BRAD MARSHALL BECOME AN SVG SUPERHERO



ABOUT BRAD

- Full stack dev from Kingston
- Lover of design, eater of code
- Currently building websites at 14 Theories
- Previously UI/UX, front-end for retail point of sale apps

TODAY'S TALK

- SVG fundamentals (how to code it, how to draw it)
- Adding SVG to your projects, scaling with viewBox
- Optimizing SVG (how to prepare it for the browser)
- Styling SVG with CSS
- Building an SVG Spritesheet, developing an icon system

SCALABLE VECTOR GRAPHICS

WHAT IS SVG?

- Vector graphics technology for the web
- A way of representing artwork/images using code
- A W3C Recommendation
- An HTML-like markup language (90% HTML / 10% NOT HTML)
- A file format (.svg, image/svg+xml)

WHAT SVG ISN'T

- NOT a browser plugin (100% standards-based native tech)
- SVG)

NOT a framework or library (although frameworks exist to HELP build better

HISTORY AND FUN FACTS

- Suuuuuper old (SVG 1.0 Spec published 2001)
- Current spec 1.1, 2.0 is on the way BUT...
- No longer niche, now supported in all browsers (IE9 and up)
- Enjoying a renaissance (death of flash, rise of mobile)





BITMAP

- AKA Raster images
- Made up of pixels
- A giant index of colour values
- Not scalable (scaling up loses image quality, scaling down wastes storage, bandwidth, etc.)
- PNG, JPEG, GIF (also TIFF, BMP, WEBP, HEIF)





VECTORS

- Made up of shapes and points
- Lines and curves connect the points mathematically
- INFINITELY scalable (always looks the same, always the same file size no matter what)
- ► SVG, EPS, AI



BITMAP

- Generally bigger filesize
- Files get exponentially large as artwork size increases
- Great for very complex artwork and photos (lots of shading, details or photorealistic art)
- Bitmap to vector conversions: HARD

VECTOR

- Generally smaller filesize
- Filesize does not change as artwork size increases
- Great for simple artwork and computer-generated illustrations (icons, logos, line drawings, maps)
- Vector to bitmap conversions: EASY



HOW TO CHOOSE

- Use vector for:
 - simple computer-generated graphics
 - when vector has a smaller filesize
 - when you need to scale to multiple sizes
- Use bitmaps for:
 - photos or realistic graphics
 - when a bitmap has a smaller filesize
 - when you only need one size and that size is small

WHEN IN DOUBT, SAVE BOTH AND COMPARE

(BUT SAVE YOUR ORIGINALS!)



THE OUTERMOST CONTAINER





NAMESPACES

<svg xmlns="http://www.w3.org/2000/svg"</pre>

xmlns:xlink="http://www.w3.org/1999/xlink"></svg>

VIEWBOX AND PRESERVEASPECTRATIO

<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" viewBox="0 0 400 400" preserveAspectRatio="xMinYMin meet"></svg>

SHAPE ELEMENTS



<rect>

















ATTRIBUTES

- <rect x="10" y="10" height="50" width="50" rx="5" ry="5" />
- <circle cx="600" cy="200" r="100" />
- <ellipse cx="300" cy="300" rx="250" ry="100" />
- x1="500" y1="300" x2="700" y2="100" />
- <polygon points="0 0, 100 100, 0 100" />
- <polyline points="50,375 150,375 150,325 250,325 250,375</pre> 350,375 350,250 450,250 450,375 550,375 550,175 650,175 650,375 750,375 750,100 850,100 850,375 950,375 950,25 1050,25 1050,375 1150,375" />
- 2.9,213.1,6.7z" />

<path d="M213.1,6.7c-32.4-14.4-73.7,0-88.1,30.6C110.6,4.9,67.5</pre> -9.5,36.9,6.7C2.8,22.9-13.4,62.4,13.5,110.9C33.3,145.1,67.5,17 0.3,125,217c59.3-46.7,93.5-71.9,111.5-106.1C263.4,64.2,247.2,2

GROUP ELEMENT (<G>)

```
<circle cx="256.004" cy="256.004" r="246.855" fill="#F7B239"/>
<path d="..." fill="#E09B2D"/>
<g class="eyes" fill="#F95428">
    <path d="..."/>
    <path d="..."/>
</g>
<g class="mouth">
    <path d="..." fill="#FFF"/>
    <path d="..." fill="#A81004"/>
    <path d="..." fill="#F95428"/>
</g>
<g class="lines">
    <path d="..."/>
    <path d="..."/>
    <path d="..."/>
</g>
```



TEXT ELEMENT

<text x="20" y="40">My wicked awesome text!</text>

No Line wrapping available... sorry! Use multiple <text> elements instead.

<text x="20" y="40">My wicked</text> <text x="15" y="50">awesome text!</text>

OTHER SVG ELEMENTS

<defs></defs> <clipPath></clipPath> <mask></mask> <image></image> <use></use> <symbol></symbol>

Full list:

https://developer.mozilla.org/en-US/docs/Web/SVG/Element

linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></linearGradient></li <radialGradient></radialGradient>

TIPS

- Co-ordinate system is based on top-left
- CApItalIzAtiON iS ImPoRTaNT
- Open ended elements MUST be closed with /
- top). No z-index!

Stacking order is controlled by order of elements in DOM (last elements on





SVG AS SRC

SVG AS BACKGROUND-IMAGE

<div class="my-image"></div>

.my-image {
 background-image: url(image.svg);
 background-size: 200px 300px; /* or contain, cover, etc. */
}

INLINE SVG

<!DOCTYPE html> <html> <head> <title></title> </head> <body> <!-- html here --> <svg>...</svg>

</body>
</html>

<meta charset="utf-8">

<!-- more html here -->

OTHER METHODS

Object element

</object>

Embed element

Iframe element

<iframe src="image.svg"> </iframe>

```
<object type="image/svg+xml" data="image.svg">
 Your browser does not support SVG
```

```
<embed type="image/svg+xml" src="image.svg" />
```

```
Your browser does not support iframes
```

EXTERNAL VS INLINE SVG PROS & CONS

External SVG:

external file).

Inline SVG:

Pro: You can independently target and style each individual SVG element!

Con: Bloats your markup, somewhat more complicated

Pro: Quicker, more convenient to place (just like placing a bitmap image) Con: Much more limited for styling (All styling/scripting must be done IN the



DATA URIS

As img src

<img src='data:image/svg+xml,<svg xmlns="http://www.w3.org/2000/svg" width="1000" height="500"><rect width="1000" height="500" fill="#f00"/><rect width="500" height="500" fill="#fff" x="250"/><path fill="#ff0000" d="m 499.99228,46.875 -34.11312,63.62529 c -3.87086,6.91501 -10.80627,6.27363 -17.74169,2.41138 l -24.69699,-12.78858 18.40705,97.72711 c 3.87086,17.85419 -8.54859,17.85419 -14.67765,10.13435 l -43.10105,-48.25099 -6.99738,24.503 c -0.80692,3.21777 -4.35481,6.59744 -9.67748,5.79261 l -54.50177,-11.45912 14.31524,52.04475 c 3.06451,11.58054 5.4549,16.37528 -3.09375,19.42959 l -19.42619,9.13025 93.82127,76.20838 c 3.7135,2.88151 5.58971,8.067 4.26768,12.7621 l -8.21136,26.94707 c 32.30405,-3.72371 61.24898,-9.32594 93.56939,-12.77619 2.85323,-0.30459 7.62988,4.40408 7.61029,7.71058 l -4.28024,98.72342 15.70639,0 -2.47237,-98.5117 c -0.0197,-3.3065 4.31372,-8.22689 7.16695,-7.9223 32.32041,3.45026 61.26538,9.05248 93.56942,12.77619 l -8.21134,-26.94707 c -1.32203,-4.6951 0.55417,-9.88059 4.26767,-12.7621 l 93.82125,-76.20838 -19.42617,-9.13025 c -8.54867,-3.05431 -6.15828,-7.84905 -3.09377,-19.42959 l 14.31527,-52.04475 -54.5018,11.45912 c -5.32267,0.80483 -8.87056,-2.57484 -9.6775,-5.79261 l -6.99737,-24.503 -43.10103,48.25099 c -6.12908,7.71984 -18.54854,7.71984 -14.67768,-10.13435 l 18.40702,-97.72711 -24.69694,12.78858 c -6.93559,3.86225 -13.87083,4.50363 -17.7417,-2.41138"/>

DATA URIS

As background-image

<div class="bgImage"></div>

```
div.bgImage {
    background-image: url('data:image/svg+xml;utf8,<svg xmlns="http://www.w3.org/2000/svg" width="1000" height="500"><
        rect width="1000" height="500" fill="#f00"/><rect width="500" height="500" fill="#fff" x="250"/><path
        fill="#ff0000" d="m 499.99228,46.875 -34.11312,63.62529 c -3.87086,6.91501 -10.80627,6.27363
        -17.74169,2.41138 l -24.69699,-12.78858 18.40705,97.72711 c 3.87086,17.85419 -8.54859,17.85419
        -14.67765,10.13435 l -43.10105,-48.25099 -6.99738,24.503 c -0.80692,3.21777 -4.35481,6.59744 -9.67748,5.79261
        l -54.50177, -11.45912 14.31524, 52.04475 c 3.06451, 11.58054 5.4549, 16.37528 -3.09375, 19.42959 l
        -19.42619,9.13025 93.82127,76.20838 c 3.7135,2.88151 5.58971,8.067 4.26768,12.7621 l -8.21136,26.94707 c
        32.30405, -3.72371 61.24898, -9.32594 93.56939, -12.77619 2.85323, -0.30459 7.62988, 4.40408 7.61029, 7.71058 l
        -4.28024,98.72342 15.70639,0 -2.47237,-98.5117 c -0.0197,-3.3065 4.31372,-8.22689 7.16695,-7.9223
        32.32041,3.45026 61.26538,9.05248 93.56942,12.77619 l -8.21134,-26.94707 c -1.32203,-4.6951 0.55417,-9.88059
        4.26767, -12.7621 l 93.82125, -76.20838 -19.42617, -9.13025 c -8.54867, -3.05431 -6.15828, -7.84905
        -3.09377,-19.42959 l 14.31527,-52.04475 -54.5018,11.45912 c -5.32267,0.80483 -8.87056,-2.57484
        -9.6775,-5.79261 l -6.99737,-24.503 -43.10103,48.25099 c -6.12908,7.71984 -18.54854,7.71984
        -14.67768,-10.13435 l 18.40702,-97.72711 -24.69694,12.78858 c -6.93559,3.86225 -13.87083,4.50363
        -17.7417,-2.41138"/></svg>');
    background-size: contain;
```

DATA URIS

Convenient - no need to serve file separately (same as inline SVG)

Only counts as one DOM node!

SHOULD I BASE64? BASE64'ING SVG DATA URIS IS CONSIDERED AN ANTI-PATTERN

MORKING WITH



THE SVG VIEWPORT

- Essentially, the <svg></svg> container element
- The "canvas" or drawing area for your artwork
- It clips your artwork / artwork is "cropped" to the viewport
- The "window" through which you can see your artwork
- Just like viewing a page in a browser window!

THE VIEWBOX

- An attribute on the <svg></svg> element that defines an imaginary box
- The box is "drawn" your artwork (but you never see it)
- The viewport automatically scales your artwork TO the viewbox.
- OR it tries to make the viewbox match the size of the viewport by scaling the artwork however necessary

WORKING WITH VIEWBOX

EXAMPLE







VIEWBOX SYNTAX

<svg viewBox="{min-x} {min-y} {width} {height}"></svg>

<svg viewBox="100 100 300 300"></svg> <!-- No units! -->

MIND YOUR CAPITALIZATIONS

PRESERVEASPECTRATIO

- For when your VIEWPORT isn't the same aspect ratio as your VIEWBOX
- Pretty much identical to background-position and background-size in CSS

ame aspect ratio as your VIEWBOX 4-position and background-size in CSS

PRESERVEASPECTRATIO SYNTAX

xMinYMin, xMidYMin, xMaxYMin

xMinYMid, xMidYMid, xMaxYMid

xMinYMax, xMidYMax, xMaxYMax

MIND YOUR CAPITALIZATIONS

preserveAspectRatio="{align} {meetOrSlice}"

meet

slice

ALIGN VALUE



MEETORSLICE VALUE







slice





4 WAYS TO STYLE

- Inline Styles (on SVG elements)
- Embedded Styles (inside SVG block)
- External Styles (in an external stylesheet)
- Presentation Attributes (on SVG elements)



NEW PROPERTIES

path.myShape {

/* Fill properties */
fill: #e2e2e2;
fill-rule: nonzero;
fill-opacity: .8;

/* Stroke properties */

stroke: #bbaacc; stroke-dasharray: 2, 4; stroke-dashoffset: 5px; stroke-linecap: round; stroke-linejoin: miter; stroke-miterlimit: 3; stroke-opacity: .5; stroke-width: 3px;

}

/* Misc. - Applies to various SVG elements */

```
color-rendering: optimizeSpeed;
shape-rendering: crispEdges;
text-rendering: optimizeLegibility;
image-rendering: optimizeQuality;
```

```
/* SVG Markers */
```

```
line.arrow {
    marker: url(#markerCircle);
    marker-start: url(#markerCircle);
    marker-mid: url(#markerArrowSmall);
    marker-end: url(#markerArrowLarge);
}
```

```
<marker id="markerCircle" markerWidth="8" markerHeight="8" refX="5" refY="5">
        <circle cx="5" cy="5" r="3"/>
</marker>
```

More complete list:

https://www.w3.org/TR/SVG/propidx.html

~I

l

EXPORTING AND **OPTIMIZING SVG**



SPRITESHE SYSTEMS



BOOKS

- SVG Essentials by J. Eisenberg & Amelia Bellamy-Royds (<u>http://</u> shop.oreilly.com/product/0636920032335.do)

Practical SVG by Chris Coyier (<u>https://abookapart.com/products/practical-svg</u>)



TUTORIALS

jenkov.com SVG Tutorial - http://tutorials.jenkov.com/svg/index.html

SVG Pocket Guide by Joni Trythall - <u>http://svgpocketguide.com/book/</u>

HOT HOT LINKS

- Main SVG Spec: <u>https://www.w3.org/TR/SVG11/</u>
- CSS Tricks articles on SVG: <u>https://css-tricks.com/?s=svg</u>
- <u>Web/SVG/Attribute/d</u>
- works/
- Hero Patterns: <u>http://www.heropatterns.com/</u>

Understanding the "d" attribute: <u>https://developer.mozilla.org/en-US/docs/</u>

How SVG Line Animation Works: <u>https://css-tricks.com/svg-line-animation-</u>

T S S

SLIDES, BLOG: <u>BMARSHALL.CA</u> @BMARSHALL ON CODEPEN SAY HI ON SLACK!

THANKS!