> *Wiki* ‣ ***API リファレンス***

D3 ライブラリの全ては名前空間 `d3` 以下に存在します・

D3 はセマンティックバージョニングを使用しています。使用中の D3 のバージョンは `d3.version` として参照する事ができます。

See one of:

- Behaviors - 再利用可能なインタラクション挙動
- Core - selection 、 transition 、 データ、ローカライズ、色、等
- Geography - 極座標データの表示、緯度と経度の計算
- Geometry - ボロノイ図や四分木といった二次元幾何学ユーティリティ
- Layouts - 要素の位置決めのための二次データの導出
- Scales - データから視覚情報への変換
- SVG - Scalable Vector Graphics 作成の為のユーティリティ
- Time - 時刻のパースとフォーマット、カレンダーと周期の計算、等

# d3 (core)

## Selections

- d3.event - インタラクションのためのユーザイベントへアクセス
- d3.mouse - 指定されたコンテナに対する相対マウス位置を取得
- d3.select - 現在のドキュメントから一つの要素を選択
- d3.selectAll - 現在のドキュメントから複数の要素を選択
- d3.selection - selection プロトタイプの拡張、もしくはインスタンス型のテスト
- d3.touch - 指定されたコンテナに対する相対タッチ位置を取得
- d3.touches - 指定されたコンテナに対する複数の相対タッチ位置を取得
- selection.append - 新しい要素の作成と追加
- selection.attr - 属性値の取得と設定
- selection.call - 現在の selection を渡しての関数呼び出し
- selection.classed - CSS クラスの追加と削除
- selection.data - データと要素の対応づけを伴う、要素のグループに対するデータの設定と取得
- selection.datum - データと要素の対応づけを伴わない、個々の要素に対するデータの設定と取得
- selection.each - 選択された要素それぞれに対する関数呼び出し
- selection.empty - selection が空の場合 true を返す
- selection.enter - 存在しない要素のためのプレースホルダを返す
- selection.exit - 不必要になった要素を返す
- selection.filter - データに基づいた selection の絞り込み
- selection.html - 要素の innnerHTML プロパティの設定と取得
- selection.insert - 要素が存在するようになる前の新しい要素の作成と挿入
- selection.interrupt - 現在の transition に対する即時割り込み
- selection.node - selection 中の最初の要素を返す
- selection.on - インタラクションのためにイベントリスナの追加と削除
- selection.order - ドキュメント中の要素を selection に合うように並べ替える
- selection.property - 未加工プロパティの取得
- selection.remove - ドキュメントからの要素の削除
- selection.select - 選択された要素それぞれから一つの子要素を選択
- selection.selectAll - 選択された要素それぞれから複数の子要素を選択
- selection.size - selection 中の要素の数を返す
- selection.sort - データに基づいたドキュメント中の要素のソート

- selection.style - スタイルプロパティの設定と取得
- selection.text - 要素の textContent プロパティの設定と取得
- selection.transition - 選択された要素の遷移を開始

## Transitions

- d3.ease - 遷移タイミングをカスタマイズ
- d3.timer - カスタムアニメーションタイマの開始
- d3.interpolate - 二値の補間
- d3.interpolateArray - 二つの配列の値の補間
- d3.interpolateHcl - 二つの HCL 形式色データの補間
- d3.interpolateHsl - 二つの HSL 形式色データの補間
- d3.interpolateLab - 二つの L*a*b* 形式色データの補間
- d3.interpolateNumber - 二つの数値の補間
- d3.interpolateNumber - 二つの任意のオブジェクトの補間
- d3.interpolateRgb - 二つの RGB 形式色データの補間
- d3.interpolateRound - 二つの整数の補間
- d3.interpolateRound - 二つの文字列の補間
- d3.interpolateString - interpolate two strings.
- d3.interpolateTransform - 二つの二次元配列の補間
- d3.interpolateZoom - 二つの点の滑らかなズームとパン
- d3.interpolators - カスタム補間関数の登録
- d3.timer.flush - ゼロ遅延タイマの即時実行
- d3.transition - 遷移アニメーションの開始
- ease - パラメータ制御平滑化関数
- interpolate - パラメータ制御補間関数
- transition.attr - 新しい属性値へ滑らかに遷移
- transition.attrTween - 二つの属性値の間で滑らかに遷移
- transition.call - 現在の transition に対する関数の呼び出し
- transition.delay - 要素毎のミリセカンド単位の遅延の指定
- transition.duration - 要素毎のミリセカンド単位の持続時間の指定
- transition.each - 遷移終了イベントのリスナを追加
- transition.ease - 遷移平滑化関数の指定
- transition.empty - transition が空のとき true を返す
- transition.filter - データに基づいた transition の絞り込み
- transition.node - transition の最初のノードを返す
- transition.remove - transition の終了時に指定された要素を削除
- transition.select - 選択された要素それぞれのうち一つの子要素について遷移を開始
- transition.selectAll - 選択された要素の全ての子要素について遷移を開始
- transition.size - selection 中の要素の数を返す
- transition.style - 新しいスタイルプロパティ値へ滑らかに遷移
- transition.styleTween - 二つのスタイルプロパティ値の間で滑らかに遷移
- transition.text - 遷移開始時に内容テキストを設定
- transition.transition - この遷移が終了したら、同じ要素で別の新しい遷移を開始
- transition.tween - transition の一部として実行される、カスタム tween 演算子を指定

## Working with Arrays

- d3.ascending - compare two values for sorting.
- d3.bisectLeft - search for a value in a sorted array.
- d3.bisector - bisect using an accessor or comparator.
- d3.bisectRight - search for a value in a sorted array.

- d3.bisect - search for a value in a sorted array.
- d3.descending - compare two values for sorting.
- d3.deviation - compute the standard deviation of an array of numbers.
- d3.entries - list the key-value entries of an associative array.
- d3.extent - find the minimum and maximum value in an array.
- d3.keys - list the keys of an associative array.
- d3.map - a shim for ES6 maps, since objects are not hashes!
- map.empty - returns false if the map has at least one entry.
- map.entries - returns the map's array of entries (key-values objects).
- map.forEach - calls the specified function for each entry in the map.
- map.get - returns the value for the specified key.
- map.has - returns true if the map contains the specified key.
- map.keys - returns the map's array of keys.
- map.remove - removes the entry for specified key.
- map.set - sets the value for the specified key.
- map.size - returns the number of entries in the map.
- map.values - returns the map's array of values.
- d3.max - find the maximum value in an array.
- d3.mean - compute the arithmetic mean of an array of numbers.
- d3.median - compute the median of an array of numbers (the 0.5-quantile).
- d3.merge - merge multiple arrays into one array.
- d3.min - find the minimum value in an array.
- d3.nest - group array elements hierarchically.
- nest.entries - evaluate the nest operator, returning an array of key-values tuples.
- nest.key - add a level to the nest hierarchy.
- nest.map - evaluate the nest operator, returning an associative array.
- nest.rollup - specify a rollup function for leaf values.
- nest.sortKeys - sort the current nest level by key.
- nest.sortValues - sort the leaf nest level by value.
- d3.pairs - returns an array of adjacent pairs of elements.
- d3.permute - reorder an array of elements according to an array of indexes.
- d3.quantile - compute a quantile for a sorted array of numbers.
- d3.range - generate a range of numeric values.
- d3.set - a shim for ES6 sets, since objects are not hashes!
- set.add - adds the specified value.
- set.empty - returns true if the set has at least one value.
- set.forEach - calls the specified function for each value in the set.
- set.has - returns true if the set contains the specified value.
- set.remove - removes the specified value.
- set.size - returns the number of values in the set.
- set.values - returns the set's array of values.
- d3.shuffle - randomize the order of an array.
- d3.sum - compute the sum of an array of numbers.
- d3.transpose - transpose an array of arrays.
- d3.values - list the values of an associated array.
- d3.variance - compute the variance of an array of numbers.
- d3.zip - transpose a variable number of arrays.

## Math

- d3.random.bates - generate a random number with a Bates distribution.

- d3.random.irwinHall - generate a random number with an Irwin–Hall distribution.
- d3.random.logNormal - generate a random number with a log-normal distribution.
- d3.random.normal - generate a random number with a normal distribution.
- d3.transform - compute the standard form of a 2D matrix transform.

## Loading External Resources

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

## String Formatting

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

## CSV Formatting (d3.csv)

- d3.csv.formatRows - format an array of tuples into a CSV string.
- d3.csv.format - format an array of objects into a CSV string.
- d3.csv.parseRows - parse a CSV string into tuples, ignoring the header row.
- d3.csv.parse - parse a CSV string into objects using the header row.
- d3.csv - request a comma-separated values (CSV) file.
- d3.dsv - create a parser/formatter for the specified delimiter and mime type.
- d3.tsv.formatRows - format an array of tuples into a TSV string.
- d3.tsv.format - format an array of objects into a TSV string.
- d3.tsv.parseRows - parse a TSV string into tuples, ignoring the header row.
- d3.tsv.parse - parse a TSV string into objects using the header row.
- d3.tsv - request a tab-separated values (TSV) file.

## Localization

- d3.locale - create a new locale using the specified strings.
- locale.numberFormat - create a new number formatter.
- locale.timeFormat - create a new time formatter / parser.

## Colors

- d3.hcl - specify a color in HCL space.
- d3.hsl - specify a color in HSL space.
- d3.lab - specify a color in L*a*b* space.

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

### Namespaces

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

### Internals

- d3.dispatch - create a custom event dispatcher.
- d3.functor - create a function that returns a constant.
- d3.rebind - rebind an inherited getter/setter method to a subclass.
- dispatch.on - register or unregister an event listener.
- dispatch.type - dispatch an event to registered listeners.

## d3.scale (Scales)

### Quantitative

- d3.scale.identity - construct a linear identity scale.
- d3.scale.linear - construct a linear quantitative scale.
- d3.scale.log - construct a quantitative scale with an logarithmic transform.
- d3.scale.pow - construct a quantitative scale with an exponential transform.
- d3.scale.quantile - construct a quantitative scale mapping to quantiles.
- d3.scale.quantize - construct a linear quantitative scale with a discrete output range.
- d3.scale.sqrt - construct a quantitative scale with a square root transform.
- d3.scale.threshold - construct a threshold scale with a discrete output range.
- identity.copy - create a new scale from an existing scale.
- identity.domain - get or set the scale's domain and range.
- identity.invert - equivalent to identity; the identity function.
- identity.range - equivalent to identity.domain.
- identity.tickFormat - get a formatter for displaying tick values.
- identity.ticks - get representative values from the domain.
- identity - the identity function.
- linear.clamp - enable or disable clamping of the output range.
- linear.copy - create a new scale from an existing scale.
- linear.domain - get or set the scale's input domain.

- linear.interpolate - get or set the scale's output interpolator.
- linear.invert - get the domain value corresponding to a given range value.
- linear.nice - extend the scale domain to nice round numbers.
- linear.rangeRound - set the scale's output range, and enable rounding.
- linear.range - get or set the scale's output range.
- linear.tickFormat - get a formatter for displaying tick values.
- linear.ticks - get representative values from the input domain.
- linear - get the range value corresponding to a given domain value.
- log.clamp - enable or disable clamping of the output range.
- log.copy - create a new scale from an existing scale.
- log.domain - get or set the scale's input domain.
- log.interpolate - get or set the scale's output interpolator.
- log.invert - get the domain value corresponding to a given range value.
- log.nice - extend the scale domain to nice powers of ten.
- log.rangeRound - set the scale's output range, and enable rounding.
- log.range - get or set the scale's output range.
- log.tickFormat - get a formatter for displaying tick values.
- log.ticks - get representative values from the input domain.
- log - get the range value corresponding to a given domain value.
- pow.clamp - enable or disable clamping of the output range.
- pow.copy - create a new scale from an existing scale.
- pow.domain - get or set the scale's input domain.
- pow.exponent - get or set the exponent power.
- pow.interpolate - get or set the scale's output interpolator.
- pow.invert - get the domain value corresponding to a given range value.
- pow.nice - extend the scale domain to nice round numbers.
- pow.rangeRound - set the scale's output range, and enable rounding.
- pow.range - get or set the scale's output range.
- pow.tickFormat - get a formatter for displaying tick values.
- pow.ticks - get representative values from the input domain.
- pow - get the range value corresponding to a given domain value.
- quantile.copy - create a new scale from an existing scale.
- quantile.domain - get or set the scale's input domain (as discrete values).
- quantile.invertExtent - get the domain values for the specified range value.
- quantile.quantiles - get the scale's quantile bin thresholds.
- quantile.range - get or set the scale's output range (as discrete values).
- quantile - get the range value corresponding to a given domain value.
- quantize.copy - create a new scale from an existing scale.
- quantize.domain - get or set the scale's input domain.
- quantize.invertExtent - get the domain values for the specified range value.
- quantize.range - get or set the scale's output range (as discrete values).
- quantize - get the range value corresponding to a given domain value.
- threshold.copy - create a new scale from an existing scale.
- threshold.domain - get or set the scale's input domain.
- threshold.invertExtent - get the domain values for the specified range value.
- threshold.range - get or set the scale's output range (as discrete values).
- threshold - get the range value corresponding to a given domain value.

## Ordinal

- d3.scale.category10 - construct an ordinal scale with ten categorical colors.

- d3.scale.category20b - construct an ordinal scale with twenty categorical colors.
- d3.scale.category20c - construct an ordinal scale with twenty categorical colors.
- d3.scale.category20 - construct an ordinal scale with twenty categorical colors.
- d3.scale.ordinal - construct an ordinal scale.
- ordinal.copy - create a new scale from an existing scale.
- ordinal.domain - get or set the scale's input domain.
- ordinal.rangeBands - divide a continuous output range for discrete bands.
- ordinal.rangeBand - get the discrete range band width.
- ordinal.rangeExtent - get the minimum and maximum values of the output range.
- ordinal.rangePoints - divide a continuous output range for discrete points.
- ordinal.rangeRoundBands - divide a continuous output range for discrete bands.
- ordinal.rangeRoundPoints - divide a continuous output range for discrete points.
- ordinal.range - get or set the scale's output range.
- ordinal - get the range value corresponding to a given domain value.

## d3.svg (SVG)

### Shapes

- arc.centroid - compute the arc centroid.
- arc.cornerRadius - get or set the corner radius accessor.
- arc.endAngle - get or set the end angle accessor.
- arc.innerRadius - get or set the inner radius accessor.
- arc.outerRadius - get or set the outer radius accessor.
- arc.padAngle - get or set the pad angle accessor.
- arc.padRadius - get or set the pad radius accessor.
- arc.startAngle - get or set the start angle accessor.
- arc - generate a solid arc, as in a pie or donut chart.
- area.angle - get or set the *angle* accessors.
- area.defined - control whether the area is defined at a given point.
- area.defined - control whether the area is defined at a given point.
- area.endAngle - get or set the *angle* (topline) accessor.
- area.innerRadius - get or set the inner *radius* (baseline) accessor.
- area.interpolate - get or set the interpolation mode.
- area.outerRadius - get or set the outer *radius* (topline) accessor.
- area.radius - get or set the *radius* accessors.
- area.startAngle - get or set the *angle* (baseline) accessor.
- area.tension - get or set the cardinal spline tension.
- area.x0 - get or set the *x0*-coordinate (baseline) accessor.
- area.x1 - get or set the *x1*-coordinate (topline) accessor.
- area.x - get or set the *x*-coordinate accessors.
- area.y0 - get or set the *y0*-coordinate (baseline) accessor.
- area.y1 - get or set the *y1*-coordinate (topline) accessor.
- area.y - get or set the *y*-coordinate accessors.
- area - generate a piecewise linear area, as in an area chart.
- area - generate a piecewise linear area, as in a polar area chart.
- chord.endAngle - get or set the arc end angle accessor.
- chord.radius - get or set the arc radius accessor.
- chord.source - get or set the source arc accessor.
- chord.startAngle - get or set the arc start angle accessor.
- chord.target - get or set the target arc accessor.

- chord - generate a quadratic Bézier connecting two arcs, as in a chord diagram.
- d3.svg.arc - create a new arc generator.
- d3.svg.area.radial - create a new area generator.
- d3.svg.area - create a new area generator.
- d3.svg.chord - create a new chord generator.
- d3.svg.diagonal.radial - create a new diagonal generator.
- d3.svg.diagonal - create a new diagonal generator.
- d3.svg.line.radial - create a new radial line generator.
- d3.svg.line - create a new line generator.
- d3.svg.symbolTypes - the array of supported symbol types.
- d3.svg.symbol - create a new symbol generator.
- diagonal.projection - get or set an optional point transform.
- diagonal.source - get or set the source point accessor.
- diagonal.target - get or set the target point accessor.
- diagonal - generate a two-dimensional Bézier connector, as in a node-link diagram.
- diagonal - generate a two-dimensional Bézier connector, as in a node-link diagram.
- line.angle - get or set the *angle* accessor.
- line.defined - control whether the line is defined at a given point.
- line.defined - control whether the line is defined at a given point.
- line.interpolate - get or set the interpolation mode.
- line.interpolate - get or set the interpolation mode.
- line.radius - get or set the *radius* accessor.
- line.tension - get or set the cardinal spline tension.
- line.tension - get or set the cardinal spline tension.
- line.x - get or set the *x*-coordinate accessor.
- line.y - get or set the *y*-coordinate accessor.
- line - generate a piecewise linear curve, as in a line chart.
- line - generate a piecewise linear curve, as in a polar line chart.
- symbol.size - get or set the symbol size (in square pixels) accessor.
- symbol.type - get or set the symbol type accessor.
- symbol - generate categorical symbols, as in a scatterplot.

## Axes

- axis.innerTickSize - specify the size of inner ticks.
- axis.orient - get or set the axis orientation.
- axis.outerTickSize - specify the size of outer ticks.
- axis.scale - get or set the axis scale.
- axis.tickFormat - override the tick formatting for labels.
- axis.tickPadding - specify padding between ticks and tick labels.
- axis.tickSize - specify the size of major, minor and end ticks.
- axis.ticks - control how ticks are generated for the axis.
- axis.tickValues - specify tick values explicitly.
- axis - creates or updates an axis for the given selection or transition.
- d3.svg.axis - create a new axis generator.

## Controls

- brush.clear - reset the brush extent.
- brush.empty - whether or not the brush extent is empty.
- brush.event - dispatch brush events after setting the extent.
- brush.extent - the brush's extent in zero, one or two dimensions.

- brush.on - listeners for when the brush is moved.
- brush.x - the brush's *x*-scale, for horizontal brushing.
- brush.y - the brush's *y*-scale, for vertical brushing.
- brush - apply a brush to the given selection or transition.
- d3.svg.brush - click and drag to select one- or two-dimensional regions.

## d3.time (Time)

### Time Formatting

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

### Time Scales

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

### Time Intervals

- d3.time.dayOfYear - computes the day number.
- d3.time.days - alias for day.range.
- d3.time.day - every day (12:00 AM).
- d3.time.fridayOfYear - computes the friday-based week number.
- d3.time.fridays - alias for friday.range.
- d3.time.friday - every Friday (e.g., February 5, 12:00 AM).
- d3.time.hours - alias for hour.range.
- d3.time.hour - every hour (e.g., 1:00 AM).
- d3.time.interval - a time interval in local time.
- d3.time.minutes - alias for minute.range.
- d3.time.minute - every minute (e.g., 1:02 AM).
- d3.time.mondayOfYear - computes the monday-based week number.
- d3.time.mondays - alias for monday.range.
- d3.time.monday - every Monday (e.g., February 5, 12:00 AM).
- d3.time.months - alias for month.range.
- d3.time.month - every month (e.g., February 1, 12:00 AM).
- d3.time.saturdayOfYear - computes the saturday-based week number.
- d3.time.saturdays - alias for saturday.range.
- d3.time.saturday - every Saturday (e.g., February 5, 12:00 AM).

- d3.time.seconds - alias for second.range.
- d3.time.second - every second (e.g., 1:02:03 AM).
- d3.time.sundayOfYear - computes the sunday-based week number.
- d3.time.sundays - alias for sunday.range.
- d3.time.sunday - every Sunday (e.g., February 5, 12:00 AM).
- d3.time.thursdayOfYear - computes the thursday-based week number.
- d3.time.thursdays - alias for thursday.range.
- d3.time.thursday - every Thursday (e.g., February 5, 12:00 AM).
- d3.time.tuesdayOfYear - computes the tuesday-based week number.
- d3.time.tuesdays - alias for tuesday.range.
- d3.time.tuesday - every Tuesday (e.g., February 5, 12:00 AM).
- d3.time.wednesdayOfYear - computes the wednesday-based week number.
- d3.time.wednesdays - alias for wednesday.range.
- d3.time.wednesday - every Wednesday (e.g., February 5, 12:00 AM).
- d3.time.weekOfYear - alias for sundayOfYear.
- d3.time.weeks - alias for sunday.range.
- d3.time.week - alias for sunday.
- d3.time.years - alias for year.range.
- d3.time.year - every year (e.g., January 1, 12:00 AM).
- interval.ceil - rounds up to the nearest interval.
- interval.floor - rounds down to the nearest interval.
- interval.offset - returns a date offset by some interval.
- interval.range - returns dates within the specified range.
- interval.round - rounds up or down to the nearest interval.
- interval.utc - returns the UTC-equivalent time interval.
- interval - alias for interval.floor.

# d3.layout (Layouts)

## Bundle

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

## Chord

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

## Cluster

- cluster.children - get or set the accessor function for child nodes.
- cluster.links - compute the parent-child links between tree nodes.
- cluster.nodeSize - specify a fixed size for each node.
- cluster.nodes - compute the cluster layout and return the array of nodes.
- cluster.separation - get or set the spacing function between neighboring nodes.
- cluster.size - get or set the layout size in *x* and *y*.

- cluster.sort - get or set the comparator function for sibling nodes.
- cluster - alias for cluster.nodes.
- d3.layout.cluster - cluster entities into a dendrogram.

## Force

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

## Hierarchy

- d3.layout.hierarchy - derive a custom hierarchical layout implementation.
- hierarchy.children - get or set the accessor function for child nodes.
- hierarchy.links - compute the parent-child links between tree nodes.
- hierarchy.nodes - compute the layout and return the array of nodes.
- hierarchy.revalue - recompute the hierarchy values.
- hierarchy.sort - get or set the comparator function for sibling nodes.
- hierarchy.value - get or set the value accessor function.
- hierarchy - alias for hierarchy.nodes.

## Histogram

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

## Pack

- d3.layout.pack - produce a hierarchical layout using recursive circle-packing.
- pack.children - get or set the children accessor function.
- pack.links - compute the parent-child links between tree nodes.
- pack.nodes - compute the pack layout and return the array of nodes.
- pack.padding - specify the layout padding in (approximate) pixels.
- pack.radius - specify the node radius, rather than deriving it from value.
- pack.size - specify the layout size in *x* and *y*.

- pack.sort - control the order in which sibling nodes are traversed.
- pack.value - get or set the value accessor used to size circles.
- pack - alias for pack.nodes.

### Partition

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

### Pie

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

### Stack

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

### Tree

- d3.layout.tree - position a tree of nodes tidily.
- tree.children - get or set the children accessor function.
- tree.links - compute the parent-child links between tree nodes.
- tree.nodeSize - specify a fixed size for each node.
- tree.nodes - compute the tree layout and return the array of nodes.
- tree.separation - get or set the spacing function between neighboring nodes.
- tree.size - specify the layout size in *x* and *y*.
- tree.sort - control the order in which sibling nodes are traversed.
- tree - alias for tree.nodes.

### Treemap

- d3.layout.treemap - use recursive spatial subdivision to display a tree of nodes.
- treemap.children - get or set the children accessor function.
- treemap.links - compute the parent-child links between tree nodes.
- treemap.mode - change the treemap layout algorithm.
- treemap.nodes - compute the treemap layout and return the array of nodes.

- treemap.padding - specify the padding between a parent and its children.
- treemap.round - enable or disable rounding to exact pixels.
- treemap.size - specify the layout size in *x* and *y*.
- treemap.sort - control the order in which sibling nodes are traversed.
- treemap.sticky - make the layout sticky for stable updates.
- treemap.value - get or set the value accessor used to size treemap cells.
- treemap - alias for treemap.nodes.

## d3.geo (Geography)

### Paths

- circle.angle - specify the angular radius in degrees.
- circle.origin - specify the origin in latitude and longitude.
- circle.precision - specify the precision of the piecewise circle.
- circle - generate a piecewise circle as a Polygon.
- d3.geo.area - compute the spherical area of a given feature.
- d3.geo.bounds - compute the latitude-longitude bounding box for a given feature.
- d3.geo.centroid - compute the spherical centroid of a given feature.
- d3.geo.circle - create a circle generator.
- d3.geo.distance - compute the great-arc distance between two points.
- d3.geo.graticule - create a graticule generator.
- d3.geo.interpolate - interpolate between two points along a great arc.
- d3.geo.length - compute the length of a line string or the perimeter of a polygon.
- d3.geo.path - create a new geographic path generator.
- d3.geo.rotation - create a rotation function for the specified angles [λ, φ, γ].
- graticule.extent - get or set the major & minor extents.
- graticule.lines - generate an array of LineStrings of meridians and parallels.
- graticule.majorExtent - get or set the major extent.
- graticule.majorStep - get or set the major step intervals.
- graticule.minorExtent - get or set the minor extent.
- graticule.minorStep - get or set the minor step intervals.
- graticule.outline - generate a Polygon of the graticule's extent.
- graticule.precision - get or set the latitudinal precision.
- graticule.step - get or set the major & minor step intervals.
- graticule - generate a MultiLineString of meridians and parallels.
- path.area - compute the projected area of a given feature.
- path.bounds - compute the projected bounds of a given feature.
- path.centroid - compute the projected centroid of a given feature.
- path.context - get or set the render context.
- path.pointRadius - get or set the radius to display point features.
- path.projection - get or set the geographic projection.
- path - project the specified feature and render it to the context.
- rotation.invert - inverse-rotate the given location around the sphere.
- rotation - rotate the given location around the sphere.

### Projections

- albers.parallels - get or set the projection's two standard parallels.
- d3.geo.albersUsa - a composite Albers projection for the United States.
- d3.geo.albers - the Albers equal-area conic projection.
- d3.geo.azimuthalEqualArea.raw - the raw azimuthal equal-area projection.

- d3.geo.azimuthalEqualArea - the azimuthal equal-area projection.
- d3.geo.azimuthalEquidistant.raw - the azimuthal equidistant projection.
- d3.geo.azimuthalEquidistant - the azimuthal equidistant projection.
- d3.geo.conicConformal.raw - the raw conic conformal projection.
- d3.geo.conicConformal - the conic conformal projection.
- d3.geo.conicEqualArea.raw the raw conic equal-area (a.k.a. Albers) projection.
- d3.geo.conicEqualArea the conic equal-area (a.k.a. Albers) projection.
- d3.geo.conicEquidistant.raw - the raw conic equidistant projection.
- d3.geo.conicEquidistant - the conic equidistant projection.
- d3.geo.equirectangular.raw - the raw equirectangular (plate carrée) projection.
- d3.geo.equirectangular - the equirectangular (plate carreé) projection.
- d3.geo.gnomonic.raw - the raw gnomonic projection.
- d3.geo.gnomonic - the gnomonic projection.
- d3.geo.mercator.raw - the raw Mercator projection.
- d3.geo.mercator - the spherical Mercator projection.
- d3.geo.orthographic.raw - the raw azimuthal orthographic projection.
- d3.geo.orthographic - the azimuthal orthographic projection.
- d3.geo.projectionMutator - create a standard projection from a mutable raw projection.
- d3.geo.projection - create a standard projection from a raw projection.
- d3.geo.stereographic.raw - the raw azimuthal stereographic projection.
- d3.geo.stereographic - the azimuthal stereographic projection.
- d3.geo.transverseMercator.raw - the raw transverse Mercator projection.
- projection.center - get or set the projection's center location.
- projection.clipAngle - get or set the radius of the projection's clip circle.
- projection.clipExtent - get or set the projection's viewport clip extent, in pixels.
- projection.invert - invert the projection for the specified point.
- projection.precision - get or set the precision threshold for adaptive resampling.
- projection.rotate - get or set the projection's three-axis rotation.
- projection.scale - get or set the projection's scale factor.
- projection.stream - wrap the specified stream listener, projecting input geometry.
- projection.translate - get or set the projection's translation position.
- projection - project the specified location.

### Streams

- clipExtent.extent - sets the clip extent.
- d3.geo.clipExtent - a stream transform that clips geometries to a given axis-aligned rectangle.
- d3.geo.stream - convert a GeoJSON object to a geometry stream.
- d3.geo.transform - transform streaming geometries.
- stream.lineEnd - indicate the end of a line or ring.
- stream.lineStart - indicate the start of a line or ring.
- stream.point - indicate an *x*, *y* (and optionally *z*) coordinate.
- stream.polygonEnd - indicate the end of a polygon.
- stream.polygonStart - indicate the start of a polygon.
- stream.sphere - indicate a sphere.
- transform.stream - wraps a given stream.

## d3.geom (Geometry)

### Voronoi

- d3.geom.voronoi - create a Voronoi layout with default accessors.

- voronoi.clipExtent - get or set the clip extent for the tesselation.
- voronoi.links - compute the Delaunay mesh as a network of links.
- voronoi.triangles - compute the Delaunay mesh as a triangular tessellation.
- voronoi.x - get or set the x-coordinate accessor for each point.
- voronoi.y - get or set the y-coordinate accessor for each point.
- voronoi - compute the Voronoi tessellation 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.find - find the closest point in the quadtree.
- quadtree.visit - recursively visit nodes in the quadtree.

### Polygon

- d3.geom.polygon - create a polygon from the specified array of points.
- polygon.area - compute the counterclockwise area of this polygon.
- polygon.centroid - compute the area centroid of this polygon.
- polygon.clip - clip the specified polygon to this polygon.

### Hull

- d3.geom.hull - create a convex hull layout with default accessors.
- hull - compute the convex hull for the given array of points.
- hull.x - get or set the *x*-coordinate accessor.
- hull.y - get or set the *y*-coordinate accessor.

## d3.behavior (Behaviors)

### Drag

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

### Zoom

- d3.behavior.zoom - create a zoom behavior.
- zoom.center - an optional focal point for mousewheel zooming.
- zoom.duration - get or set the dblclick transition duration.
- zoom.event - dispatch zoom events after setting the scale or translate.
- zoom.on - listeners for when the scale or translate changes.
- zoom.scaleExtent - optional limits on the scale factor.
- zoom.scale - the current scale factor.
- zoom.size - the dimensions of the viewport.
- zoom.translate - the current translate offset.
- zoom.x - an optional scale whose domain is bound to the *x* extent of the viewport.
- zoom.y - an optional scale whose domain is bound to the *y* extent of the viewport.
- zoom - apply the zoom behavior to the selected elements.