SVG
is for everybody
SVG is

Sharp at any size or resolution

Small in file size
SVG is

**Styleable & Responsive** - you can apply CSS

**Scriptable** - shapes can be in the DOM

**Systematic** - easy to build an icon system

**Animateable** - in many comfortable ways

**Interactive** - links! hovers! games!

**Accessible** - in all the right ways
Big Promises / Can’t Use
Nope. You can use SVG now and it’s pretty dang easy.
## SVG (basic support)

Method of displaying basic Vector Graphics features using the embed or object elements. Refers to the SVG 1.1 spec.

<table>
<thead>
<tr>
<th>Current aligned</th>
<th>Usage relative</th>
<th>Show all</th>
</tr>
</thead>
<tbody>
<tr>
<td>IE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Firefox</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Chrome</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Safari</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>Opera</td>
<td></td>
<td></td>
</tr>
<tr>
<td>iOS Safari</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Opera Mini</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Android Browser</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chrome for Android</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Partial support in Android 4 refers to not supporting masking.
Even if you need to go deeper, it’s not that bad. We’ll get into that in a bit.
Hey

that looks like HTML
(It's XML)
SVG syntax can be learned!

Unlike any other image format.
(You don’t “learn” JPG, unless you’re like Rain Man.)
SVG can be made from essentially anything “vector”.

.epg / .pdf / .ai / .whatever
open in Illustrator, Save As .svg
“Scaleable Vector Graphics”

SVG can be made from essentially anything “vector”.
SVG will never replace raster graphics.
Rasterfarian *GET IT?*

(This image would be very *bad* at being vector.)
(This image would be very good at being vector.)

wpapcommunity.com/site/index.php/gallery/musisi-jazz/bob-marley-rastaman-2048
Todd Parker’s Chart

DEFINITELY USE SVG

TEST & DECIDE

USE BITMAP

https://docs.google.com/presentation/d/1CNQLbqC0krocy...
How do you get SVG to use?
Learn the SVG syntax and hand-craft them.
<svg viewBox="0 0 100 100" xmlns="http://www.w3.org/2000/svg">
  <rect x="10" y="10" width="100" height="100" />
  <circle cx="30" cy="30" r="30" fill="green" />
  <polygon points="60,20 100,40 100,80 60,100 20,80 20,40" />
  <path stroke="sienna" stroke-width="2" fill="none" d="M 80,180
    Q  50,120  80, 60
    Q  90, 40  80, 20
    Q 100, 20 120, 20
    Q 110, 40 120, 60
    Q 150,120 120, 180
    Z" />
  <!-- there are a few more -->
</svg>
Basic Shapes

- line
- rect
- circle
- ellipse
- polygon
- polyline
- path
Make your own images in a vector editing program.
(p.s. you can copy and paste SVG out of Illustrator CC)
Illustrator

$19.99 / month

Extremely robust
Lots of training/tutorials available
Professional digital design for Mac.

Sketch gives you the power, flexibility and speed you always wanted to create amazing designs in an easy-to-use package. Finally you can focus on what you are good at: design. And Sketch is great for web and great exporting.

Available on the Mac App Store for just $99. Requires OS X 10.8+

Special prices are available for Volume and Educational licenses. You can also Extend an existing license with additional features.

Work better, faster.

Sketch is built for modern graphic designers. Easily create complex shapes with our state-of-the-art vector engine. With a new, intuitive interface, Sketch makes it fast to design and easy to share.

Produce better results.

Easily make complex shapes with our state-of-the-art vector engine. Sketch makes it fast to design and easy to share. Create stunning designs that are ready for print, web or both. With Sketch, you can create high-quality graphics that will impress your clients.

Buy now

Sketch

$99
Draw vectors, export SVG code.
Spend your time drawing, not writing code.

WebCode
$39.99

"WebCode is, without question, the best choice for easy JavaScript+Canvas and CSS drawing available today. At $49.99, WebCode is a bargain, and a real value I highly recommend it."

Cole Joplin, developer

JAVASCRIPT + CANVAS
If you are doing a lot of HTML5 Canvas drawing, you'll love WebCode.

Your drawings will be converted to JavaScript code instantly, saving you countless hours of boring work."
The power of the desktop. The freedom of the iPad.

iDraw is a powerful feature-packed vector design and illustration application available for both Mac OS X and iPad.
Inkscape is a professional vector graphics editor for Windows, Mac OS X, and Linux. It’s free and open source.

Inkscape is a powerful, free design tool. Whether you are an illustrator, designer, web designer or just someone who needs to create some vector imagery, Inkscape is for you!

Features
- What is Inkscape and how can you get it?
- Find out what Inkscape is capable of.

Overview

Gallery
Showcase of creations from the inkscape community.

Developers
Google Summer of Code 2014
- This is Inkscape’s 10th year mentoring in Google’s Summer of Code. Students, please see our Summer of Code wiki page for directions.
- Find an idea that interests you, and let us know!
- Join our thriving community

Recent News
Inkscape sponsoring LGM attendance
March 2, 2014
- Thanks to generous donations from users, the Inkscape Project is pleased to pay for five developers to attend the Libre Graphics Meeting, to be held in Leipzig in April. Tavmjong Bah will be presenting and is sponsored to go as Inkscape’s event coordinator; please contact him if you are an Inkscape developer and would like to be added to the sponsorship list.
- Tav will also be attending the colocated SVG Working Group Meeting.
Go snag some vector stuff off the internet.
Stock Vector Illustration:

Ireland four leaf lucky clovers background for Happy St. Patrick's Day. EPS 10.

Image ID: 124006018
Release Information: N/A
Copyright: Allies Interactive

Vector Format

This image is a vector illustration and can be scaled to any size without loss of resolution. This image will download as a .eps file. You will need a vector editor to use this file (such as Adobe Illustrator).
- Adjust colors
- Change artboard size
- Remove cruft
- Save it
How to Use SVG
LIKE, ON WEBPAGES YA TEAMSTERS
<img src="clover.svg" alt="clover">
HTML

SVG-as-img

```html
<img src="clover.png" alt="clover">
```
It's February 1st today, which I've decided to declare International box-sizing Awareness Day. In honor of, you guessed it, the most humble and undersung, yet awesome and useful CSS property: box-sizing.

The date corresponds to Paul Irish's post where he introduced the concept of using it on every single element on the page. We've talked about it around here a few times as well.

awesome and useful CSS property: box-sizing.

The date corresponds to Paul Irish's post where he introduced the concept of using it on every single element on the page. We've talked about it around here a few times as well.
CSS

SVG as background-image

clover {
  background: url(clover.svg);
}

Great way to support gradients in IE 9. Compass can automatically do that for you.

www.sitepoint.com/getting-started-svg-gradients/
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Corporis, tenetur, commodi illo neque aspersatur placeat itaque autem veritas. Adipisci, impedit, omnis laboriosam modi ratione consequuntur commodi velit provident debitis eius?
.htaccess

content-type fix

AddType image/svg+xml .svg .svgz
Because SVG **can** be resized worry-free, you **probably will** resize them most of the time.

http://blog.froont.com/9-basic-principles-of-responsive-web-design/
CSS

```css
img {
  max-width: 100%;
  height: auto;
}

img[src$=".svg"] {
  max-width: auto;
  width: 100%;
  height: auto;
}
```

Don’t get bigger than the container element.
Also, *don’t scale up*.

Don’t get bigger than the container element.
*Scale up if you need to.*
Fill the entire background of this element no matter what. Scale up if you have to.

Firefox used to have an issue with blurring SVG when resizing, but not anymore since v24.
Inline SVG

You can put `<svg>` right up in your `<html>`, homegirl.
1<h1>Arrow Image</h1>

3<img src="https://s3-us-west-2.amazonaws.com/s.cdpn.io/3/arrow.svg"/>
Three Big Advantages to Using Inline SVG:

1) Shapes are in the DOM: CSS Styling! JavaScript stuff!
2) No additional HTTP Request (it’s like a data URI, but better)
3) You can use define once and use in multiple places
SVG Templating

Using inline `<svg>` as an icon system
Sites need a system for icons.
<svg>
  <path d="M80.893,40.234c- ...
  <path d="M6-0.076-0.02 ...
  <path d="M16-0.36-1.09 ...
</svg>
<svg>
  <defs>
    <g id="shape-icon-1">
      <path d="M80.893,40.234c- ..."
    </g>

    <g id="shape-icon-2">
      <path d="M6-0.076-0.02 ..."
      <path d="M16-0.36-1.09 ..."
    </g>
  </defs>
</svg>
<body>
<svg>
<defs>
  <g id="shape-icon-1">
    <path d="M80.893,40.234c- ...
  </g>

  <g id="shape-icon-2">
    <path d="M6-0.076-0.02 ...
    <path d="M16-0.36-1.09 ...
  </g>

</defs>
</svg>
<body>

<?php include "svg-defs.php"; ?>
<svg viewBox="0 0 100 25" class="shape-tab">
  <use xlink:href="#shape-icon-1"></use>
</svg>
<nav class="main-nav" id="main-nav">

<a href="/" class="home">
    <svg viewBox="0 0 100 25" class="shape-tab">
        <use xlink:href="#shape-tab"></use>
    </svg>
    <span>Blog</span>
</a>

<a href="/video-screencasts/" class="videos">
    <svg viewBox="0 0 100 25" class="shape-tab">
        <use xlink:href="#shape-tab"></use>
    </svg>
    <span>Videos</span>
</a>
</nav>
On CodeKit 2.0

The following is a guest post by Bryan Jones, the creator of CodeKit. I’ve been using CodeKit for a couple of years now and I’ve talked about it plenty. In my opinion it changed the game in front end development making it easy to use advanced tools that, while useful for developers, are frequently hard to grasp and understand. It’s a great tool for designers to try and master and it will definitely be a part of my future workflow.
is loaded with real-world-y examples....

Read Article →

More Blog Posts →

SOCIAL

Twitter
Twitter
Pushpin
Facebook
NBA
Google+
YouTube

PROJECTS

CodePen is a social code playground for web designers and developers. Build demos, find inspiration, or troubleshoot code. With CodePen PRO you can teach students, pair program, host files, and more!

ShopTalk is a podcast about all things web design and development hosted by Dave Rupert and me. The show is recorded live and covers the week in #hotdrama and listener Q&A.

LINKAGE
May 27th, 2014
Amelia Island, Florida

"The first day of CSSConf was even nerdier than the first day of JSConf."
—CSSConf 2013 attendee

CSSConf is where developers come together to talk about building the world's most engaging user interfaces. The presenters push the boundaries of what is possible — talking about the latest technologies, cutting edge techniques, and tools. We look forward to seeing you on Amelia Island!

2014.cssconf.com

codepen.io/elrumordelaluz/pen/dhwLi
Even better, let’s get that block of SVG `<defs>` out of the HTML entirely.
<svg>
  <defs>
    <g id="shape-icon-1">
      <path d="M80.893,40.234c- ...
    </g>

    <g id="shape-icon-2">
      <path d="M6-0.076-0.02 ...
      <path d="M16-0.36-1.09 ...
    </g>
  </defs>
</svg>
browser caching!
IE (any version) won’t allow `<use>` with an external source.

Jonathan Neal has a fixer-upper script. (XHRs in the guts so it’s like straight up inline SVG.)

SVG for Everybody

Use external SVG spritemaps today. **SVG for Everybody** minds the gap between SVG-capable browsers and those which do not support external SVG spritemaps.

To use `svg4everybody`, add it in the `<head>` of your document.

```html
<script src="/path/to/svg4everybody.ie8.min.js"></script>
```
Using `<symbol>` instead of `<g>` is even **better**!
<svg xmlns="http://www.w3.org/2000/svg">
  <defs>
    <symbol id="icon" viewBox="0 0 100 100">
      <desc>Icon of a clamdigger.</desc>
      <path d="M6-0.076-0.02 ..." />
      <path d="M16-0.36-1.09 ..." />
    </symbol>
  </defs>
</svg>

<svg class="icon">
  <use xlink:href="#icon" />
</svg>

Or, do add viewBox if you want/need
width: auto;
behavior.
Fragment Identifiers

```html
.specific-icon {
    background: url('svg.svg#svgView(viewBox(0,0,48,41))');
}

<svg
    <view id='monkey-view' viewPort='...' />
</svg>

<img src='faces.svg#monkey-view' alt=''>
```
This is the idea of SVG Sprites
It’s the same idea, only **way better** because:

1) Vector (awesome)
2) No `background-position` craziness
3) More CSS control (i.e. SVG-specific, `@media`, etc)
4) All the other advantages of inline SVG (accessible, multiple links, scriptable)
5) Fallbacks are possible
Create multi-color themes with CSS to match your brand

Want to style every icon green, or match all seven of your brand colors? We have sample themes with variables in SCSS & LESS to get you going.

Select an example theme

- Multi-color
- Multi-tone
- Contained
```html
div class="logo">
  svg class="icon-blend">
    use xlink:href="#blend-logo"></use>
  </svg>
</div>
```
```
<svg class="icon-blend">
  <use xlink:href="#blend-logo"></use>
</svg>
```

```
.logo {
  height: 0;
  padding-top: 100%;
  position: relative;
  max-width: 300px;
  margin: 0 auto;
}

.icon-blend {
  position: absolute;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  max-height: 396px;
}
```
HTML
1. [[http://codepen.io/chriscoyier/pen/c742377ce1a62237dda89aef5b1c021]]
2. <div class="logo">
3.  <svg class="icon-blend">
4.   <use xlink:href="#blend-logo"></use>
5.  </svg>
6. </div>

CSS
1. .icon-blend {
2.   filter: grayscale(1);
3. }

JS
1. ...
```html
<div class="logo">
<svg class="icon-blend">
<use xlink:href="#blend-logo"></use>
</svg>
</div>
```

```css
#blender-top {
    transform: rotate(-4deg);
    transform-origin: bottom left;
}
```
HTML

```html
  <div class="logo">
    <svg class="icon-blend">
      <use xlink:href="#blend-logo"></use>
    </svg>
  </div>
```

CSS

```css
@media (max-width: 300px) {
  #button {
    display: none;
  }
  #light-1, #light-2 {
    display: none;
  }
}
```

JS

```javascript

```
HTML

```html
[[[http://codepen.io/chriscoyier/pen/c742377ce1a62237dda09aef5ba1c021]]

```  

```html
<div class="logo">
  <svg class="icon-blend">
    <use xlink:href="#blend-logo"></use>
  </svg>
</div>
```  

CSS

```css
@media (max-width: 300px) {
  #button {
    display: none;
  }
  #light-1, #light-2 {
    display: none;
  }
}

@media (max-width: 200px) {
  #goop-1, #goop-2, #goop-3 {
    fill: green;
  }
}
```
```html
1. <div class="logo">
2.   <svg class="icon-blend">
3.     <use xlink:href="#blend-logo"></use>
4.   </svg>
5. </div>
```
Responsive Icons

Media Queries in CSS + SVG
All this stuff is possible on inline SVG.

You get some advantages of SVG with `<img>` or `background-image`, but not as many.
SVG vs. Icon Fonts
Cagematch!
THE ANCIENT PAST

Icon Fonts are Awesome

1. Because you can easily change the size
2. Because you can easily change the color
3. Because you can easily shadow their shape
4. Because they can have transparent knockouts, which work in IEs6 unlike alpha transparent pngs.
5. Because you can do all the other stuff image based icons can do, like change opacity or rotate or whatever.
6. You'll be able to do things like add strokes to them with text-stroke or add gradients/textures with background-clip: text; once browser support is a bit deeper.

![Icon Fonts Example](https://css-tricks.com/examples/IconFont/)

```
.data-icon="&#x21db;" data-icon="&#x25c9;" data-icon="&#x25e8;" data-icon="&#x25ce;" data-icon="&#x002b;" data-icon="&#x2600;"
```
<table>
<thead>
<tr>
<th></th>
<th>Icon Font</th>
<th>SVG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vector</td>
<td>yup (but anti-aliasing...)</td>
<td>yup</td>
</tr>
<tr>
<td>CSS Control</td>
<td>some (single color)</td>
<td>better</td>
</tr>
<tr>
<td>Positioning</td>
<td>die XXX murder</td>
<td>good</td>
</tr>
<tr>
<td>Weird Failures*</td>
<td>yup</td>
<td>less</td>
</tr>
<tr>
<td>Semantics</td>
<td>none <code>&lt;span&gt;</code> / <code>::before</code></td>
<td>better <code>&lt;svg&gt;</code></td>
</tr>
<tr>
<td>Accessibility</td>
<td>be very careful</td>
<td>better, <code>&lt;title&gt;</code>&lt;desc&gt;</td>
</tr>
<tr>
<td>Ease of use</td>
<td>meh</td>
<td>better</td>
</tr>
<tr>
<td>Browser Support</td>
<td>IE 6+</td>
<td>IE 9+</td>
</tr>
</tbody>
</table>

Weird Failures: CORS/Fx + more?, CDN/network fail, Chrome non-active tab, no @font-face support (Opera Mini, Nokia XPress, Blackberry 4&5, Android 2.1, Windows Phone 7, 7.8, 8?, more?)
Guess which one I like?
Tip: it’s the one that isn’t icon fonts.
If you're going to use icon fonts, read this.

It goes into how it can be done perfectly* in regards to accessibility, browser support, and failures.

*95%
Build Tools

Don’t freak out.
grunt-svgsstore

Merge SVGs from a folder.

Why?
Because Chris Coyer asked.

Getting Started
This plugin requires Grunt as well as install and use Grunt plugins. Once you’re familiar with that process:

gem install grunt-svgsstore --user-dev

Once the plugin has been installed, it may be enabled inside your Gruntfile with this line:

    grunt.loadNpmTasks('grunt-svgsstore');

gulp-svgsprites

Gulp plugin for working with SVGs

Notice: This is a new project with a deliberately minimal feature set. If there’s something missing that you’d like to see, feel free to contribute!
Drag & Drop ur SVGs on the Grumpicon plz.

Download it!

8 Grumpicons

- clubs
  - 1146 bytes
- droplet
  - 1286 bytes
- headphones
  - 1663 bytes
- home
  - 524 bytes
- icon_1975
  - 1158 bytes
- icon_64846
  - 2333 bytes
Optimizing SVG

1. Removing meta junk
2. Simplifying shapes
3. Reducing crazy precision
4. Removing hidden elements
5. Removing whitespace

Kyle Foster has great videos detailing SVG optimization

youtube.com/watch?v=iVzW3XuOm7E&feature=youtu.be
youtube.com/watch?v=1AdX8odLC8M&feature=youtu.be
Raster images are a grid of pixels so it’s easy to understand they have an intrinsic width and height.
SVG has a **Viewport** and **viewBox**

**Viewport** is the size of the element. Think how the browser window is the viewport for the whole site but an iframe has its own smaller viewport. Likewise, SVG has its own viewport.

**viewBox** is an attribute on `<svg>`. It is the “canvas” on which SVG instructions are carried out. It is essentially a nexus with an aspect ratio (don’t think of them as pixels).
<svg class="buddy" viewBox="0 0 300 200">
  .buddy {
    width: 400px;
    height: 200px;
  }
</svg>
<svg class="buddy" viewBox="0 0 300 200">
  .buddy {
    width: 200px;
    height: 200px;
  }
</svg>
<svg class="buddy" viewBox="0 0 300 200" preserveAspectRatio="xMinYMax">
  <rect width="200px" height="200px"/>
  .buddy {
    width: 200px;
    height: 200px;
  }
</svg>
Dealing with Fallbacks

e.g. dealing with IE 8 and Android 2.3
Remember, one kind of fallback is **nothing**.

Like if you’re using an icon to enhance a word.
SVGeezy

A simple JavaScript plugin which detects SVG images on your website, and automatically "looks" for a standard image fallback for those older, less capable browsers. Those browsers need lovin' too dawg.

Download from GitHub

or

Share this on Twitter
<img src="images/horse.svg" alt="horse">

<script src="/js/libs/svgeezy.js"></script>
<script>
    svgeezy.init(false, "png");
</script>

</body>
<img src="images/horse.png" alt="horse">

<script src="/js/libs/svgeezy.js"></script>
<script>
  svgeezy.init(false, 'png');
</script>

</body>
It works, but it requests two images in non-supporting browsers.
... but, SVG is more well supported than the responsive images stuff is.
Picturefill
A responsive image polyfill

The picture element and associated features are W3C standard HTML features that allow web developers to deliver an appropriate image to every user depending on a variety of conditions like screen size, viewport size, screen resolution, and more. Picturefill is a JavaScript file (or a polyfill to be more specific) that enables support for the picture element and associated features in browsers that do not yet support them, so you can start using them today!

Picturefill is developed and maintained by filament group, with help from the members of the Responsive Images Community Group and the developer community at large. For Picturefill version 2, a special thanks goes to Shawn Jansepar for all his work on polyfilling the new spec.

Contributing, Bug Reports, and More information

For more information on the development of Picturefill, and how you can file bugs or contribute fixes, check out the the code project site on Github: Picturefill on Github.

Downloading Picturefill

Picturefill Version 2.0.0

For native picture element support, you’ll want Picturefill version 2. These downloads include the matchMedia polyfill for browsers that need it (like IE9).
Fallback for CSS background-image

This works because it combines multiple backgrounds and old-syntax gradients, thus it works for Android 2.x where multiple backgrounds alone did not. You could use Modernizr too, but if you just need it for this, it would be overkill.

[Ref: pauginer.tumblr.com/post/36614680636/invisible-gradient-technique]
Also, **background** is generally decorative and non-vital to content, so no fallback is often acceptable.
SVG for Everybody

Use external SVG spritemaps today. SVG for Everybody minds the gap between SVG-capable browsers and those which do not support external SVG spritemaps.

To use svg4everybody, add it in the <head> of your document.

```html
<script src="path/to/svg4everybody.ie8.min.js"></script>
```

Only IE6-8 require the script run this early, in order to shiv the svg and use elements.

Usage

`<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 64 64">
  <g id="codepen"></g>
  <g id="youtube"></g>
  <g id="twitter"></g>
</svg>`

The preceding spritemap may be referenced without assistance in Chrome, Firefox, and Opera. This script polyfills the experience in IE9-11.

```html
<svg role="img" title="CodePen" viewBox="0 0 64 64">xuse xlink:href="spriteMap.svg#codepen"
</use>
<svg role="img" title="YouTube" viewBox="0 0 64 64">xuse xlink:href="spriteMap.svg#youtube"
</use>
<svg role="img" title="Twitter" viewBox="0 0 64 64">xuse xlink:href="spriteMap.svg#twitter"
</use>
</svg>
```

Supporting (most) browsers

Just leaves alone

IE 9-11

Ajax’s in `<g>`

IE 6-8

 `<img src="icon.png"`
Animating SVG

Let us count the ways.
1. SVG Syntax `<animate>` (“SMIL”)
2. With CSS `@keyframes` / transitions
3. With JavaScript
One first rule is the morph can be made only between certain amount of points, I choose 22 thinking it's enough to get a decent detail level. In case of a less complicated number, like 'i' for example, I can stack points together or in line.
<button id="button">

<svg viewBox="0 0 194.6 185.1">

<polygon fill="#FFD41D" points="97.3,0 127.4,60.9 ...">

<animate id="animation-to-check" begin="indefinite" fill="freeze" attributeName="points" dur="500ms" to="110,58.2 147.3,0 192.1,29 ...">

</svg>

<span id="button-text">Save</span>

</button>
For the colorful.

This is an experiment using only CSS to animate SVG. The idea is to give the impression of elements being ‘painted’ onto the canvas without using JavaScript. The animation is achieved by creating and animating SVG clipping masks. For the full experience, please view in Chrome or Safari. To find out more, check out the tutorial.

Learn more about SVG clipping masks>

Share this demo Facebook Twitter Google+
How do you feel about our customer service?

**Could have been better**
SVG Filters & Blend Modes

The OG of cool web effects.
<svg width="0" height="0">
  <filter id="blur">
    <feGaussianBlur stdDeviation="5" />
  </filter>
</svg>

<div class="an-event-apart"></div>

.an-event-apart {
  -webkit-filter: url(#blur);
  filter: url(#blur);
}
SVG Filters

You can use CSS filters on SVG, but SVG has its own special filters too.
Accessible SVG

Thankfully it ain’t that hard.
Tips for Creating Accessible SVG

Published May 6, 2014

Scalable Vector Graphics (SVG) exists in a quantum state of accessibility. Which is to say that SVG is both accessible and inaccessible at the same time.

Happily, there is no SVG uncertainty principle.

We know which aspects of SVG are accessible to whom, which are not, and what we can do to improve both.
1. Use SVG
2. Use Inline SVG
3. Provide a `<title>` and aria-labeledby
4. Provide a `<desc>` and aria-labeledby
5. Use `<text>`
6. Make links focusable (if parts)
7. Use ARIA roles
8. Create alternatives
A light green rectangle with rounded corners and a dark green border.
A Compendium of SVG Information

Published July 22, 2014 by Chris Coyler

A huge pile of information about SVG.

How to Use SVG

These are overview articles covering lots of stuff relating to SVG. Why to use it and the basics of how to use it. Mostly: `<img>`, `background-image`, `<svg>`, `<iframe>`, and `<object> / <embed>`. 

Howdy, Chris!
Thank you!

An Event Apart

@chriscoyier