D3.js中文版api-接口文档

2014-09-23 11:47:03   阅读25477次

Api参考

此文档翻译自 [API Reference](https://github.com/mbostock/d3/wiki/API-Reference) （英语），版本为 2013-9-9 。不能保证文档的同步更新，因此，需要了解最新的开发特性，请移步英文版 [API 参考](https://github.com/mbostock/d3/wiki/API-Reference) 。

d3 库所提供的所有 API 都在 d3 命名空间下。d3 库使用[语义版本命名法（semantic versioning）](http://semver.org/)。 你可以用 d3.version 查看当前的版本信息。

**[d3 (核心部分)](https://github.com/mbostock/d3/wiki/Core)**

**[选择集](https://github.com/mbostock/d3/wiki/%E9%80%89%E6%8B%A9%E9%9B%86)**

* [d3.select](https://github.com/mbostock/d3/wiki/Selections#wiki-d3_select) - 从当前文档中选择一系列元素。
* [d3.selectAll](https://github.com/mbostock/d3/wiki/Selections#wiki-d3_selectAll) - 从当前文档中选择多项元素。
* [selection.attr](https://github.com/mbostock/d3/wiki/Selections#wiki-attr) - 设置或获取指定属性。
* [selection.classed](https://github.com/mbostock/d3/wiki/Selections#wiki-classed) - 添加或删除选定元素的 CSS 类（CSS class）。
* [selection.style](https://github.com/mbostock/d3/wiki/Selections#wiki-style) - 设置或删除 CSS 属性。style优先级高于attr。
* [selection.property](https://github.com/mbostock/d3/wiki/Selections#wiki-property) - 设置或获原生的属性值（raw property）。
* [selection.text](https://github.com/mbostock/d3/wiki/Selections#wiki-text) - 设置或获取选定元素的标签体文本内容。
* [selection.html](https://github.com/mbostock/d3/wiki/Selections#wiki-html) - 设置或获取选定元素的 HTML 内容（类似 innerHTML ）
* [selection.append](https://github.com/mbostock/d3/wiki/Selections#wiki-append) - 创建并添加新元素到选定元素后。
* [selection.insert](https://github.com/mbostock/d3/wiki/Selections#wiki-insert) - 创建并添加新元素到选定元素前。
* [selection.remove](https://github.com/mbostock/d3/wiki/Selections#wiki-remove) - 从当前文档对象中删除选定的元素。
* [selection.data](https://github.com/mbostock/d3/wiki/Selections#wiki-data) - 设置或获取一组元素的绑定数据（get or set data for a group of elements, while computing a relational join.）
* [selection.enter](https://github.com/mbostock/d3/wiki/Selections#wiki-enter) - 返回缺失元素的占位对象（placeholder），指向绑定的数据中比选定元素集多出的一部分元素。
* [selection.exit](https://github.com/mbostock/d3/wiki/Selections#wiki-exit) - 返回多余元素的元素集，即选择元素中比绑定数据多出的一部分。(关于data, enter, exit原理的[示例1](http://bost.ocks.org/mike/join/), [示例2](http://bl.ocks.org/mbostock/3808218), [示例3](http://bl.ocks.org/mbostock/5779690))
* [selection.datum](https://github.com/mbostock/d3/wiki/Selections#wiki-datum) - 设置或获取单独元素的数据，不进行关联。（get or set data for individual elements, without computing a join.）
* [selection.filter](https://github.com/mbostock/d3/wiki/Selections#wiki-filter) - 根据绑定的数据过滤选择集。
* [selection.sort](https://github.com/mbostock/d3/wiki/Selections#wiki-sort) - 根据绑定的数据对选择的元素进行排序。
* [selection.order](https://github.com/mbostock/d3/wiki/Selections#wiki-order) - 对文档中的元素重排序以匹配选择集。
* [selection.on](https://github.com/mbostock/d3/wiki/Selections#wiki-on) - 添加或删除事件监听器。
* [selection.transition](https://github.com/mbostock/d3/wiki/Selections#wiki-transition) - 启动一个过渡效果（返回 Transition 对象），可以理解为动画。
* [selection.interrupt](https://github.com/mbostock/d3/wiki/Selections#wiki-interrupt) - 立即停止所有正在进行的动画动作。
* [selection.each](https://github.com/mbostock/d3/wiki/Selections#wiki-each) - 为每个选择的元素集调用指定的函数。
* [selection.call](https://github.com/mbostock/d3/wiki/Selections#wiki-call) - 为当前选择的元素集调用指定的函数。
* [selection.empty](https://github.com/mbostock/d3/wiki/Selections#wiki-empty) - 测试选择集是否为空。
* [selection.node](https://github.com/mbostock/d3/wiki/Selections#wiki-node) - 返回选择集中的第一个元素。
* [selection.size](https://github.com/mbostock/d3/wiki/Selections#wiki-size) - 返回选择集中的元素个数。
* [selection.select](https://github.com/mbostock/d3/wiki/Selections#wiki-select) - 选择所选的元素中的第一个子元素组成新的选择集。
* [selection.selectAll](https://github.com/mbostock/d3/wiki/Selections#wiki-selectAll) - 选择所选的元素中的多个子元素组成新的选择集。
* [d3.selection](https://github.com/mbostock/d3/wiki/Selections#wiki-d3_selection) - 选择集对象原型（可通过 d3.selection.prototype 为选择集增强功能）。
* [d3.event](https://github.com/mbostock/d3/wiki/Selections#wiki-d3_event) - 获取当前交互的用户事件。
* [d3.mouse](https://github.com/mbostock/d3/wiki/Selections#wiki-d3_mouse) - 获取鼠标的相对某元素的坐标。
* [d3.touches](https://github.com/mbostock/d3/wiki/Selections#wiki-d3_touches) - 获取相对某元素的触控点坐标。

**[过渡效果](https://github.com/mbostock/d3/wiki/Transitions)**

* [d3.transition](https://github.com/mbostock/d3/wiki/Transitions#wiki-d3_transition) - 开始一个动画过渡。[简单教程](http://bost.ocks.org/mike/transition/)
* [transition.delay](https://github.com/mbostock/d3/wiki/Transitions#wiki-delay) - 指定每个元素过渡的延迟时间（单位：毫秒ms）。
* [transition.duration](https://github.com/mbostock/d3/wiki/Transitions#wiki-duration) - 指定每个元素过渡的持续时间（单位：毫秒ms）。
* [transition.ease](https://github.com/mbostock/d3/wiki/Transitions#wiki-ease) - 指定过渡的缓冲函数。
* [transition.attr](https://github.com/mbostock/d3/wiki/Transitions#wiki-attr) - 平滑过渡到新的attr属性值（起始属性值为当前属性）。
* [transition.attrTween](https://github.com/mbostock/d3/wiki/Transitions#wiki-attrTween) - 在不同attr属性值之间平滑过渡（起始属性值可在过渡函数中设置,甚至整个过渡函数都可以自定义）。
* [transition.style](https://github.com/mbostock/d3/wiki/Transitions#wiki-style) - 平滑过渡到新的style属性值。
* [transition.styleTween](https://github.com/mbostock/d3/wiki/Transitions#wiki-styleTween) - 在不同style属性值之间平滑过渡。
* [transition.text](https://github.com/mbostock/d3/wiki/Transitions#wiki-text) - 在过渡开始时设置文本内容。
* [transition.tween](https://github.com/mbostock/d3/wiki/Transitions#wiki-tween) - 使某个属性过渡到一个新的属性值，该属性可以是非attr或非style属性，比如text。
* [transition.select](https://github.com/mbostock/d3/wiki/Transitions#wiki-select) - 选择每个当前元素的某个子元素进行过渡。
* [transition.selectAll](https://github.com/mbostock/d3/wiki/Transitions#wiki-selectAll) - 选择每个当前元素的多个子元素进行过渡。
* [transition.filter](https://github.com/mbostock/d3/wiki/Transitions#wiki-filter) - 通过数据筛选出当前元素中的部分元素进行过渡。
* [transition.transition](https://github.com/mbostock/d3/wiki/Transitions#wiki-transition) - 当前过渡结束后开始新的过渡。
* [transition.remove](https://github.com/mbostock/d3/wiki/Transitions#wiki-remove) - 过渡结束后移除当前元素。
* [transition.empty](https://github.com/mbostock/d3/wiki/Transitions#wiki-empty) - 如果过渡为空就返回true。如果当前元素中没有非null元素，则此过渡为空。
* [transition.node](https://github.com/mbostock/d3/wiki/Transitions#wiki-node) - 返回过渡中的第一个元素。
* [transition.size](https://github.com/mbostock/d3/wiki/Transitions#wiki-size) - 返回过渡中当前元素的数量。
* [transition.each](https://github.com/mbostock/d3/wiki/Transitions#wiki-each) - 遍历每个元素执行操作。不指定触发类型时，立即执行操作。当指定触发类型为'start'或'end'时,会在过渡开始或结束时执行操作。
* [transition.call](https://github.com/mbostock/d3/wiki/Transitions#wiki-call) - 以当前过渡为this执行某个函数。
* [d3.ease](https://github.com/mbostock/d3/wiki/Transitions#wiki-d3_ease) - 定制过渡的缓冲函数。
* [ease](https://github.com/mbostock/d3/wiki/Transitions#wiki-_ease) - 缓冲函数。缓冲函数可让动画效果更自然，比如elastic缓冲函数可用以模拟弹性物体的运动。是一种插值函数的特例。
* [d3.timer](https://github.com/mbostock/d3/wiki/Transitions#wiki-d3_timer) - 开始一个定制的动画计时。功能类似于setTimeout，但内部用requestAnimationFrame实现，更高效。
* [d3.timer.flush](https://github.com/mbostock/d3/wiki/Transitions#wiki-d3_timer_flush) - 立刻执行当前没有延迟的计时。可用于处理闪屏问题。
* [d3.interpolate](https://github.com/mbostock/d3/wiki/Transitions#wiki-d3_interpolate) - 生成一个插值函数，在两个参数间插值。差值函数的类型会根据输入参数的类型（数字、字符串、颜色等）而自动选择。
* [interpolate](https://github.com/mbostock/d3/wiki/Transitions#wiki-_interpolate) - 插值函数。输入参数在[0, 1]之间。
* [d3.interpolateNumber](https://github.com/mbostock/d3/wiki/Transitions#wiki-d3_interpolateNumber) - 在两个数字间插值。
* [d3.interpolateRound](https://github.com/mbostock/d3/wiki/Transitions#wiki-d3_interpolateRound) - 在两个数字间插值，返回值会四舍五入取整。
* [d3.interpolateString](https://github.com/mbostock/d3/wiki/Transitions#wiki-d3_interpolateString) - 在两个字符串间插值。解析字符串中的数字，对应的数字会插值。
* [d3.interpolateRgb](https://github.com/mbostock/d3/wiki/Transitions#wiki-d3_interpolateRgb) - 在两个RGB颜色间插值。
* [d3.interpolateHsl](https://github.com/mbostock/d3/wiki/Transitions#wiki-d3_interpolateHsl) - 在两个HSL颜色间插值。
* [d3.interpolateLab](https://github.com/mbostock/d3/wiki/Transitions#wiki-d3_interpolateLab) - 在两个L\*a\*b\*颜色间插值。
* [d3.interpolateHcl](https://github.com/mbostock/d3/wiki/Transitions#wiki-d3_interpolateHcl) - 在两个HCL颜色间插值。
* [d3.interpolateArray](https://github.com/mbostock/d3/wiki/Transitions#wiki-d3_interpolateArray) - 在两个数列间插值。d3.interpolateArray( [0, 1], [1, 10, 100] )(0.5); // returns [0.5, 5.5, 100]
* [d3.interpolateObject](https://github.com/mbostock/d3/wiki/Transitions#wiki-d3_interpolateObject) - 在两个object间插值。d3.interpolateArray( {x: 0, y: 1}, {x: 1, y: 10, z: 100} )(0.5); // returns {x: 0.5, y: 5.5, z: 100}
* [d3.interpolateTransform](https://github.com/mbostock/d3/wiki/Transitions#wiki-d3_interpolateTransform) - 在两个2D仿射变换间插值。
* [d3.interpolateZoom](https://github.com/mbostock/d3/wiki/Transitions#wiki-d3_interpolateZoom) - 在两个点之间平滑地缩放平移。[示例](http://bl.ocks.org/mbostock/3828981)
* [d3.interpolators](https://github.com/mbostock/d3/wiki/Transitions#wiki-d3_interpolators) - 添加一个自定义的插值函数.

**[数据操作(Working with Arrays)](https://github.com/mbostock/d3/wiki/Arrays)**

* [d3.ascending](https://github.com/mbostock/d3/wiki/Arrays#wiki-d3_ascending) - 升序排序函数.
* [d3.descending](https://github.com/mbostock/d3/wiki/Arrays#wiki-d3_descending) - 降序排序函数.
* [d3.min](https://github.com/mbostock/d3/wiki/Arrays#wiki-d3_min) - 获取数组中的最小值.
* [d3.max](https://github.com/mbostock/d3/wiki/Arrays#wiki-d3_max) - 获取数组中的最大值.
* [d3.extent](https://github.com/mbostock/d3/wiki/Arrays#wiki-d3_extent) - 获取数组的范围(最小值和最大值).
* [d3.sum](https://github.com/mbostock/d3/wiki/Arrays#wiki-d3_sum) - 获取数组中数字之和.
* [d3.mean](https://github.com/mbostock/d3/wiki/Arrays#wiki-d3_mean) -获取数组中数字的算术平均值.
* [d3.median](https://github.com/mbostock/d3/wiki/Arrays#wiki-d3_median) - 获取数组中数字的中位数 (相当于 0.5-quantile的值).
* [d3.quantile](https://github.com/mbostock/d3/wiki/Arrays#wiki-d3_quantile) - 获取排好序的数组的一个分位数(quantile).
* [d3.bisect](https://github.com/mbostock/d3/wiki/Arrays#wiki-d3_bisect) - 通过二分法获取某个数在排好序的数组中的插入位置(同d3.bisectRight).
* [d3.bisectRight](https://github.com/mbostock/d3/wiki/Arrays#wiki-d3_bisectRight) - 获取某个数在排好序的数组中的插入位置(相等的值归入右边).
* [d3.bisectLeft](https://github.com/mbostock/d3/wiki/Arrays#wiki-d3_bisectLeft) - 获取某个数在排好序的数组中的插入位置(相等的值归入左边).
* [d3.bisector](https://github.com/mbostock/d3/wiki/Arrays#wiki-d3_bisector) - 自定义一个二分函数.
* [d3.shuffle](https://github.com/mbostock/d3/wiki/Arrays#wiki-d3_shuffle) - 洗牌，随机排列数组中的元素.
* [d3.permute](https://github.com/mbostock/d3/wiki/Arrays#wiki-d3_permute) - 以指定顺序排列数组中的元素.
* [d3.zip](https://github.com/mbostock/d3/wiki/Arrays#wiki-d3_zip) - 将多个数组合并成一个数组的数组，新数组的的第i个元素是原来各个数组中第i个元素组成的数组.
* [d3.transpose](https://github.com/mbostock/d3/wiki/Arrays#wiki-d3_transpose) - 矩阵转置，通过d3.zip实现.
* [d3.pairs](https://github.com/mbostock/d3/wiki/Arrays#wiki-d3_pairs) - 返回临近元素对的数组，d3.pairs([1, 2, 3, 4]); // returns [ [1, 2], [2, 3], [3, 4] ].
* [d3.keys](https://github.com/mbostock/d3/wiki/Arrays#wiki-d3_keys) - 返回关联数组(哈希表、json、object对象)的key组成的数组.
* [d3.values](https://github.com/mbostock/d3/wiki/Arrays#wiki-d3_values) - 返回关联数组的value组成的数组.
* [d3.entries](https://github.com/mbostock/d3/wiki/Arrays#wiki-d3_entries) - 返回关联数组的key-value实体组成的数组, d3.entries({ foo: 42 }); // returns [{key: "foo", value: 42}].
* [d3.merge](https://github.com/mbostock/d3/wiki/Arrays#wiki-d3_merge) - 将多个数组连成一个，类似于原生方法concat. d3.merge([ [1], [2, 3] ]); // returns [1, 2, 3].
* [d3.range](https://github.com/mbostock/d3/wiki/Arrays#wiki-d3_range) - 获得一个数列. d3.range([start, ]stop[, step])
* [d3.nest](https://github.com/mbostock/d3/wiki/Arrays#wiki-d3_nest) - 获得一个nest对象，将数组组织成层级结构. 示例：<http://bl.ocks.org/phoebebright/raw/3176159/>
* [nest.key](https://github.com/mbostock/d3/wiki/Arrays#wiki-nest_key) - 为nest层级结构增加一个层级.
* [nest.sortKeys](https://github.com/mbostock/d3/wiki/Arrays#wiki-nest_sortKeys) - 将当前的nest层级结构按key排序.
* [nest.sortValues](https://github.com/mbostock/d3/wiki/Arrays#wiki-nest_sortValues) - 将叶nest层级按value排序.
* [nest.rollup](https://github.com/mbostock/d3/wiki/Arrays#wiki-nest_rollup) - 设置修改叶节点值的函数.
* [nest.map](https://github.com/mbostock/d3/wiki/Arrays#wiki-nest_map) - 执行nest操作, 返回一个关联数组(json).
* [nest.entries](https://github.com/mbostock/d3/wiki/Arrays#wiki-nest_entries) - 执行nest操作, 返回一个key-value数组. 如果nest.map返回的结果类似于{ foo: 42 }, 则nest.entries返回的结果类似于[{key: "foo", value: 42}].
* [d3.map](https://github.com/mbostock/d3/wiki/Arrays#wiki-d3_map) - 将javascript的object转化为hash,屏蔽了object的原型链功能导致的与hash不一致的问题。
* [map.has](https://github.com/mbostock/d3/wiki/Arrays#wiki-map_has) - map有某个key就返回true.
* [map.get](https://github.com/mbostock/d3/wiki/Arrays#wiki-map_get) - 返回map中某个key对应的value.
* [map.set](https://github.com/mbostock/d3/wiki/Arrays#wiki-map_set) - 设置map中某个key对应的value.
* [map.remove](https://github.com/mbostock/d3/wiki/Arrays#wiki-map_remove) - 删除map中的某个key.
* [map.keys](https://github.com/mbostock/d3/wiki/Arrays#wiki-map_keys) - 返回map中所有key组成的数组.
* [map.values](https://github.com/mbostock/d3/wiki/Arrays#wiki-map_values) - 返回map中所有value组成的数组.
* [map.entries](https://github.com/mbostock/d3/wiki/Arrays#wiki-map_entries) - 返回map中所有entry（key-value键值对）组成的数组.类似于{ foo: 42 }转化成[{key: "foo", value: 42}]
* [map.forEach](https://github.com/mbostock/d3/wiki/Arrays#wiki-map_forEach) - 对map中每一个entry执行某个函数.
* [d3.set](https://github.com/mbostock/d3/wiki/Arrays#wiki-d3_set) - 将javascript的array转化为set,屏蔽了array的object原型链功能导致的与set不一致的问题。set中的value是array中每个值转换成字符串的结果。set中的value是去重过的。
* [set.has](https://github.com/mbostock/d3/wiki/Arrays#wiki-set_has) - 返回set中是否含有某个value.
* [set.add](https://github.com/mbostock/d3/wiki/Arrays#wiki-set_add) - 添加某个value.
* [set.remove](https://github.com/mbostock/d3/wiki/Arrays#wiki-set_remove) - 删除某个value.
* [set.values](https://github.com/mbostock/d3/wiki/Arrays#wiki-set_values) - 返回set中的值组成的数组.set中的value是去重过的.
* [set.forEach](https://github.com/mbostock/d3/wiki/Arrays#wiki-set_forEach) - 对set中每一个value执行某个函数.

**[Math](https://github.com/mbostock/d3/wiki/Math)**

* [d3.random.normal](https://github.com/mbostock/d3/wiki/Math#wiki-random_normal) - 利用正态分布产生一个随机数.
* [d3.random.logNormal](https://github.com/mbostock/d3/wiki/Math#wiki-random_logNormal) - 利用对数正态分布产生一个随机数.
* [d3.random.irwinHall](https://github.com/mbostock/d3/wiki/Math#wiki-random_irwinHall) - 利用Irwin–Hall分布（简单可行并且容易编程的正态分布实现方法）产生一个随机数.
* [d3.transform](https://github.com/mbostock/d3/wiki/Math#wiki-transform) - 将svg的tranform格式转化为标准的2D转换矩阵字符串格式.

**[载入外部资源(Loading External Resources)](https://github.com/mbostock/d3/wiki/Requests)**

* [d3.xhr](https://github.com/mbostock/d3/wiki/Requests#wiki-d3_xhr) - 发起XMLHttpRequest请求获取资源。
* [xhr.header](https://github.com/mbostock/d3/wiki/Requests#wiki-header) - 设置 request header。
* [xhr.mimeType](https://github.com/mbostock/d3/wiki/Requests#wiki-mimeType) - 设置 Accept request header，并重写 response MIME type。
* [xhr.response](https://github.com/mbostock/d3/wiki/Requests#wiki-response) - 设置response返回值转化函数。如 function(request) { return JSON.parse(request.responseText); }
* [xhr.get](https://github.com/mbostock/d3/wiki/Requests#wiki-get) - 发起GET请求。
* [xhr.post](https://github.com/mbostock/d3/wiki/Requests#wiki-post) - 发起POST请求。
* [xhr.send](https://github.com/mbostock/d3/wiki/Requests#wiki-send) - 以指定的方法和数据发起请求。
* [xhr.abort](https://github.com/mbostock/d3/wiki/Requests#wiki-abort) - 终止当前请求。
* [xhr.on](https://github.com/mbostock/d3/wiki/Requests#wiki-on) - 为请求添加"beforesend", "progress", "load" 或 "error" 等事件监听器。
* [d3.text](https://github.com/mbostock/d3/wiki/Requests#wiki-d3_text) - 请求一个text文件。
* [d3.json](https://github.com/mbostock/d3/wiki/Requests#wiki-d3_json) - 请求一个JSON。
* [d3.html](https://github.com/mbostock/d3/wiki/Requests#wiki-d3_html) - 请求一个html文本片段。
* [d3.xml](https://github.com/mbostock/d3/wiki/Requests#wiki-d3_xml) - 请求一个XML文本片段。
* [d3.csv](https://github.com/mbostock/d3/wiki/CSV) - 请求一个CSV(comma-separated values, 逗号分隔值)文件。
* [d3.tsv](https://github.com/mbostock/d3/wiki/CSV#wiki-tsv) - 请求一个TSV(tab-separated values, tab分隔值)文件。

**[字符串格式化(String Formatting)](https://github.com/mbostock/d3/wiki/Formatting)**

* [d3.format](https://github.com/mbostock/d3/wiki/Formatting#wiki-d3_format) - 将数字转化成指定格式的字符串。转化的格式非常丰富，且非常智能。
* [d3.formatPrefix](https://github.com/mbostock/d3/wiki/Formatting#wiki-d3_formatPrefix) - 以指定的值和精度获得一个[SI prefix]对象。这个函数可用来自动判断数据的量级， 如K(千)，M(百万)等等。示例: var prefix = d3.formatPrefix(1.21e9); console.log(prefix.symbol); // "G"; console.log(prefix.scale(1.21e9)); // 1.21
* [d3.requote](https://github.com/mbostock/d3/wiki/Formatting#wiki-d3_requote) - 将字符串转义成可在正则表达式中使用的格式。如 d3.requote('$'); // return "\$"
* [d3.round](https://github.com/mbostock/d3/wiki/Formatting#wiki-d3_round) - 设置某个数按小数点后多少位取整。与toFixed()类似，但返回格式为number。 如 d3.round(1.23); // return 1; d3.round(1.23, 1); // return 1.2; d3.round(1.25, 1); // return 1.3

**[CSV 格式化 (d3.csv)](https://github.com/mbostock/d3/wiki/CSV)**

* [d3.csv](https://github.com/mbostock/d3/wiki/CSV#wiki-csv) - 获取一个CSV (comma-separated values, 冒号分隔值)文件。
* [d3.csv.parse](https://github.com/mbostock/d3/wiki/CSV#wiki-parse) - 将CSV文件字符串转化成object的数组，object的key由第一行决定。如： [{"Year": "1997", "Length": "2.34"}, {"Year": "2000", "Length": "2.38"}]
* [d3.csv.parseRows](https://github.com/mbostock/d3/wiki/CSV#wiki-parseRows) - 将CSV文件字符串转化成数组的数组。如： [ ["Year", "Length"],["1997", "2.34"],["2000", "2.38"] ]
* [d3.csv.format](https://github.com/mbostock/d3/wiki/CSV#wiki-format) - 将object的数组转化成CSV文件字符串，是d3.csv.parse的逆操作。
* [d3.csv.formatRows](https://github.com/mbostock/d3/wiki/CSV#wiki-formatRows) - 将数组的数组转化成CSV文件字符串，是d3.csv.parseRows的逆操作。
* [d3.tsv](https://github.com/mbostock/d3/wiki/CSV#wiki-tsv) - 获取一个TSV (tab-separated values, tab分隔值)文件。
* [d3.tsv.parse](https://github.com/mbostock/d3/wiki/CSV#wiki-tsv_parse) - 类似于d3.csv.parse。
* [d3.tsv.parseRows](https://github.com/mbostock/d3/wiki/CSV#wiki-tsv_parseRows) - 类似于d3.csv.parseRows。
* [d3.tsv.format](https://github.com/mbostock/d3/wiki/CSV#wiki-tsv_format) - 类似于d3.csv.format。
* [d3.tsv.formatRows](https://github.com/mbostock/d3/wiki/CSV#wiki-tsv_formatRows) - 类似于d3.csv.formatRows。
* [d3.dsv](https://github.com/mbostock/d3/wiki/CSV#wiki-dsv) - 创建一个类似于d3.csv的文件处理对象，可以自定义分隔符和mime type。如：var dsv = d3.dsv("|", "text/plain");

**[Colors](https://github.com/mbostock/d3/wiki/Colors)**

* [d3.rgb](https://github.com/mbostock/d3/wiki/Colors#wiki-d3_rgb) - 指定一种颜色，创建一个RGB颜色对象。支持多种颜色格式的输入。
* [rgb.brighter](https://github.com/mbostock/d3/wiki/Colors#wiki-rgb_brighter) - 增强颜色的亮度，变化幅度由参数决定。
* [rgb.darker](https://github.com/mbostock/d3/wiki/Colors#wiki-rgb_darker) - 减弱颜色的亮度，变化幅度由参数决定。
* [rgb.hsl](https://github.com/mbostock/d3/wiki/Colors#wiki-rgb_hsl) - 将RGB颜色对象转化成HSL颜色对象。
* [rgb.toString](https://github.com/mbostock/d3/wiki/Colors#wiki-rgb_toString) - RGB颜色转化为字符串格式。
* [d3.hsl](https://github.com/mbostock/d3/wiki/Colors#wiki-d3_hsl) - 创建一个HSL颜色对象。支持多种颜色格式的输入。
* [hsl.brighter](https://github.com/mbostock/d3/wiki/Colors#wiki-hsl_brighter) - 增强颜色的亮度，变化幅度由参数决定。
* [hsl.darker](https://github.com/mbostock/d3/wiki/Colors#wiki-hsl_darker) - 减弱颜色的亮度，变化幅度由参数决定。
* [hsl.rgb](https://github.com/mbostock/d3/wiki/Colors#wiki-hsl_rgb) - 将HSL颜色对象转化成RGB颜色对象。
* [hsl.toString](https://github.com/mbostock/d3/wiki/Colors#wiki-hsl_toString) - HSL颜色转化为字符串格式。
* [d3.lab](https://github.com/mbostock/d3/wiki/Colors#wiki-d3_lab) - 创建一个Lab颜色对象。支持多种颜色格式的输入。
* [lab.brighter](https://github.com/mbostock/d3/wiki/Colors#wiki-lab_brighter) - 增强颜色的亮度，变化幅度由参数决定。
* [lab.darker](https://github.com/mbostock/d3/wiki/Colors#wiki-lab_darker) - 减弱颜色的亮度，变化幅度由参数决定。
* [lab.rgb](https://github.com/mbostock/d3/wiki/Colors#wiki-lab_rgb) - 将Lab颜色对象转化成RGB颜色对象。
* [lab.toString](https://github.com/mbostock/d3/wiki/Colors#wiki-lab_toString) - Lab颜色转化为字符串格式。
* [d3.hcl](https://github.com/mbostock/d3/wiki/Colors#wiki-d3_hcl) - 创建一个HCL颜色对象。支持多种颜色格式的输入。
* [hcl.brighter](https://github.com/mbostock/d3/wiki/Colors#wiki-hcl_brighter) - 增强颜色的亮度，变化幅度由参数决定。
* [hcl.darker](https://github.com/mbostock/d3/wiki/Colors#wiki-hcl_darker) - 减弱颜色的亮度，变化幅度由参数决定。
* [hcl.rgb](https://github.com/mbostock/d3/wiki/Colors#wiki-hcl_rgb) - 将HCL颜色对象转化成RGB颜色对象。
* [hcl.toString](https://github.com/mbostock/d3/wiki/Colors#wiki-hcl_toString) - HCL颜色转化为字符串格式。

**[命名空间](https://github.com/mbostock/d3/wiki/Namespaces)**

* [d3.ns.prefix](https://github.com/mbostock/d3/wiki/Namespaces#wiki-prefix) - 获取或扩展已知的XML命名空间。
* [d3.ns.qualify](https://github.com/mbostock/d3/wiki/Namespaces#wiki-qualify) - 验证命名空间前缀是否存在, 如"xlink:href"中xlink是已知的命名空间。

**[内部方法（Internals）](https://github.com/mbostock/d3/wiki/Internals)**

* [d3.functor](https://github.com/mbostock/d3/wiki/Internals#wiki-functor) - 函数化。将非函数变量转化为只返回该变量值的函数。输入函数，则返回原函数；输入值，则返回一个函数，该函数只返回原值。
* [d3.rebind](https://github.com/mbostock/d3/wiki/Internals#wiki-rebind) - 将一个对象的方法绑定到另一个对象上。
* [d3.dispatch](https://github.com/mbostock/d3/wiki/Internals#wiki-d3_dispatch) - 创建一个定制的事件。
* [dispatch.on](https://github.com/mbostock/d3/wiki/Internals#wiki-dispatch_on) - 添加或移除一个事件监听器。对一个事件可添加多个监听器。
* [dispatch.type](https://github.com/mbostock/d3/wiki/Internals#wiki-_dispatch) - 触发事件。其中‘type’为要触发的事件的名称。

**[d3.scale(Scales)](https://github.com/mbostock/d3/wiki/Scales)**

**[定量变换(Quantitative)](https://github.com/mbostock/d3/wiki/Quantitative-Scales" \l "wiki-quantitative)**

* [d3.scale.linear](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-linear) - 创建一个线性定量变换。（建议参考源码以深入理解各种变换。）
* [linear](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-_linear) - 输入一个定义域的值，返回一个值域的值。
* [linear.invert](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-linear_invert) - 反变换，输入值域值返回定义域值。
* [linear.domain](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-linear_domain) - get或set定义域。
* [linear.range](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-linear_range) - get或set值域。
* [linear.rangeRound](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-linear_rangeRound) - 设置值域，并对结果取整。
* [linear.interpolate](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-linear_interpolate) - get或set变换的插值函数，如将默认的线性插值函数替换成取整的线性插值函数d3\_interpolateRound。
* [linear.clamp](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-linear_clamp) - 设置值域是否闭合，默认不闭合。当值域闭合时，如果插值结果在值域之外，会取值域的边界值。如值域为[1, 2],插值函数的计算结果为3，如果不闭合，最终结果为3；如果闭合，最终结果为2。
* [linear.nice](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-linear_nice) - 扩展定义域范围使定义域更规整。如[0.20147987687960267, 0.996679553296417] 变成 [0.2, 1]。
* [linear.ticks](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-linear_ticks) - 从定义域中取出有代表性的值。通常用于坐标轴刻度的选取。
* [linear.tickFormat](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-linear_tickFormat) - 获取格式转化函数，通常用于坐标轴刻度的格式转化。如：var x = d3.scale.linear().domain([-1, 1]); console.log(x.ticks(5).map(x.tickFormat(5, "+%"))); // ["-100%", "-50%", "+0%", "+50%", "+100%"]
* [linear.copy](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-linear_copy) - 从已有的变换中复制出一个变换。
* [d3.scale.sqrt](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-sqrt) - 创建一个求平方根的定量转换。
* [d3.scale.pow](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-pow) - 创建一个指数变换。（可参考linear对应函数的注释）
* [pow](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-_pow) - 输入一个定义域的值，返回一个值域的值。
* [pow.invert](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-pow_invert) - 反变换，输入值域值返回定义域值。
* [pow.domain](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-pow_domain) - get或set定义域。
* [pow.range](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-pow_range) - get或set值域。
* [pow.rangeRound](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-pow_rangeRound) - 设置值域，并对结果取整。
* [pow.interpolate](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-pow_interpolate) - get或set变换的插值函数。
* [pow.clamp](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-pow_clamp) - 设置值域是否闭合，默认不闭合。
* [pow.nice](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-pow_nice) - 扩展定义域范围使定义域更规整。
* [pow.ticks](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-pow_ticks) - 从定义域中取出有代表性的值。通常用于坐标轴刻度的选取。
* [pow.tickFormat](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-pow_tickFormat) - 获取格式转化函数，通常用于坐标轴刻度的格式转化。
* [pow.exponent](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-pow_exponent) - get或set指数的幂次。默认为1次幂。
* [pow.copy](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-pow_copy) - 从已有的变换中复制出一个变换。
* [d3.scale.log](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-log) - 创建一个对数变换。（可参考linear对应函数的注释）
* [log](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-_log) - 输入一个定义域的值，返回一个值域的值。
* [log.invert](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-log_invert) - 反变换，输入值域值返回定义域值。
* [log.domain](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-log_domain) - get或set定义域。
* [log.range](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-log_range) - get或set值域。
* [log.rangeRound](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-log_rangeRound) - 设置值域，并对结果取整。
* [log.interpolate](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-log_interpolate) - get或set变换的插值函数。
* [log.clamp](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-log_clamp) - 设置值域是否闭合，默认不闭合。
* [log.nice](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-log_nice) - 扩展定义域范围使定义域更规整。
* [log.ticks](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-log_ticks) - 从定义域中取出有代表性的值。通常用于坐标轴刻度的选取。
* [log.tickFormat](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-log_tickFormat) - 获取格式转化函数，通常用于坐标轴刻度的格式转化。
* [log.copy](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-log_copy) - 从已有的变换中复制出一个变换。
* [d3.scale.quantize](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-quantize) - 创建一个quantize线性变换,定义域为一个数值区间，值域为几个离散值。
* [quantize](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-_quantize) - 输入数值，返回离散值。如： var q = d3.scale.quantize().domain([0, 1]).range(['a', 'b', 'c']); //q(0.3) === 'a', q(0.4) === 'b', q(0.6) === 'b', q(0.7) ==='c;
* [quantize.invertExtent](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-quantize_invertExtent) - 返回得到某个离散值的值域范围。 // q.invertExtent('a') 的结果为 [0, 0.3333333333333333]
* [quantize.domain](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-quantize_domain) - get或set变换的定义域。
* [quantize.range](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-quantize_range) - get或set变换的值域。
* [quantize.copy](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-quantize_copy) - 从已有的变换中复制出一个变换。
* [d3.scale.threshold](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-threshold) - 构建一个threshold(阈值)线性变换。定义域为分隔值数值序列，值域为离散值。它与quantize的区别是quantize指定的值域为一个区间，然后均分这个区间为多个小区间，以对应各离散值。threshold则指定各小区间的边界分隔值。示例: var t = d3.scale.threshold().domain([0, 1]).range(['a', 'b', 'c']); t(-1) === 'a'; t(0) === 'b'; t(0.5) === 'b'; t(1) === 'c'; t(1000) === 'c'; t.invertExtent('a'); //returns [undefined, 0] t.invertExtent('b'); //returns [0, 1] t.invertExtent('c'); //returns [1, undefined]
* [threshold](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-_threshold) - 输入数值，返回离散值。
* [threshold.invertExtent](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-threshold_invertExtent) - 输入离散值，返回数值。
* [threshold.domain](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-threshold_domain) - get或set变换的定义域。
* [threshold.range](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-threshold_range) - get或set变换的值域。
* [threshold.copy](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-threshold_copy) - 从已有的变换中复制出一个变换。
* [d3.scale.quantile](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-quantile) - 构建一个quantile线性变换。使用方法与quantize完全类似，区别是quantile根据中位数来分隔区间，quantize根据算数平均值来分隔区间。[example](http://stackoverflow.com/questions/19258996/what-is-the-difference-between-d3-scale-quantize-and-d3-scale-quantile)
* [quantile](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-_quantile) - 输入数值，返回离散值。
* [quantile.invertExtent](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-quantile_invertExtent) - 输入离散值，返回数值。
* [quantile.domain](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-quantile_domain) - get或set变换的定义域。
* [quantile.range](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-quantile_range) - get或set变换的值域。
* [quantile.quantiles](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-quantile_quantiles) - 获得quantile变换的分隔值。示例： var q = d3.scale.quantile().domain([0, 1]).range(['a', 'b', 'c']); q.quantiles() returns [0.33333333333333326, 0.6666666666666665]
* [quantile.copy](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-quantile_copy) - 从已有的变换中复制出一个变换。
* [d3.scale.identity](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-identity) - 构建一个identity线性变换。特殊的linear线性变换，此变换定义域和值域相同，只在一些d3内部的axis或brush模块中用到。
* [identity](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-_identity) - identity线性变换函数。返回输入值。
* [identity.invert](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-_identity) - 和identity函数相同，返回输入值。
* [identity.domain](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-identity_domain) - get或set变换的定义域。
* [identity.range](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-identity_domain) - get或set变换的值域。
* [identity.ticks](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-identity_ticks) - 从定义域中取出有代表性的值。通常用于坐标轴刻度的选取。
* [identity.tickFormat](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-identity_tickFormat) - 获取格式转化函数，通常用于坐标轴刻度的格式转化。
* [identity.copy](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-identity_copy) - 从已有的变换中复制出一个变换。

**[序数变换（Ordinal）](https://github.com/mbostock/d3/wiki/Ordinal-Scales" \l "wiki-ordinal)**

* [d3.scale.ordinal](https://github.com/mbostock/d3/wiki/Ordinal-Scales#wiki-ordinal) - 构建一个ordinal变换对象。ordinal变换的输入定义域和输出值域都是离散的。而quantitative变换的输入定义域是连续的，这是两者最大的不同。
* [ordinal](https://github.com/mbostock/d3/wiki/Ordinal-Scales#wiki-_ordinal) - 输入一个离散值，返回一个离散值。不在当前定义域中的输入值会自动加入定义域。
* [ordinal.domain](https://github.com/mbostock/d3/wiki/Ordinal-Scales#wiki-ordinal_domain) - get或set变换的定义域。
* [ordinal.range](https://github.com/mbostock/d3/wiki/Ordinal-Scales#wiki-ordinal_range) - get或set变换的值域。
* [ordinal.rangePoints](https://github.com/mbostock/d3/wiki/Ordinal-Scales#wiki-ordinal_rangePoints) - 用几个离散点来分割一个连续的区间。详情请看链接中的图例。
* [ordinal.rangeBands](https://github.com/mbostock/d3/wiki/Ordinal-Scales#wiki-ordinal_rangeBands) - 用几个离散区间来分割一个连续的区间。详情请看链接中的图例。
* [ordinal.rangeRoundBands](https://github.com/mbostock/d3/wiki/Ordinal-Scales#wiki-ordinal_rangeRoundBands) - 用几个离散区间来分割一个连续的区间，区间边界和宽度会取整。详情请看链接中的图例。
* [ordinal.rangeBand](https://github.com/mbostock/d3/wiki/Ordinal-Scales#wiki-ordinal_rangeBand) - 获取离散区间的宽度。
* [ordinal.rangeExtent](https://github.com/mbostock/d3/wiki/Ordinal-Scales#wiki-ordinal_rangeExtent) - 获取输出域的最小最大值。
* [ordinal.copy](https://github.com/mbostock/d3/wiki/Ordinal-Scales#wiki-ordinal_copy) - 从已有的变换中复制出一个变换。
* [d3.scale.category10](https://github.com/mbostock/d3/wiki/Ordinal-Scales#wiki-category10) - 用10种颜色构建一个ordinal变换。
* [d3.scale.category20](https://github.com/mbostock/d3/wiki/Ordinal-Scales#wiki-category20) - 用20种颜色构建一个ordinal变换。
* [d3.scale.category20b](https://github.com/mbostock/d3/wiki/Ordinal-Scales#wiki-category20b) - 用另外20种颜色构建一个ordinal变换。
* [d3.scale.category20c](https://github.com/mbostock/d3/wiki/Ordinal-Scales#wiki-category20c) - 用另外20种颜色构建一个ordinal变换。

**[d3.svg (SVG)](https://github.com/mbostock/d3/wiki/SVG)**

**[Shapes](https://github.com/mbostock/d3/wiki/SVG-Shapes)**

* [d3.svg.line](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-line) - 创建一个线段生成器.
* [line](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-_line) - 在折线图里生成一段折线.
* [line.x](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-line_x) - 设置或获取*x*轴访问器.
* [line.y](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-line_y) - 设置或获取*y*轴访问器
* [line.interpolate](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-line_interpolate) - 设置或获取插值模式.
* [line.tension](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-line_tension) - 获取或设置曲线张力访问器(cardinal spline tension).
* [line.defined](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-line_defined) - 定义线条在某一点是否存在.
* [d3.svg.line.radial](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-line_radial) - 创建辐射线生成器.
* [line](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-_line_radial) - 生成分段的线性曲线，用于纬度线／雷达线图表.
* [line.radius](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-line_radial_radius) - 获取或设置*radius*访问器.
* [line.angle](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-line_radial_angle) - 获取或设置*angle*访问器.
* [line.defined](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-line_radial_defined) - 设置或获取线条定义存取器.
* [d3.svg.area](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-area) - 创建一个新的区域生成器.
* [area](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-_area) - 生成一个线性的区域,用于区域图表.
* [area.x](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-area_x) - 获取或设置*x*坐标的访问器.
* [area.x0](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-area_x0) - 获取或设置*x0*坐标(基线)的访问器.
* [area.x1](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-area_x1) - 获取或设置*x1*坐标(背线)的访问器.
* [area.y](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-area_y) - 获取或设置*y*坐标的访问器.
* [area.y0](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-area_y0) - 获取或设置*y0*坐标(基线)的访问器.
* [area.y1](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-area_y1) - 获取或设置*y1*坐标(背线)的访问器.
* [area.interpolate](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-area_interpolate) - 获取或设置插值模式.
* [area.tension](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-area_tension) - 获取或设置张力访问器(the cardinal spline tension).
* [area.defined](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-area_defined) - 判断获取或定义区域定义存取器.
* [d3.svg.area.radial](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-area_radial) - 创建新的区域生成器.
* [area](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-_area_radial) - 生成分段的线性区域,用于纬度/雷达图表.
* [area.radius](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-area_radial_radius) - 获取或设置*radius*访问器.
* [area.innerRadius](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-area_radial_innerRadius) - 获取或设置内部的*radius*(基线)访问器.
* [area.outerRadius](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-area_radial_outerRadius) - 获取或设置外部的*radius*(背线)访问器.
* [area.angle](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-area_radial_angle) - 获取或设置*angle*访问器.
* [area.startAngle](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-area_radial_startAngle) - 获取或设置内部的*angle*(基线)访问器.
* [area.endAngle](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-area_radial_endAngle) - 获取或设置外部的*angle*(背线)访问器.
* [area.defined](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-area_radial_defined) - 判断获取或定义区域定义存取器.
* [d3.svg.arc](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-arc) - 创建弧度生成器.
* [arc](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-_arc) - 生成一个线性弧度,用于饼图或甜甜圈图.
* [arc.innerRadius](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-arc_innerRadius) - 获取或设置内部的半径访问器.
* [arc.outerRadius](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-arc_outerRadius) - 获取或设置外部的半径访问器.
* [arc.startAngle](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-arc_startAngle) - 获取或设置起始角度访问器.
* [arc.endAngle](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-arc_endAngle) - 获取或设置结束角度访问器.
* [arc.centroid](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-arc_centroid) - 计算弧的重心点.
* [d3.svg.symbol](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-symbol) - 创建符号生成器.
* [symbol](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-_symbol) - 生成指定的符号,用于散列图.
* [symbol.type](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-symbol_type) - 获取或设置符号类型访问器.
* [symbol.size](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-symbol_size) - 获取或设置符号尺寸(in square pixels) 访问器.
* [d3.svg.symbolTypes](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-symbolTypes) - 被支持的符号类型数组.
* [d3.svg.chord](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-chord) - 创建新的弦生成器.
* [chord](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-_chord) - 生成一个二次贝塞尔曲线连接两个弧, 用于弦图.
* [chord.radius](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-chord_radius) - 获取或设置弧半径访问器.
* [chord.startAngle](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-chord_startAngle) - 获取或设置弧起始角度访问器.
* [chord.endAngle](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-chord_endAngle) - 获取或设置弧结束角度访问器.
* [chord.source](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-chord_source) - 获取或设置源弧度访问器.
* [chord.target](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-chord_target) - 获取或设置目标弧度访问器.
* [d3.svg.diagonal](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-diagonal) - 创建新的斜线生成器.
* [diagonal](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-_diagonal) - 生成一个二维贝塞尔连接器, 用于节点连接图.
* [diagonal.source](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-diagonal_source) - 获取或设置源点访问器.
* [diagonal.target](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-diagonal_target) - 获取或设置目标点访问器.
* [diagonal.projection](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-diagonal_projection) - 获取或设置一个可选的点变换器.
* [d3.svg.diagonal.radial](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-diagonal_radial) - 创建一个新的斜线生成器.
* [diagonal](https://github.com/mbostock/d3/wiki/SVG-Shapes#wiki-_diagonal_radial) - 创建一个二维贝塞尔连接器,用于节点连接图.

**[坐标轴(Axes)](https://github.com/mbostock/d3/wiki/SVG-Axes)**

* [d3.svg.axis](https://github.com/mbostock/d3/wiki/SVG-Axes#wiki-axis) - 创建一个axis生成器。
* [axis](https://github.com/mbostock/d3/wiki/SVG-Axes#wiki-_axis) - 正式在页面中生成axis。
* [axis.scale](https://github.com/mbostock/d3/wiki/SVG-Axes#wiki-scale) - get或set坐标轴的scale尺度变换，该尺度变换设定了数值和像素位置的转换规则。
* [axis.orient](https://github.com/mbostock/d3/wiki/SVG-Axes#wiki-orient) - get或set坐标轴刻度方向。
* [axis.ticks](https://github.com/mbostock/d3/wiki/SVG-Axes#wiki-ticks) - 控制坐标轴刻度的产生方式。
* [axis.tickValues](https://github.com/mbostock/d3/wiki/SVG-Axes#wiki-tickValues) - 设置特定的坐标轴的值。
* [axis.tickSize](https://github.com/mbostock/d3/wiki/SVG-Axes#wiki-tickSize) - 指定坐标轴上刻度线的像素长度。
* [axis.innerTickSize](https://github.com/mbostock/d3/wiki/SVG-Axes#wiki-innerTickSize) - get或set坐标轴小刻度线的像素长度。
* [axis.outerTickSize](https://github.com/mbostock/d3/wiki/SVG-Axes#wiki-outerTickSize) - get或set坐标轴大刻度线的像素长度。
* [axis.tickPadding](https://github.com/mbostock/d3/wiki/SVG-Axes#wiki-tickPadding) - 指定坐标轴刻度和刻度文字之间的像素距离。
* [axis.tickFormat](https://github.com/mbostock/d3/wiki/SVG-Axes#wiki-tickFormat) - 设置刻度文字的格式。

**[Controls](https://github.com/mbostock/d3/wiki/SVG-Controls)**

* [d3.svg.brush](https://github.com/mbostock/d3/wiki/SVG-Controls#wiki-brush) - 点击拖拽选择一个二维区域。
* [brush](https://github.com/mbostock/d3/wiki/SVG-Controls#wiki-_brush) - 在页面中某个区域中正式绑定一个brush。
* [brush.x](https://github.com/mbostock/d3/wiki/SVG-Controls#wiki-brush_x) - get或set brush的x变换,用于水平方向的拖拽。
* [brush.y](https://github.com/mbostock/d3/wiki/SVG-Controls#wiki-brush_y) - get或set brush的y变换,用于垂直方向的拖拽。
* [brush.extent](https://github.com/mbostock/d3/wiki/SVG-Controls#wiki-brush_extent) - get或set brush的选取范围（extent）。
* [brush.clear](https://github.com/mbostock/d3/wiki/SVG-Controls#wiki-brush_clear) - 设置brush的选取范围（extent）为空。
* [brush.empty](https://github.com/mbostock/d3/wiki/SVG-Controls#wiki-brush_empty) - 判断brush的选取范围（extent）是否为空。
* [brush.on](https://github.com/mbostock/d3/wiki/SVG-Controls#wiki-brush_on) - get或set brush的事件监听器。可监听3种事件：brushstart, brush, brushend。
* [brush.event](https://github.com/mbostock/d3/wiki/SVG-Controls#wiki-brush_event) - 通过程序触发监听事件，在通过程序设置extent后使用。

**[d3.time (Time)](https://github.com/mbostock/d3/wiki/Time)**

**[时间格式转换(Time Formatting)](https://github.com/mbostock/d3/wiki/Time-Formatting)**

* [d3.time.format](https://github.com/mbostock/d3/wiki/Time-Formatting#wiki-format) - 创建基于某种时间格式的本地时间格式转换器。
* [format](https://github.com/mbostock/d3/wiki/Time-Formatting#wiki-_format) - 将一个date对象转换成特定时间格式的字符串。
* [format.parse](https://github.com/mbostock/d3/wiki/Time-Formatting#wiki-parse) - 将特定时间格式的字符串转换成date对象。
* [d3.time.format.utc](https://github.com/mbostock/d3/wiki/Time-Formatting#wiki-format_utc) - 创建基于某种时间格式的世界标准时间（UTC）格式转换器。
* [d3.time.format.iso](https://github.com/mbostock/d3/wiki/Time-Formatting#wiki-format_iso) - 创建基于某种时间格式的ISO世界标准时间（ISO 8601 UTC）格式转换器。

**[时间变换(Time Scales)](https://github.com/mbostock/d3/wiki/Time%20Scales)**

* [d3.time.scale](https://github.com/mbostock/d3/wiki/Time-Scales#wiki-scale) - 创建一个线性时间变换，定义域为数值区间，值域为时间区间。常用于时间坐标轴的创建。详情可参考d3.scale.linear。
* [scale](https://github.com/mbostock/d3/wiki/Time-Scales#wiki-_scale) - 输入为一个数值，返回为一个时间。
* [scale.invert](https://github.com/mbostock/d3/wiki/Time-Scales#wiki-invert) - 反变换，输入时间返回数值。
* [scale.domain](https://github.com/mbostock/d3/wiki/Time-Scales#wiki-domain) - get或set变换的定义域。
* [scale.nice](https://github.com/mbostock/d3/wiki/Time-Scales#wiki-nice) - 扩展定义域范围使定义域更规整。
* [scale.range](https://github.com/mbostock/d3/wiki/Time-Scales#wiki-range) - get或set变换的值域。
* [scale.rangeRound](https://github.com/mbostock/d3/wiki/Time-Scales#wiki-rangeRound) - 设置值域，并对结果取整。
* [scale.interpolate](https://github.com/mbostock/d3/wiki/Time-Scales#wiki-interpolate) - get或set变换的插值函数，如将默认的线性插值函数替换成指数插值函数。
* [scale.clamp](https://github.com/mbostock/d3/wiki/Time-Scales#wiki-clamp) - 设置值域是否闭合，默认不闭合。当值域闭合时，如果插值结果在值域之外，会取值域的边界值。详情参考linear.clamp。
* [scale.ticks](https://github.com/mbostock/d3/wiki/Time-Scales#wiki-ticks) - 从定义域中取出有代表性的值。通常用于坐标轴刻度的选取。
* [scale.tickFormat](https://github.com/mbostock/d3/wiki/Time-Scales#wiki-tickFormat) - 获取格式转化函数，通常用于坐标轴刻度的格式转化。
* [scale.copy](https://github.com/mbostock/d3/wiki/Time-Scales#wiki-copy) - 从已有的时间变换中复制出一个变换。

**[Time Intervals](https://github.com/mbostock/d3/wiki/Time-Intervals)**

* [d3.time.interval](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-interval) - 返回一个对于本地时间时间间隔器.
* [interval](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-_interval) - 效果同interval.floor方法.
* [interval.range](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-interval_range) - 返回指定区间内日期.
* [interval.floor](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-interval_floor) - 下舍入到最近的间隔值.
* [interval.round](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-interval_round) - 上舍入或下舍入到最近的间隔值.
* [interval.ceil](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-interval_ceil) - 上舍入到最近的间隔值.
* [interval.offset](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-interval_offset) - 返回指定时间间隔的日期偏移量.
* [interval.utc](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-interval_utc) - 返回对应的UTC时间间隔.
* [d3.time.day](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-day) - 返回指定时间基于天起始的时间(默认起始是12:00am).
* [d3.time.days](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-day) - 返回指定时间区间和间隔条件的基于天的所有时间,效果同day.range.
* [d3.time.dayOfYear](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-dayOfYear) - 计算指定时间在年中的天数.
* [d3.time.hour](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-hour) - 返回指定时间基于小时起始的时间(e.g., 1:00 AM).
* [d3.time.hours](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-hours) - 返回指定时间区间和间隔条件的基于小时的所有时间, 效果同hour.range.
* [d3.time.minute](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-minute) - 返回指定时间基于分钟起始的时间 (e.g., 1:02 AM).
* [d3.time.minutes](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-minutes) - 返回指定时间区间和间隔条件的基于分钟的所有时间,效果同minute.range.
* [d3.time.month](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-month) - 返回指定时间基于月起始的时间(e.g., February 1, 12:00 AM).
* [d3.time.months](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-months) - 返回指定时间区间和间隔条件的基于月的所有时间,效果同month.range.
* [d3.time.second](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-second) - 返回指定时间基于秒起始的时间(e.g., 1:02:03 AM).
* [d3.time.seconds](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-seconds) - 返回指定时间区间和间隔条件的基于秒的所有时间,效果同second.range.
* [d3.time.sunday](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-sunday) - 返回指定时间基于Sunday起始的时间(e.g., February 5, 12:00 AM).
* [d3.time.sundays](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-sundays) - 返回指定时间区间和间隔条件的基于sunday的所有时间, 效果同sunday.range.
* [d3.time.sundayOfYear](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-sundayOfYear) - 计算以sunday为基点的指定时间在一年中的周数.
* [d3.time.monday](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-monday) - every Monday (e.g., February 5, 12:00 AM).
* [d3.time.mondays](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-mondays) - alias for monday.range.
* [d3.time.mondayOfYear](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-mondayOfYear) - computes the monday-based week number.
* [d3.time.tuesday](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-tuesday) - every Tuesday (e.g., February 5, 12:00 AM).
* [d3.time.tuesdays](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-tuesdays) - alias for tuesday.range.
* [d3.time.tuesdayOfYear](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-tuesdayOfYear) - computes the tuesday-based week number.
* [d3.time.wednesday](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-wednesday) - every Wednesday (e.g., February 5, 12:00 AM).
* [d3.time.wednesdays](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-wednesdays) - alias for wednesday.range.
* [d3.time.wednesdayOfYear](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-tuesdayOfYear) - computes the wednesday-based week number.
* [d3.time.thursday](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-thursday) - every Thursday (e.g., February 5, 12:00 AM).
* [d3.time.thursdays](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-thursdays) - alias for thursday.range.
* [d3.time.thursdayOfYear](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-thursdayOfYear) - computes the thursday-based week number.
* [d3.time.friday](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-friday) - every Friday (e.g., February 5, 12:00 AM).
* [d3.time.fridays](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-fridays) - alias for friday.range.
* [d3.time.fridayOfYear](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-fridayOfYear) - computes the friday-based week number.
* [d3.time.saturday](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-saturday) - every Saturday (e.g., February 5, 12:00 AM).
* [d3.time.saturdays](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-saturdays) - alias for saturday.range.
* [d3.time.saturdayOfYear](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-saturdayOfYear) - computes the saturday-based week number.
* [d3.time.week](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-week) - alias for sunday.
* [d3.time.weeks](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-weeks) - alias for sunday.range.
* [d3.time.weekOfYear](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-weekOfYear) - alias for sundayOfYear.
* [d3.time.year](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-year) - 返回指定时间基于年起始的时间(e.g., January 1, 12:00 AM).
* [d3.time.years](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-years) - 返回指定时间区间和间隔条件的所有时间,效果同year.range.

**[构图(d3.layout)](https://github.com/mbostock/d3/wiki/Layouts)**

**[Bundle](https://github.com/mbostock/d3/wiki/Bundle-Layout)**

* [d3.layout.bundle](https://github.com/mbostock/d3/wiki/Bundle-Layout#wiki-bundle) - construct a new default bundle layout.
* [bundle](https://github.com/mbostock/d3/wiki/Bundle-Layout#wiki-_bundle) - apply Holten's *hierarchical bundling* algorithm to edges.

**[弦图(Chord)](https://github.com/mbostock/d3/wiki/Chord-Layout)**

* [d3.layout.chord](https://github.com/mbostock/d3/wiki/Chord-Layout#wiki-chord) - 初始化一个弦图对象, 返回一个 Chord 实例
* [chord.matrix](https://github.com/mbostock/d3/wiki/Chord-Layout#wiki-matrix) - 设置或者获取弦图实例对应的矩阵数据
* [chord.padding](https://github.com/mbostock/d3/wiki/Chord-Layout#wiki-padding) - 设置或获取弦图各段圆弧之间的间隔角度
* [chord.sortGroups](https://github.com/mbostock/d3/wiki/Chord-Layout#wiki-sortGroups) - 设置或获取矩阵分组的排序函数
* [chord.sortSubgroups](https://github.com/mbostock/d3/wiki/Chord-Layout#wiki-sortSubgroups) - 设置或获取矩阵二级分组的排序函数
* [chord.sortChords](https://github.com/mbostock/d3/wiki/Chord-Layout#wiki-sortChords) - 设置或获取弦图在z序上的排序函数(决定哪一组显示在最上层)
* [chord.chords](https://github.com/mbostock/d3/wiki/Chord-Layout#wiki-chords) - 该函数会将参数处理成对 chord 更友好的格式并返回, 若没有提供参数, 会调用matrix()来获取数据
* [chord.groups](https://github.com/mbostock/d3/wiki/Chord-Layout#wiki-groups) - 该函数参数处理成更易于理解的分组信息, 若没有提供参数, 会调用matrix()来获取数据

**[集群(Cluster)](https://github.com/mbostock/d3/wiki/Cluster-Layout)**

* [d3.layout.cluster](https://github.com/mbostock/d3/wiki/Cluster-Layout#wiki-cluster) - 用默认设置生成一个集群布局对象.
* [cluster.sort](https://github.com/mbostock/d3/wiki/Cluster-Layout#wiki-sort) - 获取或设置一个函数, 用来给兄弟节点(同一父结点的子结点)的排序.
* [cluster.children](https://github.com/mbostock/d3/wiki/Cluster-Layout#wiki-children) - 获取或设置子结点的访问器.
* [cluster.nodes](https://github.com/mbostock/d3/wiki/Cluster-Layout#wiki-nodes) - 计算并返回指定结点的子结点在集群中的信息(坐标,深度等).
* [cluster.links](https://github.com/mbostock/d3/wiki/Cluster-Layout#wiki-links) - 指定一个子结点数组(通常是**nodes**函数返回值), 计算它们与父结点的连接信息.
* [cluster.separation](https://github.com/mbostock/d3/wiki/Cluster-Layout#wiki-separation) - 获取或设置相邻结点间的间隔(不仅限于兄弟结点).
* [cluster.size](https://github.com/mbostock/d3/wiki/Cluster-Layout#wiki-size) - 获取或设置布局的 *宽* 和 *高* 的大小.
* [cluster.nodeSize](https://github.com/mbostock/d3/wiki/Cluster-Layout#wiki-nodeSize) - 为结点指定大小.

**[力学(Force)](https://github.com/mbostock/d3/wiki/Force-Layout)**

* [d3.layout.force](https://github.com/mbostock/d3/wiki/Force-Layout#wiki-force) -节点（node）基于物理模拟的位置连接。
* [force.on](https://github.com/mbostock/d3/wiki/Force-Layout#wiki-on) - 监听布局位置的变化。(仅支持"start","step","end"三种事件)
* [force.nodes](https://github.com/mbostock/d3/wiki/Force-Layout#wiki-nodes) - 获得或设置布局中的节点（node）阵列组。
* [force.links](https://github.com/mbostock/d3/wiki/Force-Layout#wiki-links) - 获得或设置布局中节点间的连接（Link）阵列组。.
* [force.size](https://github.com/mbostock/d3/wiki/Force-Layout#wiki-size) - 获取或设置布局的 *宽* 和 *高* 的大小.
* [force.linkDistance](https://github.com/mbostock/d3/wiki/Force-Layout#wiki-linkDistance) - 获取或设置节点间的连接线距离.
* [force.linkStrength](https://github.com/mbostock/d3/wiki/Force-Layout#wiki-linkStrength) - 获取或设置节点间的连接强度.
* [force.friction](https://github.com/mbostock/d3/wiki/Force-Layout#wiki-friction) - 获取或设置摩擦系数.
* [force.charge](https://github.com/mbostock/d3/wiki/Force-Layout#wiki-charge) - 获取或设置节点的电荷数.(电荷数决定结点是互相排斥还是吸引)
* [force.gravity](https://github.com/mbostock/d3/wiki/Force-Layout#wiki-gravity) - 获取或设置节点的引力强度.
* [force.theta](https://github.com/mbostock/d3/wiki/Force-Layout#wiki-theta) - 获取或设置电荷间互相作用的强度.
* [force.start](https://github.com/mbostock/d3/wiki/Force-Layout#wiki-start) - 开启或恢复结点间的位置影响.
* [force.resume](https://github.com/mbostock/d3/wiki/Force-Layout#wiki-resume) - 设置冷却系数为0.1,并重新调用start()函数.
* [force.stop](https://github.com/mbostock/d3/wiki/Force-Layout#wiki-stop) - 立刻终止结点间的位置影响.(等同于将*冷却系数*设置为0)
* [force.alpha](https://github.com/mbostock/d3/wiki/Force-Layout#wiki-alpha) - 获取或设置布局的冷却系数.(冷却系数为0时,节点间不再互相影响)
* [force.tick](https://github.com/mbostock/d3/wiki/Force-Layout#wiki-tick) - 让布局运行到下一步.
* [force.drag](https://github.com/mbostock/d3/wiki/Force-Layout#wiki-drag) - 获取当前布局的拖拽对象实例以便进一步绑定处理函数.

**[层级布局(Hierarchy)](https://github.com/mbostock/d3/wiki/Hierarchy-Layout)**

* [d3.layout.hierarchy](https://github.com/mbostock/d3/wiki/Hierarchy-Layout#wiki-hierarchy) - 获得一个自定义的层级布局的实现.
* [hierarchy.sort](https://github.com/mbostock/d3/wiki/Hierarchy-Layout#wiki-sort) - 获取或设置一个函数, 用来给兄弟节点(同一父结点的子结点)的排序.
* [hierarchy.children](https://github.com/mbostock/d3/wiki/Hierarchy-Layout#wiki-children) - 获取或设置子结点的访问器.
* [hierarchy.nodes](https://github.com/mbostock/d3/wiki/Hierarchy-Layout#wiki-nodes) - 计算并返回指定结点的子结点信息.
* [hierarchy.links](https://github.com/mbostock/d3/wiki/Hierarchy-Layout#wiki-links) - 指定一个子结点数组(通常是**nodes**函数返回值), 计算它们与父结点的连接信息.
* [hierarchy.value](https://github.com/mbostock/d3/wiki/Hierarchy-Layout#wiki-value) - 获取或设置结点的**值**访问器.
* [hierarchy.revalue](https://github.com/mbostock/d3/wiki/Hierarchy-Layout#wiki-revalue) - 重新计算层级布局.

**[直方图(Histogram)](https://github.com/mbostock/d3/wiki/Histogram-Layout)**

* [d3.layout.histogram](https://github.com/mbostock/d3/wiki/Histogram-Layout#wiki-histogram) - 构建一个默认直方图(用来表示一组离散数字的分布,横轴表示区间,纵轴表示区间内样本数量或样本百分比).
* [histogram.value](https://github.com/mbostock/d3/wiki/Histogram-Layout#wiki-value) - 获取或设置值访问器.
* [histogram.range](https://github.com/mbostock/d3/wiki/Histogram-Layout#wiki-range) - 获取或设置合法值范围.
* [histogram.bins](https://github.com/mbostock/d3/wiki/Histogram-Layout#wiki-bins) - 指定如何将数据分组到不同的区间(bin)里, 返回一个[构造函数](https://github.com/mbostock/d3/wiki/Histogram-Layout#wiki-_histogram).
* [histogram](https://github.com/mbostock/d3/wiki/Histogram-Layout#wiki-_histogram) - 根据已设置的区间将数据分组,返回已分组的二维数组(compute the distribution of data using quantized bins).
* [histogram.frequency](https://github.com/mbostock/d3/wiki/Histogram-Layout#wiki-frequency) - 设置直方图Y轴值是区间内数据的总量还是百分比(compute the distribution as counts or probabilities).

**[层包(Pack)](https://github.com/mbostock/d3/wiki/Pack-Layout)**

* [d3.layout.pack](https://github.com/mbostock/d3/wiki/Pack-Layout#wiki-pack) - 用递归的圆环表现一个多层级布局.
* [pack.sort](https://github.com/mbostock/d3/wiki/Pack-Layout#wiki-sort) - 获取或设置一个函数, 用来给兄弟节点(同一父结点的子结点)排序.
* [pack.children](https://github.com/mbostock/d3/wiki/Pack-Layout#wiki-children) - 获取或设置子结点的访问器.
* [pack.nodes](https://github.com/mbostock/d3/wiki/Pack-Layout#wiki-nodes) - 计算并返回指定结点的子结点信息.
* [pack.links](https://github.com/mbostock/d3/wiki/Pack-Layout#wiki-links) - 指定一个子结点数组(通常是**nodes**函数返回值), 计算它们与父结点的连接信息.
* [pack.value](https://github.com/mbostock/d3/wiki/Pack-Layout#wiki-value) - 获取或设置一个函数, 用来计算圆环的大小(近似值).
* [pack.size](https://github.com/mbostock/d3/wiki/Pack-Layout#wiki-size) - 设置整个布局画布的 *宽* and *高*.
* [pack.radius](https://github.com/mbostock/d3/wiki/Pack-Layout#wiki-radius) - 如果不想结点半径与结点的值相同, 可以传入一个函数用来计算结点半径.
* [pack.padding](https://github.com/mbostock/d3/wiki/Pack-Layout#wiki-padding) - 指定相邻结点之点的间距(近似值).

**[分区(Partition)](https://github.com/mbostock/d3/wiki/Partition-Layout)**

* [d3.layout.partition](https://github.com/mbostock/d3/wiki/Partition-Layout#wiki-partition) - 将一棵树递归的分区.
* [partition.sort](https://github.com/mbostock/d3/wiki/Partition-Layout#wiki-sort) - 获取或设置一个函数, 用来给兄弟节点(同一父结点的子结点)排序.
* [partition.children](https://github.com/mbostock/d3/wiki/Partition-Layout#wiki-children) - 获取或设置子结点的访问器.
* [partition.nodes](https://github.com/mbostock/d3/wiki/Partition-Layout#wiki-nodes) - 计算并返回指定结点的子结点信息.
* [partition.links](https://github.com/mbostock/d3/wiki/Partition-Layout#wiki-links) - 指定一个子结点数组(通常是**nodes**函数返回值), 计算它们与父结点的连接信息.
* [partition.value](https://github.com/mbostock/d3/wiki/Partition-Layout#wiki-value) - 设置一个函数来来计算分区的值.
* [partition.size](https://github.com/mbostock/d3/wiki/Partition-Layout#wiki-size) - 设置整个布局画布的 *宽* and *高*.

**[饼图(Pie)](https://github.com/mbostock/d3/wiki/Pie-Layout)**

* [d3.layout.pie](https://github.com/mbostock/d3/wiki/Pie-Layout#wiki-pie) - 构建一个默认的饼图.
* [pie](https://github.com/mbostock/d3/wiki/Pie-Layout#wiki-_pie) - 该函数将传入的原始参数转换成可用于饼图或者环形图的数据结构.
* [pie.value](https://github.com/mbostock/d3/wiki/Pie-Layout#wiki-value) - 获取或设置值访问器.
* [pie.sort](https://github.com/mbostock/d3/wiki/Pie-Layout#wiki-sort) - 设置饼图顺时针方向的排序方法.
* [pie.startAngle](https://github.com/mbostock/d3/wiki/Pie-Layout#wiki-startAngle) - 设置或获取整个饼图的起始角度.
* [pie.endAngle](https://github.com/mbostock/d3/wiki/Pie-Layout#wiki-endAngle) - 设置或获取整个饼图的终止角度.

**[堆叠图(Stack)](https://github.com/mbostock/d3/wiki/Stack-Layout)**

* [d3.layout.stack](https://github.com/mbostock/d3/wiki/Stack-Layout#wiki-stack) - 构建一个默认的堆叠图(用来展示一系列x轴相同的面积图或者立方图).
* [stack](https://github.com/mbostock/d3/wiki/Stack-Layout#wiki-_stack) - 计算每一层的基线.
* [stack.values](https://github.com/mbostock/d3/wiki/Stack-Layout#wiki-values) - 设置或者获取每层的值访问器.
* [stack.order](https://github.com/mbostock/d3/wiki/Stack-Layout#wiki-order) - 设置每层的排序.
* [stack.offset](https://github.com/mbostock/d3/wiki/Stack-Layout#wiki-offset) - 指定总的基线算法.
* [stack.x](https://github.com/mbostock/d3/wiki/Stack-Layout#wiki-x) - 设置或获取每层的x轴访问器.
* [stack.y](https://github.com/mbostock/d3/wiki/Stack-Layout#wiki-y) - 设置或获取每层的y轴访问器.
* [stack.out](https://github.com/mbostock/d3/wiki/Stack-Layout#wiki-out) - 设置或获取用来储存基线的输出函数.

**[树(Tree)](https://github.com/mbostock/d3/wiki/Tree-Layout)**

* [d3.layout.tree](https://github.com/mbostock/d3/wiki/Tree-Layout#wiki-tree) - position a tree of nodes tidily.
* [tree.sort](https://github.com/mbostock/d3/wiki/Tree-Layout#wiki-sort) - 设置或获取一个函数, 用来给兄弟节点(同一父结点的子结点)排序.
* [tree.children](https://github.com/mbostock/d3/wiki/Tree-Layout#wiki-children) - 设置或获取子结点的访问器.
* [tree.nodes](https://github.com/mbostock/d3/wiki/Tree-Layout#wiki-nodes) - 计算并返回指定结点的子结点信息.
* [tree.links](https://github.com/mbostock/d3/wiki/Tree-Layout#wiki-links) - 指定一个子结点数组(通常是**nodes**函数返回值), 计算它们与父结点的连接信息.
* [tree.separation](https://github.com/mbostock/d3/wiki/Tree-Layout#wiki-separation) - 设置或获取相隔结点之间的间隔计算函数.
* [tree.size](https://github.com/mbostock/d3/wiki/Tree-Layout#wiki-size) - 指定整个布局的宽和高.
* [tree.nodeSize](https://github.com/mbostock/d3/wiki/Tree-Layout#wiki-nodeSize) - 给全部结点指定一个固定的大小(会导致[tree.size](https://github.com/mbostock/d3/wiki/Tree-Layout#wiki-size)失效).

**[矩阵树(Treemap)](https://github.com/mbostock/d3/wiki/Treemap-Layout)**

* [d3.layout.treemap](https://github.com/mbostock/d3/wiki/Treemap-Layout#wiki-treemap) - 返回一个矩阵树对象(用矩阵来展示一颗树).
* [treemap.sort](https://github.com/mbostock/d3/wiki/Treemap-Layout#wiki-sort) - 设置或获取一个函数, 用来给兄弟节点(同一父结点的子结点)排序.
* [treemap.children](https://github.com/mbostock/d3/wiki/Treemap-Layout#wiki-children) - 设置或获取子结点的访问器.
* [treemap.nodes](https://github.com/mbostock/d3/wiki/Treemap-Layout#wiki-nodes) - 计算并返回指定结点的子结点信息.
* [treemap.links](https://github.com/mbostock/d3/wiki/Treemap-Layout#wiki-links) - 指定一个子结点数组(通常是**nodes**函数返回值), 计算它们与父结点的连接信息.
* [treemap.value](https://github.com/mbostock/d3/wiki/Treemap-Layout#wiki-value) - 设置或获取一个用来计算单元格大小的值访问器.
* [treemap.size](https://github.com/mbostock/d3/wiki/Treemap-Layout#wiki-size) - 指定整个布局的宽和高.
* [treemap.padding](https://github.com/mbostock/d3/wiki/Treemap-Layout#wiki-padding) - 指定父结点和子结点的间距.
* [treemap.round](https://github.com/mbostock/d3/wiki/Treemap-Layout#wiki-round) - 禁用或启用边界补偿.
* [treemap.sticky](https://github.com/mbostock/d3/wiki/Treemap-Layout#wiki-sticky) - 让布局更"粘"以保证在更新数据时有平滑的动画效果.
* [treemap.mode](https://github.com/mbostock/d3/wiki/Treemap-Layout#wiki-mode) - 更改矩阵树的布局算法.

**[d3.geo (Geography)](https://github.com/mbostock/d3/wiki/Geo)**

**[Paths](https://github.com/mbostock/d3/wiki/Geo-Paths)**

* [d3.geo.path](https://github.com/mbostock/d3/wiki/Geo-Paths#wiki-path) - 创建一个新的地理路径生成器.
* [path](https://github.com/mbostock/d3/wiki/Geo-Paths#wiki-_path) - 投射指定的特性并且渲染到上下文.
* [path.projection](https://github.com/mbostock/d3/wiki/Geo-Paths#wiki-path_projection) - 获取活设置地理投影.
* [path.context](https://github.com/mbostock/d3/wiki/Geo-Paths#wiki-path_context) - 获取活设置渲染上下文.
* [path.pointRadius](https://github.com/mbostock/d3/wiki/Geo-Paths#wiki-path_pointRadius) -获取或设置半径去现实点的特性.
* [path.area](https://github.com/mbostock/d3/wiki/Geo-Paths#wiki-path_area) - 计算指定特性的投射区域.
* [path.centroid](https://github.com/mbostock/d3/wiki/Geo-Paths#wiki-path_centroid) - 计算指定特性的投射重心点.
* [path.bounds](https://github.com/mbostock/d3/wiki/Geo-Paths#wiki-path_bounds) - 计算指定特性的投射边界.
* [d3.geo.graticule](https://github.com/mbostock/d3/wiki/Geo-Paths#wiki-graticule) - 创建地理坐标网生成器.
* [graticule](https://github.com/mbostock/d3/wiki/Geo-Paths#wiki-_graticule) - 生产一个子午线或平行线的MultiLineStrings.
* [graticule.lines](https://github.com/mbostock/d3/wiki/Geo-Paths#wiki-graticule_lines) - 生成一个子午线和平行线的LineString的数组.
* [graticule.outline](https://github.com/mbostock/d3/wiki/Geo-Paths#wiki-graticule_outline) - 生成一个表示该坐标网的外框多边形.
* [graticule.extent](https://github.com/mbostock/d3/wiki/Geo-Paths#wiki-graticule_extent) - 获取或设置主要的和次要的范围.
* [graticule.majorExtent](https://github.com/mbostock/d3/wiki/Geo-Paths#wiki-graticule_majorExtent) - get or set the major extent.
* [graticule.minorExtent](https://github.com/mbostock/d3/wiki/Geo-Paths#wiki-graticule_minorExtent) - get or set the minor extent.
* [graticule.step](https://github.com/mbostock/d3/wiki/Geo-Paths#wiki-graticule_step) - get or set the major & minor step intervals.
* [graticule.majorStep](https://github.com/mbostock/d3/wiki/Geo-Paths#wiki-graticule_majorStep) - get or set the major step intervals.
* [graticule.minorStep](https://github.com/mbostock/d3/wiki/Geo-Paths#wiki-graticule_minorStep) - get or set the minor step intervals.
* [graticule.precision](https://github.com/mbostock/d3/wiki/Geo-Paths#wiki-graticule_precision) - get or set the latitudinal precision.
* [d3.geo.circle](https://github.com/mbostock/d3/wiki/Geo-Paths#wiki-circle) - create a circle generator.
* [circle](https://github.com/mbostock/d3/wiki/Geo-Paths#wiki-_circle) - generate a piecewise circle as a Polygon.
* [circle.origin](https://github.com/mbostock/d3/wiki/Geo-Paths#wiki-circle_origin) - specify the origin in latitude and longitude.
* [circle.angle](https://github.com/mbostock/d3/wiki/Geo-Paths#wiki-circle_angle) - specify the angular radius in degrees.
* [circle.precision](https://github.com/mbostock/d3/wiki/Geo-Paths#wiki-circle_precision) - specify the precision of the piecewise circle.
* [d3.geo.area](https://github.com/mbostock/d3/wiki/Geo-Paths#wiki-area) - compute the spherical area of a given feature.
* [d3.geo.bounds](https://github.com/mbostock/d3/wiki/Geo-Paths#wiki-bounds) - compute the latitude-longitude bounding box for a given feature.
* [d3.geo.centroid](https://github.com/mbostock/d3/wiki/Geo-Paths#wiki-centroid) - compute the spherical centroid of a given feature.
* [d3.geo.distance](https://github.com/mbostock/d3/wiki/Geo-Paths#wiki-distance) - compute the great-arc distance between two points.
* [d3.geo.interpolate](https://github.com/mbostock/d3/wiki/Geo-Paths#wiki-interpolate) - interpolate between two points along a great arc.
* [d3.geo.length](https://github.com/mbostock/d3/wiki/Geo-Paths#wiki-length) - compute the length of a line string or the circumference of a polygon.
* [d3.geo.rotation](https://github.com/mbostock/d3/wiki/Geo-Paths#wiki-rotation) - create a rotation function for the specified angles [λ, φ, γ].
* [rotation](https://github.com/mbostock/d3/wiki/Geo-Paths#wiki-_rotation) - rotate the given location around the sphere.
* [rotation.invert](https://github.com/mbostock/d3/wiki/Geo-Paths#wiki-rotation_invert) - inverse-rotate the given location around the sphere.

**[Projections](https://github.com/mbostock/d3/wiki/Geo-Projections)**

* [d3.geo.projection](https://github.com/mbostock/d3/wiki/Geo-Projections#wiki-projection) - create a standard projection from a raw projection.
* [projection](https://github.com/mbostock/d3/wiki/Geo-Projections#wiki-_projection) - project the specified location.
* [projection.invert](https://github.com/mbostock/d3/wiki/Geo-Projections#wiki-invert) - invert the projection for the specified point.
* [projection.rotate](https://github.com/mbostock/d3/wiki/Geo-Projections#wiki-rotate) - get or set the projection’s three-axis rotation.
* [projection.center](https://github.com/mbostock/d3/wiki/Geo-Projections#wiki-center) - get or set the projection’s center location.
* [projection.translate](https://github.com/mbostock/d3/wiki/Geo-Projections#wiki-translate) - get or set the projection’s translation position.
* [projection.scale](https://github.com/mbostock/d3/wiki/Geo-Projections#wiki-scale) - get or set the projection’s scale factor.
* [projection.clipAngle](https://github.com/mbostock/d3/wiki/Geo-Projections#wiki-clipAngle) - get or set the radius of the projection’s clip circle.
* [projection.clipExtent](https://github.com/mbostock/d3/wiki/Geo-Projections#wiki-clipExtent) - get or set the projection’s viewport clip extent, in pixels.
* [projection.precision](https://github.com/mbostock/d3/wiki/Geo-Projections#wiki-precision) - get or set the precision threshold for adaptive resampling.
* [projection.stream](https://github.com/mbostock/d3/wiki/Geo-Projections#wiki-stream) - wrap the specified stream listener, projecting input geometry.
* [d3.geo.projectionMutator](https://github.com/mbostock/d3/wiki/Geo-Projections#wiki-projectionMutator) - create a standard projection from a mutable raw projection.
* [d3.geo.albers](https://github.com/mbostock/d3/wiki/Geo-Projections#wiki-albers) - the Albers equal-area conic projection.
* [albers.parallels](https://github.com/mbostock/d3/wiki/Geo-Projections#wiki-albers_parallels) - get or set the projection's two standard parallels.
* [d3.geo.albersUsa](https://github.com/mbostock/d3/wiki/Geo-Projections#wiki-albersUsa) - a composite Albers projection for the United States.
* [d3.geo.azimuthalEqualArea](https://github.com/mbostock/d3/wiki/Geo-Projections#wiki-azimuthalEqualArea) - the azimuthal equal-area projection.
* [d3.geo.azimuthalEquidistant](https://github.com/mbostock/d3/wiki/Geo-Projections#wiki-azimuthalEquidistant) - the azimuthal equidistant projection.
* [d3.geo.conicConformal](https://github.com/mbostock/d3/wiki/Geo-Projections#wiki-conicConformal) - the conic conformal projection.
* [d3.geo.conicEquidistant](https://github.com/mbostock/d3/wiki/Geo-Projections#wiki-conicEquidistant) - the conic equidistant projection.
* [d3.geo.conicEqualArea](https://github.com/mbostock/d3/wiki/Geo-Projections#wiki-conicEqualArea) the conic equal-area (a.k.a. Albers) projection.
* [d3.geo.equirectangular](https://github.com/mbostock/d3/wiki/Geo-Projections#wiki-equirectangular) - the equirectangular (plate carreé) projection.
* [d3.geo.gnomonic](https://github.com/mbostock/d3/wiki/Geo-Projections#wiki-gnomonic) - the gnomonic projection.
* [d3.geo.mercator](https://github.com/mbostock/d3/wiki/Geo-Projections#wiki-mercator) - the spherical Mercator projection.
* [d3.geo.orthographic](https://github.com/mbostock/d3/wiki/Geo-Projections#wiki-orthographic) - the azimuthal orthographic projection.
* [d3.geo.stereographic](https://github.com/mbostock/d3/wiki/Geo-Projections#wiki-stereographic) - the azimuthal stereographic projection.
* [d3.geo.azimuthalEqualArea.raw](https://github.com/mbostock/d3/wiki/Geo-Projections#wiki-azimuthalEqualArea_raw) - the raw azimuthal equal-area projection.
* [d3.geo.azimuthalEquidistant.raw](https://github.com/mbostock/d3/wiki/Geo-Projections#wiki-azimuthalEquidistant_raw) - the azimuthal equidistant projection.
* [d3.geo.conicConformal.raw](https://github.com/mbostock/d3/wiki/Geo-Projections#wiki-conicConformal_raw) - the raw conic conformal projection.
* [d3.geo.conicEquidistant.raw](https://github.com/mbostock/d3/wiki/Geo-Projections#wiki-conicEquidistant_raw) - the raw conic equidistant projection.
* [d3.geo.conicEqualArea.raw](https://github.com/mbostock/d3/wiki/Geo-Projections#wiki-conicEqualArea_raw) the raw conic equal-area (a.k.a. Albers) projection.
* [d3.geo.equirectangular.raw](https://github.com/mbostock/d3/wiki/Geo-Projections#wiki-equirectangular_raw) - the raw equirectangular (plate carrée) projection.
* [d3.geo.gnomonic.raw](https://github.com/mbostock/d3/wiki/Geo-Projections#wiki-gnomonic_raw) - the raw gnomonic projection.
* [d3.geo.mercator.raw](https://github.com/mbostock/d3/wiki/Geo-Projections#wiki-mercator_raw) - the raw Mercator projection.
* [d3.geo.orthographic.raw](https://github.com/mbostock/d3/wiki/Geo-Projections#wiki-orthographic_raw) - the raw azimuthal orthographic projection.
* [d3.geo.stereographic.raw](https://github.com/mbostock/d3/wiki/Geo-Projections#wiki-stereographic_raw) - the raw azimuthal stereographic projection.
* [d3.geo.transverseMercator.raw](https://github.com/mbostock/d3/wiki/Geo-Projections#wiki-transverseMercator_raw) - the raw transverse Mercator projection.

**[Streams](https://github.com/mbostock/d3/wiki/Geo-Streams)**

* [d3.geo.stream](https://github.com/mbostock/d3/wiki/Geo-Streams#wiki-stream) - convert a GeoJSON object to a geometry stream.
* [stream.point](https://github.com/mbostock/d3/wiki/Geo-Streams#wiki-stream_point) - indicate an *x*, *y* (and optionally *z*) coordinate.
* [stream.lineStart](https://github.com/mbostock/d3/wiki/Geo-Streams#wiki-stream_lineStart) - indicate the start of a line or ring.
* [stream.lineEnd](https://github.com/mbostock/d3/wiki/Geo-Streams#wiki-stream_lineEnd) - indicate the end of a line or ring.
* [stream.polygonStart](https://github.com/mbostock/d3/wiki/Geo-Streams#wiki-stream_polygonStart) - indicate the start of a polygon.
* [stream.polygonEnd](https://github.com/mbostock/d3/wiki/Geo-Streams#wiki-stream_polygonEnd) - indicate the end of a polygon.
* [stream.sphere](https://github.com/mbostock/d3/wiki/Geo-Streams#wiki-stream_sphere) - indicate a sphere.
* [d3.geo.transform](https://github.com/mbostock/d3/wiki/Geo-Streams#wiki-transform) - transform streaming geometries.
* [transform.stream](https://github.com/mbostock/d3/wiki/Geo-Streams#wiki-transform_stream) - wraps a given stream.
* [d3.geo.clipExtent](https://github.com/mbostock/d3/wiki/Geo-Streams#wiki-clipExtent) - a stream transform that clips geometries to a given axis-aligned rectangle.
* [clipExtent.extent](https://github.com/mbostock/d3/wiki/Geo-Streams#wiki-clipExtent_extent) - sets the clip extent.

**[d3.geom (Geometry)](https://github.com/mbostock/d3/wiki/Geometry)**

**[Voronoi](https://github.com/mbostock/d3/wiki/Voronoi-Geom)**

* [d3.geom.voronoi](https://github.com/mbostock/d3/wiki/Voronoi-Geom#wiki-voronoi) - create a Voronoi layout with default accessors.
* [voronoi](https://github.com/mbostock/d3/wiki/Voronoi-Geom#wiki-_voronoi) - compute the Voronoi tessellation for the specified points.
* [voronoi.x](https://github.com/mbostock/d3/wiki/Voronoi-Geom#wiki-x) - get or set the x-coordinate accessor for each point.
* [voronoi.y](https://github.com/mbostock/d3/wiki/Voronoi-Geom#wiki-y) - get or set the y-coordinate accessor for each point.
* [voronoi.clipExent](https://github.com/mbostock/d3/wiki/Voronoi-Geom#wiki-clipExtent) - get or set the clip extent for the tesselation.
* [voronoi.links](https://github.com/mbostock/d3/wiki/Voronoi-Geom#wiki-links) - compute the Delaunay mesh as a network of links.
* [voronoi.triangles](https://github.com/mbostock/d3/wiki/Voronoi-Geom#wiki-triangles) - compute the Delaunay mesh as a triangular tessellation.

**[Quadtree](https://github.com/mbostock/d3/wiki/Quadtree-Geom)**

* [d3.geom.quadtree](https://github.com/mbostock/d3/wiki/Quadtree-Geom#wiki-quadtree) - constructs a quadtree for an array of points.
* [quadtree.add](https://github.com/mbostock/d3/wiki/Quadtree-Geom#wiki-add) - add a point to the quadtree.
* [quadtree.visit](https://github.com/mbostock/d3/wiki/Quadtree-Geom#wiki-visit) - recursively visit nodes in the quadtree.

**[Polygon](https://github.com/mbostock/d3/wiki/Polygon-Geom)**

* [d3.geom.polygon](https://github.com/mbostock/d3/wiki/Polygon-Geom#wiki-polygon) - create a polygon from the specified array of points.
* [polygon.area](https://github.com/mbostock/d3/wiki/Polygon-Geom#wiki-area) - compute the counterclockwise area of this polygon.
* [polygon.centroid](https://github.com/mbostock/d3/wiki/Polygon-Geom#wiki-centroid) - compute the area centroid of this polygon.
* [polygon.clip](https://github.com/mbostock/d3/wiki/Polygon-Geom#wiki-clip) - clip the specified polygon to this polygon.

**[Hull](https://github.com/mbostock/d3/wiki/Hull-Geom)**

* [d3.geom.hull](https://github.com/mbostock/d3/wiki/Hull-Geom#wiki-hull) - create a convex hull layout with default accessors.
* [hull](https://github.com/mbostock/d3/wiki/Hull-Geom#wiki-_hull) - compute the convex hull for the given array of points.
* [hull.x](https://github.com/mbostock/d3/wiki/Hull-Geom#wiki-x) - get or set the *x*-coordinate accessor.
* [hull.y](https://github.com/mbostock/d3/wiki/Hull-Geom#wiki-y) - get or set the *y*-coordinate accessor.

**[d3.behavior (Behaviors)](https://github.com/mbostock/d3/wiki/Behaviors)**

**[Drag](https://github.com/mbostock/d3/wiki/Drag-Behavior)**

* [d3.behavior.drag](https://github.com/mbostock/d3/wiki/Drag-Behavior#wiki-drag)
* [drag.origin](https://github.com/mbostock/d3/wiki/Drag-Behavior#wiki-origin)
* [drag.on](https://github.com/mbostock/d3/wiki/Drag-Behavior#wiki-on)

**[Zoom](https://github.com/mbostock/d3/wiki/Zoom-Behavior)**

* [d3.behavior.zoom](https://github.com/mbostock/d3/wiki/Zoom-Behavior#wiki-zoom) - create a zoom behavior.
* [zoom](https://github.com/mbostock/d3/wiki/Zoom-Behavior#wiki-_zoom) - apply the zoom behavior to the selected elements.
* [zoom.scale](https://github.com/mbostock/d3/wiki/Zoom-Behavior#wiki-scale) - the current scale factor.
* [zoom.translate](https://github.com/mbostock/d3/wiki/Zoom-Behavior#wiki-translate) - the current translate offset.
* [zoom.scaleExtent](https://github.com/mbostock/d3/wiki/Zoom-Behavior#wiki-scaleExtent) - optional limits on the scale factor.
* [zoom.center](https://github.com/mbostock/d3/wiki/Zoom-Behavior#wiki-center) - an optional focal point for mousewheel zooming.
* [zoom.size](https://github.com/mbostock/d3/wiki/Zoom-Behavior#wiki-size) - the dimensions of the viewport.
* [zoom.x](https://github.com/mbostock/d3/wiki/Zoom-Behavior#wiki-x) - an optional scale whose domain is bound to the *x* extent of the viewport.
* [zoom.y](https://github.com/mbostock/d3/wiki/Zoom-Behavior#wiki-y) - an optional scale whose domain is bound to the *y* extent of the viewport.
* [zoom.on](https://github.com/mbostock/d3/wiki/Zoom-Behavior#wiki-on) - listeners for when the scale or translate changes.
* [zoom.event](https://github.com/mbostock/d3/wiki/Zoom-Behavior#wiki-event) - dispatch zoom events after setting the scale or translate.