d3 (core)

Selections

d3.select - select an element from the current doc.
d3.selectAll - select multiple elements from the current doc.
selection.attr - get/set attr vals.
selection.classed - add/remove CSS classes.
selection.style - get/set style properties.
selection.property - get/set raw properties.
selection.text - get/set text content.
selection.html - get/set inner HTML content.
selection.append - create and append new elements.
selection.insert - create and insert new elements before existing elements.
selection.remove - remove elements from the doc.
selection.data - get/set data for a group of elements, while computing a relational join.
selection.enter - returns placeholders for missing elements.
selection.exit - returns elements that are no longer needed.
selection.datum - get/set data for individual elements, without computing a join.
selection.filter - filter a selection based on data.
selection.sort - sort elements in the doc based on data.
selection.order - reorders elements in the doc to match the selection.
selection.on - add/remove event listeners for interaction.
selection.transition - start a trans. on the selected elements.
selection.each - call a fcn for each selected element.
selection.call - call a fcn passing in the current selection.
selection.empty - returns true if the selection is empty.
selection.node - access the first node in a selection.
selection.select - subselect a descendant element for each selected element.
selection.selectAll - subselect multiple descendants for each selected element.
d3.selection - augment the selection prototype, or test instance types.
d3.event - access the current user event for interaction.
d3.mouse - gets the mouse position relative to a specified container.
d3.touches - gets the touch positions relative to a specified container.

Transitions

d3.transition - start an animated trans..
transition.delay - specify per-element delay in ms.
transition.duration - specify per-element duration in ms.
transition.ease - specify trans. easing fcn.
transition.attr - smoothly trans to the new attr val.
transition.attrTween - smoothly trans b/w two attr vals.
transition.style - smoothly trans to the new style property val.
transition.styleTween - smoothly trans b/w two style property vals.
transition.text - set the text content when the trans starts.
transition.tween - specify a custom tween operator to run as part of the trans.
transition.select - start a trans on a descendant element for each selected element.
transition.selectAll - start a trans on multiple descendants for each selected element.
transition.filter - filter a trans based on data.
transition.transition - when this trans ends, start another one on the same elements.
transition.remove - remove sel. elements at the end of a trans.
transition.each - add a listener for transition end events.
transition.call - call a fcn passing in the current trans.
d3.ease - customize trans timing.
ease - a parametric easing fcn.
d3.timer - start a custom animation timer.
d3.timer.flush - immediately execute any zero-delay timers.
d3.interpolate - interpolate two vals.
interpolate - a parametric interpolation fcn.
d3.interpolateNumber - interpolate two numbers.
d3.interpolateRound - interpolate two integers.
d3.interpolateString - interpolate two strings.
d3.interpolateRgb - interpolate two RGB colors.
d3.interpolateHsl - interpolate two HSL colors.
d3.interpolateLab - interpolate two L*a*b* colors.
d3.interpolateHcl - interpolate two HCL colors.
d3.interpolateArray - interpolate two arrays of vals.
d3.interpolateObject - interpolate two arbitrary objects.
d3.interpolateTransform - interpolate two 2D matrix trans.
d3.interpolators - register a custom interpolator.

Working with Arrays

d3.ascending - compare two values for sorting.
d3.descending - compare two values for sorting.
d3.min - find the min value in an array.
d3.max - find the max value in an array.
d3.extent - find the min and max value in an array.
d3.sum - compute the sum of an array.
d3.mean - compute the arithmetic mean of an array.
d3.median - compute the median of an array (the 0.5-quantile).
d3.quantile - compute a quantile for a sorted array.
d3.bisect - search for a value in a sorted array.
d3.bisectRight - search for a value in a sorted array.
d3.bisectLeft - search for a value in a sorted array.
d3.bisector - bisect using an accessor.
d3.shuffle - randomize the order of an array.
d3.permute - reorder an array of elements according to an array of indexes.
d3.zip - transpose a variable number of arrays.
d3.transpose - transpose an array of arrays.
d3.keys - list the keys of an assoc array.
d3.values - list the values of an associated array.
d3.entries - list the key-value entries of an assoc array.
d3.merge - merge multiple arrays into one array.
d3.range - generate a range of numeric vals.
d3.nest - group array elements hierarchically.
nest.key - add a level to the nest hierarchy.
nest.sortKeys - sort the current nest level by key.
nest.sortValues - sort the leaf nest level by val.
nest.rollup - specify a rollup fcn for leaf vals.
nest.map - evaluate the nest operator, ret an assoc array.
nest.entries - evaluate the nest operator, ret an array of key-values tuples.
d3.map - a shim for ES6 maps, since objects are not hashes!
map.has - returns true if the map contains the specified key.
map.get - returns the value for the specified key.
map.set - sets the value for the specified key.
map.remove - removes the entry for specified key.
map.keys - returns the maps array of keys.
map.values - returns the maps array of vals.
map.entries - returns the maps array of entries (key-values objects).
map.forEach - calls the specified fcn for each entry in the map.
d3.set - a shim for ES6 sets, since objects are not hashes!
set.has - returns true if the set contains the specified val.
set.add - adds the specified val.
set.remove - removes the specified val.
set.values - returns the sets array of vals.
set.forEach - calls the specified fcn for each val in the set.

Math

d3.random.normal - generate a RV with a normal dist.
d3.random.logNormal - generate a RV with a log-normal dist.
d3.random.irwinHall - generate a RV with an IrwinHall dist.
d3.transform - compute the std form of a 2D matrix trans.

String Formatting

d3.format - format a number as a string.
d3.formatPrefix - returns the [SI prefix] for the specified val and precision.
d3.requote - quote a string for use in a regular expression.
d3.round - rounds a val to some digits after the decimal point.

Loading External Resources

d3.xhr - request a resource using XMLHttpRequest.
xhr.header - set a request header.
xhr.mimeType - set the Accept request header and override the response MIME type.
xhr.response - set a response mapping fcn.
xhr.get - issue a GET request.
xhr.post - issue a POST request.
xhr.send - issue a request with the specified method and data.
xhr.abort - abort an outstanding request.
xhr.on - add an event listener for "progress", "load" or "error" events.
d3.text - request a text file.
d3.json - request a JSON blob.
d3.html - request an HTML doc fragment.
d3.xml - request an XML doc fragment.
d3.csv - request a comma-separated values (CSV) file.
d3.tsv - request a tab-separated values (TSV) file.

CSV Formatting (d3.csv)

d3.csv - request a comma-separated values (CSV) file.
d3.csv.parse - parse a CSV string into objects using the header row.
d3.csv.parseRows - parse a CSV string into tuples, ignoring the header row.
d3.csv.format - format an array of objects into a CSV string.
d3.csv.formatRows - format an array of tuples into a CSV string.
d3.tsv - request a tab-separated values (TSV) file.
d3.tsv.parse - parse a TSV string into objects using the header row.
d3.tsv.parseRows - parse a TSV string into tuples, ignoring the header row.
d3.tsv.format - format an array of objects into a TSV string.
d3.tsv.formatRows - format an array of tuples into a TSV string.

Colors

d3.rgb - specify a color in RGB space.
rgb.brighter - increase RGB channels by some exp. factor.
rgb.darker - decrease RGB channels by some exp. factor.
rgb.hsl - convert from RGB to HSL.
rgb.toString - convert an RGB color to a string.
d3.hsl - specify a color in HSL space.
hsl.brighter - increase lightness by some exp. factor.
hsl.darker - decrease lightness by some exp. factor.
hsl.rgb - convert from HSL to RGB.
hsl.toString - convert an HSL color to a string.
d3.lab - specify a color in L*a*b* space.
lab.brighter - increase lightness by some exp. factor.
lab.darker - decrease lightness by some exp. factor.
lab.rgb - convert from L*a*b* to RGB.
lab.toString - convert a L*a*b* color to a string.
d3.hcl - specify a color in HCL space.
hcl.brighter - increase lightness by some exp. factor.
hcl.darker - decrease lightness by some exp. factor.
hcl.rgb - convert from HCL to RGB.
hcl.toString - convert an HCL color to a string.

Namespaces

d3.ns.prefix - access/extend known XML namespaces.
d3.ns.qualify - qualify a prefixed name, such as ”xlink:href”.

Internals

d3.functor - create a fcn that returns a constant.
d3.rebind - rebind an inherited getter/setter method to a subclass.
d3.dispatch - create custom event dispatchers.
dispatch.on - register an event listener.
dispatch - dispatch an event to registered listeners.

d3.scale (Scales)

Quantitative

d3.scale.linear - construct a linear quantitative scale.
linear - get the range val corresp to a given domain val.
linear.invert - get the domain val corresp to a given range val.
linear.domain - get/set the scale's input domain.
linear.range - get/set the scale's output range.
linear.rangeRound - set the scale's output range, and enable rounding.
linear.interpolate - get/set the scale's output interpolator.
linear.clamp - enable/disable clamping of the output range.
linear.nice - extend the scale domain to nice round numbers.
linear.ticks - get representative values from the input domain.
linear.tickFormat - get a formatter for displaying tick vals.
linear.copy - create a new scale from an existing scale.
d3.scale.sqrt - construct a quantitative scale with a square root trans.
d3.scale.pow - construct a quantitative scale with an exponential trans.
pow - get the range val corresp to a given domain val.
pow.invert - get the domain val corresp to a given range val.
pow.domain - get/set the scale's input domain.
pow.range - get/set the scale's output range.
pow.rangeRound - set the scale's output range, and enable rounding.
pow.interpolate - get/set the scale's output interpolator.
pow.clamp - enable/disable clamping of the output range.
pow.nice - extend the scale domain to nice round numbers.
pow.ticks - get representative values from the input domain.
pow.tickFormat - get a formatter for displaying tick vals.
pow.exponent - get/set the exponent power.
pow.copy - create a new scale from an existing scale.
d3.scale.log - construct a quantitative scale with an logarithmic trans.
log - get the range val corresp to a given domain val.
log.invert - get the domain val corresp to a given range val.
log.domain - get/set the scale's input domain.
log.range - get/set the scale's output range.
log.rangeRound - set the scale's output range, and enable rounding.
log.interpolate - get/set the scale's output interpolator.
log.clamp - enable/disable clamping of the output range.
log.nice - extend the scale domain to nice powers of ten.
log.ticks - get representative values from the input domain.
log.tickFormat - get a formatter for displaying tick vals.
log.copy - create a new scale from an existing scale.
d3.scale.quantize - construct a linear quantitative scale with a discrete output range.
quantize - get the range val corresp to a given domain val.
quantize.domain - get/set the scale's input domain.
quantize.range - get/set the scale's output range (discrete).
quantize.copy - create a new scale from an existing scale.
d3.scale.threshold - construct a threshold scale with a discrete output range.
threshold - get the range val corresp to a given domain val.
threshold.domain - get/set the scale's input domain.
threshold.range - get/set the scale's output range (discrete).
threshold.copy - create a new scale from an existing scale.
d3.scale.quantile - construct a quantitative scale mapping to quantiles.
quantile - get the range val corresp to a given domain val.
quantile.domain - get/set the scale's input domain (discrete).
quantile.range - get/set the scale's output range (discrete).
quantile.quantiles - get the scale's quantile bin thresholds.
quantile.copy - create a new scale from an existing scale.
d3.scale.identity - construct a linear identity scale.
identity - the identity fcn.
identity.invert - equivalent to identity; the identity fcn.
identity.domain - get/set the scale's domain and range.
identity.range - equivalent to identity.domain.
identity.ticks - get representative values from the domain.
identity.tickFormat - get a formatter for displaying tick vals.
identity.copy - create a new scale from an existing scale.

Ordinal

d3.scale.ordinal - construct an ordinal scale.
ordinal - get the range val corresp to a given domain val.
ordinal.domain - get/set the scale's input domain.
ordinal.range - get/set the scale's output range.
ordinal.rangePoints - divide a continuous output range for discrete points.
ordinal.rangeBands - divide a continuous output range for discrete bands.
ordinal.rangeRoundBands - divide a continuous output range for discrete bands.
ordinal.rangeBand - get the discrete range band width.
ordinal.rangeExtent - get the min and max values of the output range.
ordinal.copy - create a new scale from an existing scale.
d3.scale.category10 - constr an ord scale w/ 10 categ cols.
d3.scale.category20 - constr an ord scale w/ 20 categ cols.
d3.scale.category20b - constr an ord scale w/ 20 categ cols.
d3.scale.category20c - constr an ord scale w/ 20 categ cols.

d3.svg (SVG)

Shapes

d3.svg.line - create a new line generator.
line - generate a piecewise linear curve, as in a line chart.
line.x - get/set the x-coord accessor.
line.y - get/set the y-coord accessor.
line.interpolate - get/set the interpolation mode.
line.tension - get/set the cardinal spline tension.
line.defined - control whether the line is def at a given point.
d3.svg.line.radial - create a new radial line generator.
line - generate a piecewise linear curve, as in a polar line chart.
line.radius - get/set the rad accessor.
line.angle - get/set the angle accessor.
line.defined - control whether the line is def at a given point.
d3.svg.area - create a new area generator.
area - generate a piecewise linear area, as in an area chart.
area.x - get/set the x-coord accessors.
area.x0 - get/set the x0-coord (baseline) accessor.
area.x1 - get/set the x1-coord (topline) accessor.
area.y - get/set the y-coord accessors.
area.y0 - get/set the y0-coord (baseline) accessor.
area.y1 - get/set the y1-coord (topline) accessor.
area.interpolate - get/set the interpolation mode.
area.tension - get/set the cardinal spline tension.
area.defined - control whether the area is def at a given point.
d3.svg.area.radial - create a new area generator.
area - generate a piecewise linear area, as in a polar area chart.
area.radius - get/set the rad accessors.
area.innerRadius - get/set the inner rad (baseline) accessor.
area.outerRadius - get/set the outer rad (topline) accessor.
area.angle - get/set the angle accessors.
area.startAngle - get/set the angle (baseline) accessor.
area.endAngle - get/set the angle (topline) accessor.
area.defined - control whether the area is def at a given point.
d3.svg.arc - create a new arc generator.
arc - generate a solid arc, as in a pie/donut chart.
arc.innerRadius - get/set the inner rad accessor.
arc.outerRadius - get/set the outer rad accessor.
arc.startAngle - get/set the start angle accessor.
arc.endAngle - get/set the end angle accessor.
arc.centroid - compute the arc centroid.
d3.svg.symbol - create a new symbol generator.
symbol - generate categ symbols, as in a scatterplot.
symbol.type - get/set the symbol type accessor.
symbol.size - get/set the symbol size (in square px) accessor.
d3.svg.symbolTypes - the array of supported symbol types.
d3.svg.chord - create a new chord generator.
chord - generate a quadratic Bzier connecting two arcs, as in a chord diagram.
chord.radius - get/set the arc rad accessor.
chord.startAngle - get/set the arc start angle accessor.
chord.endAngle - get/set the arc end angle accessor.
chord.source - get/set the source arc accessor.
chord.target - get/set the target arc accessor.
d3.svg.diagonal - create a new diagonal generator.
diagonal - generate a two-dim Bzier connector, as in a node-link diagram.
diagonal.source - get/set the source point accessor.
diagonal.target - get/set the target point accessor.
diagonal.projection - get/set an optional point transform.
d3.svg.diagonal.radial - create a new diagonal generator.
diagonal - generate a two-dim Bzier connector, as in a node-link diagram.

Axes

d3.svg.axis - create a new axis generator.
axis - creates/updates an axis for the given sel. or trans.
axis.scale - get/set the axis scale.
axis.orient - get/set the axis orientation.
axis.ticks - control how ticks are generated for the axis.
axis.tickValues - specify tick values explicitly.
axis.tickSubdivide - optionally subdivide ticks uniformly.
axis.tickSize - specify the size of major, minor and end ticks.
axis.tickPadding - specify padding b/w ticks and tick labels.
axis.tickFormat - override the tick formatting for labels.

Controls

d3.svg.brush - click and drag to select one- or two-dim regions.
brush - creates or updates a brush for the given sel. or trans.
brush.x - get/set the brushs x-scale.
brush.y - get/set the brushs y-scale.
brush.extent - get/set the brushs extent.
brush.clear - reset the brush extent.
brush.empty - returns true if the brush extent is empty.
brush.on - respond to events when the brush is moved.

d3.time (Time)

Time Formatting

d3.time.format - create a new local time formatter for a given specifier.
format - format a date into a string.
format.parse - parse a string into a date.
d3.time.format.utc - create a new UTC time formatter for a given specifier.
d3.time.format.iso - the ISO 8601 UTC time formatter.

Time Scales

d3.time.scale - construct a linear time scale.
scale - get the range val corresp to a given domain val.
scale.invert - get the domain val corresp to a given range val.
scale.domain - get/set the scale's input domain.
scale.range - get/set the scale's output range.
scale.rangeRound - set the scale's output range, and enable rounding.
scale.interpolate - get/set the scale's output interpolator.
scale.clamp - enable/disable clamping of the output range.
scale.ticks - get representative values from the input domain.
scale.tickFormat - get a formatter for displaying tick vals.
scale.copy - create a new scale from an existing scale.

Time Intervals

d3.time.interval - a time interval in local time.
interval - alias for interval.floor.
interval.range - returns dates within the specified range.
interval.floor - rounds down to the nearest interval.
interval.round - rounds up/down to the nearest interval.
interval.ceil - rounds up to the nearest interval.
interval.offset - returns a date offset by some interval.
interval.utc - returns the UTC-equivalent time interval.
d3.time.day - every day (12:00 AM).
d3.time.days - alias for day.range.
d3.time.dayOfYear - computes the day num.
d3.time.hour - every hour (e.g., 1:00 AM).
d3.time.hours - alias for hour.range.
d3.time.minute - every minute (e.g., 1:02 AM).
d3.time.minutes - alias for minute.range.
d3.time.month - every month (e.g., February 1, 12:00 AM).
d3.time.months - alias for month.range.
d3.time.second - every second (e.g., 1:02:03 AM).
d3.time.seconds - alias for second.range.
d3.time.sunday - every Sunday.
d3.time.sundays - alias for sunday.range.
d3.time.sundayOfYear - computes the sun-based wk num.
d3.time.monday - every Monday.
d3.time.mondays - alias for monday.range.
d3.time.mondayOfYear - computes the mon-based wk num.
d3.time.tuesday - every Tuesday.
d3.time.tuesdays - alias for tuesday.range.
d3.time.tuesdayOfYear - computes the tues-based wk num.
d3.time.wednesday - every Wednesday.
d3.time.wednesdays - alias for wednesday.range.
d3.time.wednesdayOfYear - computes the wed-based wk num
d3.time.thursday - every Thursday.
d3.time.thursdays - alias for thursday.range.
d3.time.thursdayOfYear - computes the thurs-based wk num.
d3.time.friday - every Friday.
d3.time.fridays - alias for friday.range.
d3.time.fridayOfYear - computes the fri-based wk num.
d3.time.saturday - every Saturday.
d3.time.saturdays - alias for saturday.range.
d3.time.saturdayOfYear - computes the sat-based wk num.
d3.time.week - alias for sunday.
d3.time.weeks - alias for sunday.range.
d3.time.weekOfYear - alias for sundayOfYear.
d3.time.year - every year (e.g., January 1, 12:00 AM).
d3.time.years - alias for year.range.

d3.layout (Layouts)

Bundle

d3.layout.bundle - construct a new default bundle layout.
bundle - apply Holten's hierarchical bundling algorithm to edges.

Chord

d3.layout.chord - produce a chord diagram from a matrix of relationships.
chord.matrix - get/set the matrix data backing the layout.
chord.padding - get/set the angular padding b/w chord segments.
chord.sortGroups - get/set the comparator fcn for groups.
chord.sortSubgroups - get/set the comparator fcn for subgroups.
chord.sortChords - get/set the comparator fcn for chords (z-order).
chord.chords - retrieve the computed chord angles.
chord.groups - retrieve the computed group angles.

Cluster

d3.layout.cluster - cluster entities into a dendrogram.
cluster.sort - get/set the comparator fcn for sibling nodes.
cluster.children - get/set the accessor fcn for child nodes.
cluster.nodes - compute the cluster layout and return the array of nodes.
cluster.links - compute the parent-child links b/w tree nodes.
cluster.separation - get/set the spacing fcn b/w neighboring nodes.
cluster.size - get/set the layout size in x and y.

Force

d3.layout.force - position linked nodes using physical sim.
force.on - listen to updates in the computed layout positions.
force.nodes - get/set the array of nodes to layout.
force.links - get/set the array of links b/w nodes.
force.size - get/set the layout size in x and y.
force.linkDistance - get/set the link distance.
force.linkStrength - get/set the link strength.
force.friction - get/set the friction coefficient.
force.charge - get/set the charge strength.
force.gravity - get/set the gravity strength.
force.theta - get/set the accuracy of the charge interaction.
force.start - start/restart the sim when the nodes change.
force.resume - reheat the cooling parameter and restart sim.
force.stop - immediately terminate the sim.
force.alpha - get/set the layout's cooling parameter.
force.tick - run the layout sim one step.
force.drag - bind a behavior to nodes to allow interactive dragging.

Hierarchy

d3.layout.hierarchy - derive a custom hierarchical layout implementation.
hierarchy.sort - get/set the comparator fcn for sibling nodes.
hierarchy.children - get/set the accessor fcn for child nodes.
hierarchy.nodes - compute the layout and return the array of nodes.
hierarchy.links - compute the parent-child links b/w tree nodes.
hierarchy.value - get/set the val accessor fcn.
hierarchy.revalue - recompute the hierarchy vals.

Histogram

d3.layout.histogram - construct a new default histogram.
histogram - compute the dist of data using quantized bins.
histogram.value - get/set the val accessor fcn.
histogram.range - get/set the considered val range.
histogram.bins - specify how values are organized into bins.
histogram.frequency - compute the dist as counts/probabilities.

Pack

d3.layout.pack - produce a hierarchical layout using recursive circle-packing.
pack.sort - control the order in which sibling nodes are traversed.
pack.children - get/set the children accessor fcn.
pack.nodes - compute the pack layout and return the array of nodes.
pack.links - compute the parent-child links b/w tree nodes.
pack.value - get/set the val accessor used to size circles.
pack.size - specify the layout size in x and y.
pack.padding - specify the layout padding in (approx) px.

Partition

d3.layout.partition - recursively partition a node tree into a sunburst or icicle.
partition.sort - control the order in which sibling nodes are traversed.
partition.children - get/set the children accessor fcn.
partition.nodes - compute the partition layout and return the array of nodes.
partition.links - compute the parent-child links b/w tree nodes.
partition.value - get/set the val accessor used to size circles.
partition.size - specify the layout size in x and y.

Pie

d3.layout.pie - construct a new default pie layout.
pie - compute the start/end angles for arcs in a pie/donut chart.
pie.value - get/set the val accessor fcn.
pie.sort - control the clockwise order of pie slices.
pie.startAngle - get/set the overall start angle of the pie.
pie.endAngle - get/set the overall end angle of the pie.

Stack

d3.layout.stack - construct a new default stack layout.
stack - compute the baseline for each series in a stacked bar/area chart.
stack.values - get/set the values accessor fcn per series.
stack.order - control the order in which series are stacked.
stack.offset - specify the overall baseline algorithm.
stack.x - get/set the x-dimension accessor fcn.
stack.y - get/set the y-dimension accessor fcn.
stack.out - get/set the output fcn for storing the baseline.

Tree

d3.layout.tree - position a tree of nodes tidily.
tree.sort - control the order in which sibling nodes are traversed.
tree.children - get/set the children accessor fcn.
tree.nodes - compute the tree layout and return the array of nodes.
tree.links - compute the parent-child links b/w tree nodes.
tree.separation - get/set the spacing fcn b/w neighboring nodes.
tree.size - specify the layout size in x and y.

Treemap

d3.layout.treemap - use recursive spatial subdivision to display a tree of nodes.
treemap.sort - control the order in which sibling nodes are traversed.
treemap.children - get/set the children accessor fcn.
treemap.nodes - compute the treemap layout and return the array of nodes.
treemap.links - compute the parent-child links b/w tree nodes.
treemap.value - get/set the val accessor used to size treemap cells.
treemap.size - specify the layout size in x and y.
treemap.padding - specify the padding b/w a parent and its children.
treemap.round - enable/disable rounding to exact px.
treemap.sticky - make the layout sticky for stable updates.
treemap.mode - change the treemap layout algorithm.

d3.geo (Geography)

Paths

d3.geo.path - create a new geographic path generator.
path - project the specified feature and render it to the context.
path.projection - get/set the geographic proj.
path.context - get/set the render context.
path.pointRadius - get/set the radius to display point features.
path.area - compute the proj area of a given feature.
path.centroid - compute the proj centroid of a given feature.
path.bounds - compute the proj bounds of a given feature.
d3.geo.circle - create a circle generator.
circle - generate a piecewise circle as a Polygon.
circle.origin - specify the origin in lat and long.
circle.angle - specify the angular radius in degrees.
circle.precision - specify the precision of the piecewise circle.
d3.geo.area - compute the spherical area of a given feature.
d3.geo.bounds - compute the lat-long bounding box for a feature.
d3.geo.centroid - compute the spherical centroid of a feature.
d3.geo.distance - compute the great-arc dist b/w two points.
d3.geo.interpolate - interpolate b/w 2 points along a great arc.
d3.geo.length - compute the length of a line string/the circumf. of a polygon.

Projections

d3.geo.projection - create a standard proj from a raw proj
projection - project the specified location
projection.invert - invert the proj for the specified point
projection.rotate - get/set the proj's three-axis rotation
projection.center - get/set the proj's center location
projection.translate - get/set the proj's translation pos
projection.scale - get/set the proj's scale factor
projection.clipAngle - get/set the rad of the proj's clip circle
projection.clipExtent - get/set the proj viewport clip ext(px)
projection.precision - get/set the precision threshold for adaptive resampling
projection.stream - wrap the specified stream listener, projecting input geometry
d3.geo.projectionMutator - create a standard proj from a mutable raw proj
d3.geo.albers - the Albers equal-area conic proj
albers.parallels - get/set the proj's two standard parallels
d3.geo.albersUsa - a composite Albers proj for the US
d3.geo.azimuthalEqualArea - the azimuthal equal-area proj
d3.geo.azimuthalEquidistant - the azimuthal equidist proj
d3.geo.conicConformal - the conic conformal projection
d3.geo.conicEquidistant - the conic equidist projection
d3.geo.conicEqualArea - the conic equal-area (Albers) proj
d3.geo.equirectangular - the equirect(plate carrèe) proj
d3.geo.gnomonic - the gnomonic proj.
d3.geo.mercator - the spherical Mercator proj
d3.geo.orthographic - the azimuthal orthographic proj
d3.geo.stereographic - the azimuthal stereographic proj
d3.geo.azimuthalEqualArea.raw - the raw azim eq-area proj
d3.geo.azimuthalEquidistant.raw - the azim equidist proj
d3.geo.conicConformal.raw - the raw conic conformal proj
d3.geo.conicEquidistant.raw - the raw conic equidist proj
d3.geo.conicEqualArea.raw - the raw conic equal-area (Albers) proj
d3.geo.equirectangular.raw - raw equirect(plate carrèe) proj
d3.geo.gnomonic.raw - the raw gnomonic proj
d3.geo.mercator.raw - the raw Mercator proj
d3.geo.orthographic.raw - the raw azimuthal orthog proj
d3.geo.stereographic.raw - the raw azimuthal stereog proj
d3.geo.transverseMercator.raw - the raw transverse Mercator proj

Streams

d3.geo.stream - convert a GeoJSON object to a geometry stream.
stream.point - indicate an x, y (and optionally z) coord.
stream.lineStart - indicate the start of a line or ring.
stream.lineEnd - indicate the end of a line or ring.
stream.polygonStart - indicate the start of a polygon.
stream.polygonEnd - indicate the end of a polygon.
stream.sphere - indicate a sphere.

d3.geom (Geometry)

Voronoi

d3.geom.voronoi - compute the Voronoi diagram for the specified points.
d3.geom.delaunay - compute the Delaunay triangulation for the specified points.

Quadtree

d3.geom.quadtree - constructs a quadtree for an array of points.
quadtree.add - add a point to the quadtree.
quadtree.visit - recursively visit nodes in the quadtree.

Polygon

d3.geom.polygon -
polygon.area -
polygon.centroid -
polygon.clip -

Hull

d3.geom.hull -

d3.behavior (Behaviors)

Drag

d3.behavior.drag -
drag.origin -
drag.on -

Zoom

d3.behavior.zoom -
zoom.on -
zoom.scale -
zoom.translate -
zoom.scaleExtent -
zoom.x -
zoom.y -