



Data Visualization for Web Applications

IT 7113 Data Visualization

J.G. Zheng

Spring 2018

<http://jackzheng.net/teaching/it7113/>

Data Visualization Web Solutions



- Static solutions - static image charts
 - Manually saved as image in many data visualization tools
 - produced by programming languages like Python (matplotlib), R (ggplot2),
- Plug-ins: Java Applets, Flash/Flex
- Basic web standards
 - HTML/CSS
 - HTML 5: Canvas and SVG
- JavaScript libraries
 - SVG based: Google Charts, D3, NVD3, FusionChart, Chartist
 - Canvas based: chart.js
- Web publishing/embeds from desktop tools
 - Tableau, Power BI
- Web mapping solutions
 - HTML map tag + image <https://www.weather.gov>
 - Google map

Image Charts



- Image charts are static images linked from the HTML using the `` tag
- Pros
 - Simple and easy
 - Compatible for all browsers
 - Supported by all CMS (many CMSES do not allow scripting)
- Cons
 - Static, no interactivity
- Google provides an API for image charts
 - <https://developers.google.com/chart/image/>
 - A simple API with parameters embedded in URL
 - Example:
<https://chart.googleapis.com/chart?cht=p3&chd=t:60,40&chs=250x100&chl=Hello|World>
 - Still alive but deprecated since 2012

HTML/CSS



- HTML and CSS can be used to make basic charts and light visualizations (especially conditional formatting)
 - Simple, and easy to use (especially for bar/columns based charts)
 - Easy to be manipulated and controlled
 - No JavaScript is needed
 - Easy to mix the visuals with other parts of the content (table, paragraph, etc.)
 - “work to a certain extent, and in many situations they're probably the safest bet.”
 - Lack of library support; have to do it from scratch with careful design and positioning.
- Live examples
 - <http://caniuse.com>
 - <https://scholar.google.com/citations?user=YJm9he0AAAAJ>
- More resources
 - <https://css-tricks.com/making-charts-with-css/>
 - <https://alistapart.com/article/accessibledatavisualization>
 - <https://designmodo.com/css-graph-chart-tutorials/>

HTML5 Drawing Solutions



- Before HTML5, plugins like Flash (and Java Applets) are the most common choices for interactive charts
 - Especially before IE 9.0
- Two drawing/graphics techniques in HTML5
 - SVG: Scalable Vector Graphics
 - Canvas
- Both are standards now and supported in all major browsers
 - <http://caniuse.com/#search=svg>
 - <http://caniuse.com/#search=canvas>

Canvas



- Canvas is an HTML element which provides a drawing context controlled by scripts
 - It is a resolution-dependent bitmap canvas, focusing on pixel manipulation
 - Relies on JavaScript for the drawing
 - Very fast (>30 fps); performance depends on the size of canvas, not the complexity of the graphics
 - Commonly used for rendering photos/videos, game graphics, art, or other high speed animations.
- Live examples
 - <http://corehtml5canvas.com/code-live/>
 - <https://davidwalsh.name/canvas-demos>
 - <http://www.chartjs.org>
- For a canvas tutorial
 - https://www.w3schools.com/graphics/canvas_intro.asp

Typical Canvas Usage



- Using canvas when
 - A large amount of objects or items needs to be visualized
 - Limited interactivity
 - Objects are irregular in shape or size
 - Complex and fast (real time) animation
 - Relatively small surface
- Typical usage
 - Weather map
 - Physics (e.g. light) simulation
 - Video processing
 - Gaming

SVG



- SVG is a language for describing vector graphics.
 - Vector graphics is the use of polygons (dot, line, polygon) to represent images; each polygon is described as a vector and mathematical formula.
 - Graphics scale to screen sizes and zoom levels
- As a standalone format or when mixed with other XML, it uses the XML syntax. When mixed with HTML5 (using the <svg> tag), it uses the HTML5 syntax.
- SVG elements (and styles, animations) can be declarative (just as HTML) or created using script
- SVG elements are added to the page DOM (a programmable DOM)
 - Each element can be manipulated using JavaScript directly without complexity.
 - Can be styled using CSS
 - It also has an event model
- Too many SVG elements may slow down the page performance
- For a SVG tutorial
 - https://www.w3schools.com/graphics/svg_intro.asp

Typical SVG Usage



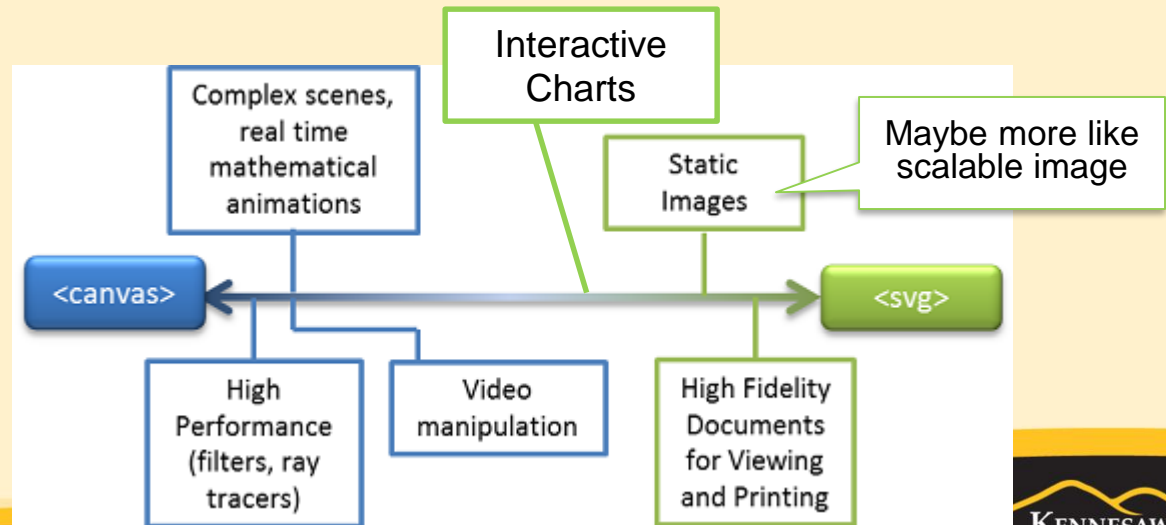
- Scalable image/animations
- Icons and logos
- Charts
- Slow speed games
- Examples:
 - <http://svg-wow.org/text-effects/text-effects.xhtml>
 - <http://svg-wow.org/camera/camera.xhtml>

High Level Summary of Canvas vs. SVG



Canvas	SVG
Pixel based (Dynamic .png)	Shape based
Single HTML element (immediate mode); canvas contents are not part of DOM	Multiple graphical elements, which become part of the DOM (retained mode)
Modified through script only	Modified through script and CSS
Event model/user interaction is granular: (x,y) coordinates based (more complex)	Event model/user interaction is abstracted (rect, path): object based (easier)
Performance is better with smaller surface, a larger number of objects (>10k), or both	Performance is better with smaller number of objects (<10k), a larger surface, or both

[https://msdn.microsoft.com/en-us/library/gg193983\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/gg193983(v=vs.85).aspx)



Personal Favor



- SVG is more natural for business data visualization and charts
 - These charts are usually pretty regular with limited number of objects
 - The interaction mode is easier to manipulate
 - More responsive in different screen sizes

Vanilla Coding vs. Using a Library



- Coding directly with CSS/Canvas/SVG is more complex and tedious, with lack of reusability and efficiency
- Writing SVG by hand can be slow and frustrating. Even for simple charts, it takes forever to write the code and to visually position each section. Much like the CSS-only solution to making graphs, your experience of making a chart with SVG by hand is likely to be a painful experience unless you're making something very small.
- Using libraries promotes reusability and efficiency at the expense of flexibility and control

JS Charting Libraries and APIs



- SVG
 - Google Charts
 - D3 and NVD3
 - High charts: <http://www.highcharts.com/blog/news/12-highcharts-goes-svg/>
 - <https://gionkunz.github.io/chartist-js/>
- Canvas
 - Chart.js <http://www.chartjs.org>
- More
 - https://en.wikipedia.org/wiki/JavaScript_graphics_library
 - <http://techslides.com/50-javascript-charting-and-graphics-libraries>

Google Charts



- <https://developers.google.com/chart/>
- A charting library provided by Google as a web API
- Free
- Uses SVG
- Gallery
 - <https://google-developers.appspot.com/chart/interactive/docs/gallery>

D3



- <https://d3js.org>
- D3.js is a JavaScript library for manipulating documents based on data.
- Uses SVG and CSS
- D3.js library uses pre-built JavaScript functions to select elements, create SVG objects, style them, or add transitions, dynamic effects or tooltips to them.
- Large datasets can be easily bound to SVG objects using simple D3.js functions to generate rich text/graphic charts and diagrams.
- D3 provides an underlying framework for general visualization; but it does not directly provide specific visual forms like charts.

D3 based Charting Library



- NVD3
 - <http://nvd3.org>
- C3.js
 - <http://c3js.org>
- An attempt to build re-usable charts and chart components for d3.js
- A library build upon D3 specifically to create charts

High Charts



- <http://www.highcharts.com>
- Uses SVG
- A large collection of chart types
- Free for non-commercial
- Cloud app
 - <https://cloud.highcharts.com/charts>
 - Create online and embed anywhere

Chart Library Selection



- Capability comparison
 - <http://socialcompare.com/en/comparison/javascript-graphs-and-charts-libraries>
 - https://en.wikipedia.org/wiki/Comparison_of_JavaScript_charting_frameworks
- Selection tips
 - <http://www.fusioncharts.com/javascript-charts-guide/>
 - <https://blog.zingchart.com/2015/09/15/how-to-choose-a-javascript-charting-library/>

Tableau Public



- Tableau visualizations can be published on the its server and then linked from or embedded in web pages
- Example
 - <http://tableaupicasso.com/bob-dylan-analyzing-55-years-of-lyrics/>
 - <http://icat.kennesaw.edu/untitled.php>

More Resources



- SVG vs. Canvas
 - http://jack-kelly.com/html5_canvas_versus_svg_for_interactive_charts_and_graphical
 - [https://msdn.microsoft.com/en-us/library/gg193983\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/gg193983(v=vs.85).aspx)
- Coding examples for charts
 - <http://codepen.io/collection/hsyFa/>
- JS charting libraries
 - https://en.wikipedia.org/wiki/JavaScript_graphics_library
 - <http://techslides.com/50-javascript-charting-and-graphics-libraries>
- Books
 - Interactive Data Visualization for the Web: An Introduction to Designing with D3, by Scott Murray, <https://www.amazon.com/dp/1491921285>
 - Using SVG with CSS3 and HTML5, <https://www.amazon.com/dp/1491921978/>
 - Data Visualization with Python and JavaScript, by Kyran Dale, <https://www.amazon.com/dp/1491920513/>