**Project 1:**

<https://maptimeboston.github.io/leaflet-intro/> ----

<http://www.ph.ucla.edu/epi/snow/snowcricketarticle.html> --doc

<https://kb.iu.edu/d/achr> ----pages

<http://bl.ocks.org/KatiRG/5f168b5c884b1f9c36a5> --- color bar graph for male and female.

<https://codebeautify.org/csv-to-tsv-converter#copy> ---- to convert csv to tsv files

<https://bl.ocks.org/DimsumPanda/689368252f55179e12185e13c5ed1fee---stacked> bar chart

How to use [d3-tip](https://github.com/caged/d3-tip) with d3's [simple bar chart example](http://bl.ocks.org/mbostock/3885304).—tooltip for bar chart

Epidemic—wide spread occurrence of an infectious disease in a community at a particular time..

<http://myweb.fsu.edu/aglerum/pages/data-viz-webliography.html>--graphs

TSV--TSV is a very efficient for Javascript/Perl/Python to process, without losing any typing information, and also easy for humans to read.

The choice depends on the application. In a nutshell, if your fields don't contain commas, use CSV; otherwise TSV is the way to go.

**TL;DR**

In both formats, the problem arises when the delimiter can appear within the fields, so it is necessary to indicate that the delimiter is not working as a field separator but as a value within the field, which can be somewhat painful.

For example, using CSV: Kalman, Rudolf, von Neumann, John, Gabor, Dennis

Some [basic approaches](https://softwareengineering.stackexchange.com/questions/26284/best-way-to-handle-delimited-files) are:

* Delete all the delimiters that appear within the field.

E.g. Kalman Rudolf, von Neumann John, Gabor Dennis

* Escape the character (usually pre-appending a backslash \).

E.g. Kalman\, Rudolf, von Neumann\, John, Gabor\, Dennis

* Enclose each field with other character (usually double quotes ").

E.g. "Kalman, Rudolf", "von Neumann, John", "Gabor, Dennis"

## [**CSV**](https://tools.ietf.org/html/rfc4180)

The fields are separated by a comma ,.

For example:

Name,Score,Country

Peter,156,GB

Piero,89,IT

Pedro,31415,ES

Advantages:

* It is more generic and useful when sharing with non-technical people, as most of software packages can read it without playing with the settings.

Disadvantages:

* Escaping the comma within the fields can be frustrating because not everybody follows the standards.
* All the extra escaping characters and quotes add weight to the final file size.

## [**TSV**](https://www.iana.org/assignments/media-types/text/tab-separated-values)

The fields are separated by a tabulation <TAB> or \t

For example:

Name<TAB>Score<TAB>Country

Peter<TAB>156<TAB>GB

Piero<TAB>89<TAB>IT

Pedro<TAB>31415<TAB>ES

Advantages:

* It is not necessary to escape the delimiter as it is not usual to have the tab-character within a field. Otherwise, it should be removed.

<https://gist.github.com/onderaltintas/6649521---website> for Mercator

<http://bl.ocks.org/sumbera/9972460--map> code

<http://bl.ocks.org/Sumbera/10463358---map> code--mer

<http://reference1.mapinfo.com/software/spectrum/lim/8_0/developerguide/Spatial/source/Development/devguide/rest/mapping/methods/rendermap.html> -----about rendermap.

<http://bl.ocks.org/Caged/6476579---> bar graph

geodata

**SVG**-- Scalable Vector Graphics--is a vector graphic format—based on XML and is **used to** display a variety of graphics on the Web and other environments. Under the hood, **SVG** documents are nothing more than simple plain text files that describe lines, curves, shapes, colors, and text.Jul 4, 2017

# What is SVG?

SVG stands for Scalable Vector Graphics. It is an XML language and file format, which allows you to code two-dimensional graphics that scale and can be manipulated via CSS or JavaScript.

[From the W3C on SVG](http://www.w3.org/Graphics/SVG/IG/resources/svgprimer.html):

*SVG is an XML language. This is important for at least three reasons. First, the code tends to adhere to agreed upon standards of how SVG should be written and how client software should respond. Second, like all XML, it is written in text, and can generally be read not only by machines but also by humans. Third, and perhaps most importantly, JavaScript can be used to manipulate both the objects and the Document Object Model, in ways quite similar to how JavaScript is used in conjunction with HTML. If you already know how to use JavaScript and HTML for web-programming, the learning curve will be pretty gentle, particularly in view of the benefits to be gained.*

SVG can be a static image, like a logo or illustration, or a complex animation. In this course we’ll talk mostly about the static version of SVG but we will touch on animation later on in the course.

All SVG graphics are plotted on a coordinate system of at least an x and y axis. When authoring SVG, we give the browser instructions on where to plot points on the coordinate system and connect them. By connecting the plotted points we can create shapes, lines, or paths.

Because SVG is vector-based, there isn’t any degradation or loss of fidelity when the graphics are scaled up. They are simply redrawn to accomodate the larger size. This makes this perfect for multi-context scenarios, like Responsive Web Design.

Your exposure to SVG is most likely from the web design and front-end development communities. I wrote the course with that in mind. We are going to approach SVG from the perspective of a web designer or web developer, presumably like yourself. In the web design world SVG is most commonly used to handle assets that we would otherwise have used bitmap files like JPEGs, PNGs, or GIFs.

The goal of this course is to familiarize you with SVG as a technology, allow you to use it immediately in your web design and develop and projects, and give you a solid foundation onto which you can build.

Throughout the course, I will link to relevant resources that will help you learn even more about SVG. Additionally, at the end there is also a lengthy list of resources for you to access. These resources are not only great for you to read, but they also document research that I did in creating the course.