ These browsers/devices will see no image, or else will need a bitmap fallback

✗ These browsers/devices will see no image, or else will need a bitmap fallback
#logo {
  /* 1x PNG for IE 7-8/Android 2-3 */
  background-image: url('/assets/logo.png');

  /* SVG for everyone else */
  background-image: url('/assets/logo.svg');
}

Same artwork, but **96 KB larger** from inefficient SVG encoding
TIP: Use “SVG Tiny” profile when exporting artwork
RESPONSIVE IMAGES
The bad news

There is no single solution (yet) for serving high-resolution images
Proposed web standards

Easy, native responsive image support coming to a browser near you at some point
Proposed `<picture>` syntax
Not implemented by any browser as of Oct 2012

```html
<picture>
  <source srcset="small-1.jpg 1x, small-2.jpg 2x">
  <source media="(min-width: 18em)"
          srcset="med-1.jpg 1x, med-2.jpg 2x">
  <source media="(min-width: 45em)"
          srcset="large-1.jpg 1x, large-2.jpg 2x">
  <!-- fallback for legacy browsers -->
  <img src="small-1.jpg">
</picture>
```
Proposed `<picture>` syntax

Not implemented by any browser as of Oct 2012

```
<picture>
  <source srcset="small-1.jpg 1x, small-2.jpg 2x">
  <source media="(min-width: 18em)"
           srcset="med-1.jpg 1x, med-2.jpg 2x">
  <source media="(min-width: 45em)"
           srcset="large-1.jpg 1x, large-2.jpg 2x">
  <!-- fallback for legacy browsers -->
  <img src="small-1.jpg">
</picture>
```
Proposed <picture> syntax

Not implemented by any browser as of Oct 2012

```html
<picture>
  <source srcset="small-1.jpg 1x, small-2.jpg 2x">
  <source media="(min-width: 18em)"
          srcset="med-1.jpg 1x, med-2.jpg 2x">
  <source media="(min-width: 45em)"
          srcset="large-1.jpg 1x, large-2.jpg 2x">
  <!-- fallback for legacy browsers -->
  <img src="small-1.jpg">
</picture>
```
Proposed `<picture>` syntax

Not implemented by any browser as of Oct 2012

```html
<picture>
  <source srcset="small-1.jpg 1x, small-2.jpg 2x">
  <source media="(min-width: 18em)"
          srcset="med-1.jpg 1x, med-2.jpg 2x">
  <source media="(min-width: 45em)"
          srcset="large-1.jpg 1x, large-2.jpg 2x">
  <!-- fallback for legacy browsers -->
  <img src="small-1.jpg">
</picture>
```
Proposed image-set syntax
Available today in Chrome, Safari 6, iOS 6

#selector {
  background-image: url(no-image-set.png);
  background-image: -webkit-image-set(url(myimage.jpg) 1x,
                                      url(myimage-hires.jpg) 2x);

  /* other prefixes for -moz, -o and -ms go here */
}
What are our options today?
Just serve one resolution
Different kinds of images require different solutions
Different kinds of images require different resolutions.
ST. PETERSBURG OPENING IN 2013

LOCAL TIME
05:14 PM

LOCAL TEMPERATURE
15°C / 59°F

MAKE A RESERVATION

Select a Destination

CHECK IN: 10/04/2012

CHECK OUT: 10/05/2012
Small, finely detailed logos look bad on Retina, especially on dark backgrounds.
1280×486px photo looks okay on Retina
Sample photo @ 1x
1280×486 = 193 KB
Sample photo @ 1x
1280×486 = 193 KB

Sample photo @ 2x
2560×972 = 602 KB
CSS @media queries
@media queries + background-size

#logo {
  background-image: url(/assets/logo.png);
  background-size: 400px, 200px;
}

@media screen and (-webkit-min-device-pixel-ratio: 1.5) {
  #logo {
    background-image: url("/assets/logo@2x.png");
  }
}

@media queries + background-size

```css
#logo {
  background-image: url(/assets/logo.png);
  background-size: 400px, 200px;
}
@media screen and (min-resolution: 1.5) {
  #logo {
    background-image: url(/assets/logo@2x.png);
  }
}
```

background-size fixes the image at its 1x size (in CSS pixels), otherwise 2x versions would be twice as large.
@media queries + background-size

```css
#logo {
    background-image: url(/assets/logo.png);
    background-size: 400px, 200px;
}

@media screen and (-webkit-min-device-pixel-ratio: 1.5) {
    #logo {
        background-image: url("/assets/logo@2x.png");
    }
}
```

Just like width breakpoints in a responsive grid, media queries establish **pixel density breakpoints** for replacing 1x images with higher-quality ones.
@media queries + background-size

```css
#logo {
  background-image: url(/assets/logo.png);
  background-size: 400px, 200px;
}

@media screen and (-webkit-min-device-pixel-ratio: 1.5) {
  #logo {
    background-image: url("/assets/logo@2x.png");
  }
}
```
It wouldn’t be CSS if there weren’t f$%&ing vendor prefixes

```css
@media
  only screen and (-webkit-min-device-pixel-ratio: 2),
  only screen and ( min--moz-device-pixel-ratio: 2),
  only screen and ( -o-min-device-pixel-ratio: 2/1),
  only screen and ( min-device-pixel-ratio: 2),
  only screen and ( min-resolution: 192dpi),
  only screen and ( min-resolution: 2dppx) {
/* Retina-specific stuff here */
}
```
Responsive CSS sprites

```css
.app-icon {
  background-image: url(icons.png);
}

@media screen and (min-device-pixel-ratio: 1.5) {
  .app-icon {
    background-image: url(icons@2x.png);
    /* Size of 1x sprite set */
    background-size: 750px 144px;
  }
}
```
Responsive CSS sprites

```css
.app-icon.logo {
  /* All other measurements in CSS px */
  background-position: -648px -90px;
  width: 82px;
}
```
HTML + JavaScript
devicePixelRatio property
Available in WebKit browsers + Opera

// Returns the scaling factor, e.g. 1.5 or 2
if((window.devicePixelRatio === undefined ? 1 : window.devicePixelRatio) > 1) {
  document.cookie='HTTP_IS_RETINA=1;path=/';
}

Sets a cookie ‘HTTP_IS_RETINA’ if scaling factor > 1
Naïve JavaScript replacement on Apple.com

Changes the src attribute of all <img> tags inside a container

Uses data attributes to track opt into replacement, track which images have been replaced

Loads both 1x and 2x versions, with noticeable “flash of non-Retina content”
**Picturefill** polyfills the proposed `<picture>` syntax using `<div>` elements and JavaScript

```html
<div data-picture>
  <div data-src="external/imgs/pie.jpg"></div>
  <div data-src="external/imgs/pie@1.5x.jpg" data-media="(min-device-pixel-ratio: 1.5)"></div>
  <div data-src="external/imgs/pie@2x.jpg" data-media="(min-device-pixel-ratio: 2)"></div>

  <!-- Fallback content for non-JS browsers -->
  <noscript><img src="external/imgs/pie.jpg"></noscript>
</div>
```
No silver bullets
Use the most appropriate strategy for each kind of image
Use the most appropriate strategy for each kind of image

Whenever possible, use CSS
Use the most appropriate strategy for each kind of image

Whenever possible, use CSS

Try serving only one resolution
Use the most appropriate strategy for each kind of image

Whenever possible, use CSS

Try serving only one resolution

Use Picturefill if true responsive images are needed
THANK YOU
BYE MY NAME WAS

DAVID DEMAREE

✉ ddemaree@adobe.com
✈ @ddemaree
📞 log.demaree.me
typekit.com
html.adobe.com