D3 JavaScript Library Overview

# Name and Purpose

The name of the library is D3.js. **D3.js** is a JavaScript library for manipulating documents based on data. Since it is open source code, using it can help you gain access to the full capabilities of the modern browsers without becoming tied down to a proprietary framework.

# Link to their website

The full source and tests are available at <https://github.com/d3/d3>.

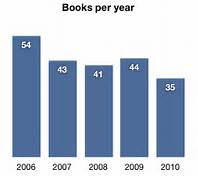
The main website can be found at <https://d3js.org/>.

# Main features

This extensive library can provide access to 30 main features.

* [Arrays](https://github.com/d3/d3/blob/master/API.md#arrays-d3-array) ([Statistics](https://github.com/d3/d3/blob/master/API.md#statistics), [Search](https://github.com/d3/d3/blob/master/API.md#search), [Transformations](https://github.com/d3/d3/blob/master/API.md#transformations), [Histograms](https://github.com/d3/d3/blob/master/API.md#histograms))
  + Array manipulation, ordering, searching, summarizing, etc.
* [Axes](https://github.com/d3/d3/blob/master/API.md#axes-d3-axis)
  + Human-readable reference marks for scales.
* [Brushes](https://github.com/d3/d3/blob/master/API.md#brushes-d3-brush)
  + Select a one- or two-dimensional region using the mouse or touch.
* [Chords](https://github.com/d3/d3/blob/master/API.md#chords-d3-chord)
  + Provides a wide range of ways to manipulate chords.
* [Collections](https://github.com/d3/d3/blob/master/API.md#collections-d3-collection) ([Objects](https://github.com/d3/d3/blob/master/API.md#objects), [Maps](https://github.com/d3/d3/blob/master/API.md#maps), [Sets](https://github.com/d3/d3/blob/master/API.md#sets), [Nests](https://github.com/d3/d3/blob/master/API.md#nests))
  + Handy data structures for elements keyed by string.
* [Colors](https://github.com/d3/d3/blob/master/API.md#colors-d3-color)
  + Color manipulation and color space conversion.
* [Dispatches](https://github.com/d3/d3/blob/master/API.md#dispatches-d3-dispatch)
  + Separate concerns using named callbacks.
* [Dragging](https://github.com/d3/d3/blob/master/API.md#dragging-d3-drag)
  + Drag and drop SVG, HTML or Canvas using mouse or touch input.
* [Delimiter-Separated Values](https://github.com/d3/d3/blob/master/API.md#delimiter-separated-values-d3-dsv)
  + Parse and format delimiter-separated values, most commonly CSV and TSV.
* [Easings](https://github.com/d3/d3/blob/master/API.md#easings-d3-ease)
  + Easing functions for smooth animation.
* [Forces](https://github.com/d3/d3/blob/master/API.md#forces-d3-force)
  + Force-directed graph layout using velocity Verlet integration.
* [Number Formats](https://github.com/d3/d3/blob/master/API.md#number-formats-d3-format)
  + Format numbers for human consumption.
* [Geographies](https://github.com/d3/d3/blob/master/API.md#geographies-d3-geo) ([Paths](https://github.com/d3/d3/blob/master/API.md#paths), [Projections](https://github.com/d3/d3/blob/master/API.md#projections), [Spherical Math](https://github.com/d3/d3/blob/master/API.md#spherical-math), [Spherical Shapes](https://github.com/d3/d3/blob/master/API.md#spherical-shapes), [Streams](https://github.com/d3/d3/blob/master/API.md#streams), [Transforms](https://github.com/d3/d3/blob/master/API.md#transforms))
  + Geographic projections, shapes and math.
* [Hierarchies](https://github.com/d3/d3/blob/master/API.md#hierarchies-d3-hierarchy)
  + Layout algorithms for visualizing hierarchical data.
* [Interpolators](https://github.com/d3/d3/blob/master/API.md#interpolators-d3-interpolate)
  + Layout algorithms for visualizing hierarchical data.
* [Paths](https://github.com/d3/d3/blob/master/API.md#paths-d3-path)
  + Serialize Canvas path commands to SVG.
* [Polygons](https://github.com/d3/d3/blob/master/API.md#polygons-d3-polygon)
  + Geometric operations for two-dimensional polygons.
* [Quadtrees](https://github.com/d3/d3/blob/master/API.md#quadtrees-d3-quadtree)
  + Two-dimensional recursive spatial subdivision.
* [Queues](https://github.com/d3/d3/blob/master/API.md#queues-d3-queue)
  + Evaluate asynchronous tasks with configurable concurrency.
* [Random Numbers](https://github.com/d3/d3/blob/master/API.md#random-numbers-d3-random)
  + Generate random numbers from various distributions.
* [Requests](https://github.com/d3/d3/blob/master/API.md#requests-d3-request)
  + A convenient alternative to asynchronous XMLHttpRequest.
* [Scales](https://github.com/d3/d3/blob/master/API.md#scales-d3-scale) ([Continuous](https://github.com/d3/d3/blob/master/API.md#continuous-scales), [Sequential](https://github.com/d3/d3/blob/master/API.md#sequential-scales), [Quantize](https://github.com/d3/d3/blob/master/API.md#quantize-scales), [Ordinal](https://github.com/d3/d3/blob/master/API.md#ordinal-scales))
  + Encodings that map abstract data to visual representation.
* [Selections](https://github.com/d3/d3/blob/master/API.md#selections-d3-selection) ([Selecting](https://github.com/d3/d3/blob/master/API.md#selecting-elements), [Modifying](https://github.com/d3/d3/blob/master/API.md#modifying-elements), [Data](https://github.com/d3/d3/blob/master/API.md#joining-data), [Events](https://github.com/d3/d3/blob/master/API.md#handling-events), [Control](https://github.com/d3/d3/blob/master/API.md#control-flow), [Local Variables](https://github.com/d3/d3/blob/master/API.md#local-variables), [Namespaces](https://github.com/d3/d3/blob/master/API.md#namespaces))
  + Transform the DOM by selecting elements and joining to data.
* [Shapes](https://github.com/d3/d3/blob/master/API.md#shapes-d3-shape) ([Arcs](https://github.com/d3/d3/blob/master/API.md#arcs), [Pies](https://github.com/d3/d3/blob/master/API.md#pies), [Lines](https://github.com/d3/d3/blob/master/API.md#lines), [Areas](https://github.com/d3/d3/blob/master/API.md#areas), [Curves](https://github.com/d3/d3/blob/master/API.md#curves), [Symbols](https://github.com/d3/d3/blob/master/API.md#symbols), [Stacks](https://github.com/d3/d3/blob/master/API.md#stacks))
  + Graphical primitives for visualization.
* [Time Formats](https://github.com/d3/d3/blob/master/API.md#time-formats-d3-time-format)
  + Parse and format times, inspired by strptime and strftime.
* [Time Intervals](https://github.com/d3/d3/blob/master/API.md#time-intervals-d3-time)
  + A calculator for humanity’s peculiar conventions of time.
* [Timers](https://github.com/d3/d3/blob/master/API.md#timers-d3-timer)
  + An efficient queue for managing thousands of concurrent animations.
* [Transitions](https://github.com/d3/d3/blob/master/API.md#transitions-d3-transition)
  + Animated transitions for [selections](https://github.com/d3/d3/blob/master/API.md#selections).
* [Voronoi Diagrams](https://github.com/d3/d3/blob/master/API.md#voronoi-diagrams-d3-voronoi)
  + Compute the Voronoi diagram of a given set of points.
* [Zooming](https://github.com/d3/d3/blob/master/API.md#zooming-d3-zoom)
  + Pan and zoom SVG, HTML or Canvas using mouse or touch input.

# An Example of how to use the library

[](https://www.bing.com/images/search?q=d3+bar+chart+examples&view=detailv2&qpvt=d3+bar+chart+examples&id=78EEC9602F452E6CB62420E90D4111A732CCFAC4&selectedIndex=4&ccid=HifbiM%2b1&simid=608034191984430396&thid=OIP.M1e27db88cfb5170593f02b9fe0171424o0)

To replicate this in D3, you will create a series of rectangles of the appropriate height along the x axis. D3's [linear scale](https://github.com/mbostock/d3/wiki/Quantitative-Scales#linear) will provide functions used to map a data point on the x and y axis.

To draw the rects, first use the [selectAll](https://github.com/mbostock/d3/wiki/Selections#d3_selectAll) method returns to create a selector, then use the [data](https://github.com/mbostock/d3/wiki/Selections#data) method to bind the array of data objects to the selection. Calling [enter](https://github.com/mbostock/d3/wiki/Selections#enter) on the selection will create any objects that don't have corresponding elements in the data array.

Since there's no existing rects returned by the selector, one will be created for each data element. Tthe selectAll/data/enter sequence can be used to add or remove elements from an existing list. The rest of the diagram all comes down to positioning and formatting.

var data = [{year: 2006, books: 54},

{year: 2007, books: 43},

{year: 2008, books: 41},

{year: 2009, books: 44},

{year: 2010, books: 35}];

var barWidth = 40;

var width = (barWidth + 10) \* data.length;

var height = 200;

var x = d3.scale.linear().domain([0, data.length]).range([0, width]);

var y = d3.scale.linear().domain([0, d3.max(data, function(datum) { return datum.books; })]).

rangeRound([0, height]);

// add the canvas to the DOM

var barDemo = d3.select("#bar-demo").

append("svg:svg").

attr("width", width).

attr("height", height);

barDemo.selectAll("rect").

data(data).

enter().

append("svg:rect").

attr("x", function(datum, index) { return x(index); }).

attr("y", function(datum) { return height - y(datum.books); }).

attr("height", function(datum) { return y(datum.books); }).

attr("width", barWidth).

attr("fill", "#2d578b");

### Adding Text and Axes

To add numbers inside the bars, [svg:text](http://www.w3.org/TR/SVG/text.html) elements must be positioned in the same place as the top of the bar and padding added to make it look right. The svg:text element has a lot of attributes for positioning and formatting. In this example, [dx](http://www.w3.org/TR/SVG/text.html#TextElementDXAttribute), [dy](http://www.w3.org/TR/SVG/text.html#TextElementDYAttribute), [text-anchor](http://www.w3.org/TR/SVG/text.html#TextAnchorProperty) and [style](http://www.w3.org/TR/SVG/styling.html#StyleAttribute) are used.

Again, selectAll is used get a selection of elements, data is bound to them with data and then enter is used to add the elements to to the chart.

barDemo.selectAll("text").

data(data).

enter().

append("svg:text").

attr("x", function(datum, index) { return x(index) + barWidth; }).

attr("y", function(datum) { return height - y(datum.books); }).

attr("dx", -barWidth/2).

attr("dy", "1.2em").

attr("text-anchor", "middle").

text(function(datum) { return datum.books;}).

attr("fill", "white");

5443414435

To add an x axis, the height of the chart must be increased to make room. A padding of 30 pixels will work nicely. By keeping the old height variable and increasing the height of the svg:svg canvas by padding, the rest of the code does not need to change. The axis could be positioned by using height + padding as the baseline, but using translate makes the x and y attributes simpler.

barDemo.selectAll("text.yAxis").

data(data).

enter().append("svg:text").

attr("x", function(datum, index) { return x(index) + barWidth; }).

attr("y", height).

attr("dx", -barWidth/2).

attr("text-anchor", "middle").

attr("style", "font-size: 12; font-family: Helvetica, sans-serif").

text(function(datum) { return datum.year;}).

attr("transform", "translate(0, 18)").

attr("class", "yAxis");

# Summary

This library will make graphing projects that range from the very simple to the unbelievably complex. The commands are straight forward and easy to understand. This would be a good way to impress a customer and before looking at this library I now believe that jquery can take graphing date to a new level.