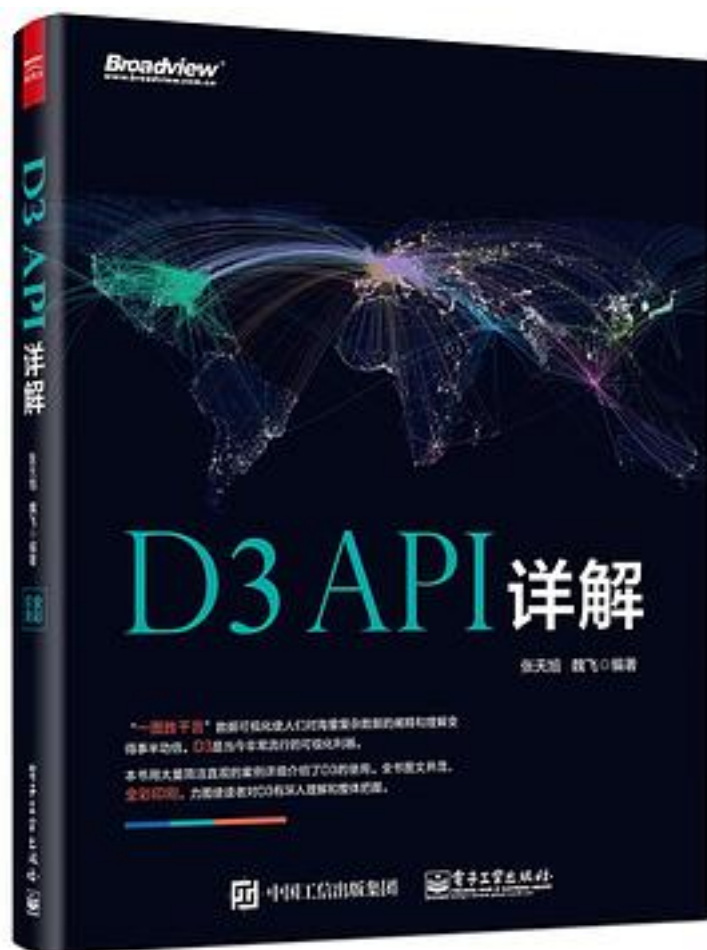


D3 API详解（全彩）



[D3 API详解（全彩）_下载链接1](#)

著者:张天旭 魏飞 编著

出版者:电子工业出版社

出版时间:2016-3

装帧:

isbn:9787121278990

D3 是目前流行的数据可视化工具。它是一个基于数据操作文档的JavaScript 函数库，因为其优雅灵活的语言风格，生动有趣的交互功能，花哨酷炫的表现力，受到越来越多的人关注和喜爱。

《D3 API详解（全彩）》基于官方API

文档，在尽量保留原文含义的基础上，对部分内容进行了删减和增补，几乎为每个函数都添加了浅显直观的案例。《D3 API详解（全彩）》涵盖了API大部分内容，包括核心函数中的选择器、过渡、数组、数学、数据请求、格式化、本地化；3类9种比例尺；D3对SVG绘图技术的封装，如SVG元素、多种路径生成器、数轴、刷子等；时间函数；12种布局，如捆、弦、树、簇、包、分区、矩形树、力、直方图、堆叠等；地理函数，如投影、流、地理路径、经纬网等；还有几何图形，像二叉树、凸包、多边形、泰森多边形等；以及交互行为中的拖动和缩放。

《D3 API详解（全彩）》适合所有想使用D3实现数据可视化方案的人使用。读者朋友可以将《D3 API详解（全彩）》作为入门参考，也可以作为速查手册。

作者介绍:

| | |
|--|----|
| 目录: 第1章 核心 (Core) | 1 |
| 1.1 选择 | 1 |
| 1.1.1 d3.select(selector) | 1 |
| 1.1.2 d3.select(node) | 2 |
| 1.1.3 d3.selectAll(selector) | 3 |
| 1.1.4 d3.selectAll(nodes) | 4 |
| 1.1.5 selection.attr(name[, value]) | 4 |
| 1.1.6 selection.classed(name[, value]) | 5 |
| 1.1.7 selection.style(name[, value[, priority]]) | 6 |
| 1.1.8 selection.property(name[, value]) | 6 |
| 1.1.9 selection.text([value]) | 7 |
| 1.1.10 selection.html([value]) | 8 |
| 1.1.11 selection.append(name) | 9 |
| 1.1.12 selection.insert(name[, before]) | 9 |
| 1.1.13 selection.remove() | 10 |
| 1.1.14 selection.data([values[, key]]) | 10 |
| 1.1.15 selection.enter() | 12 |
| 1.1.16 selection.exit() | 14 |
| 1.1.17 selection.filter(selector) | 15 |
| 1.1.18 selection.datum([value]) | 15 |
| 1.1.19 selection.sort(comparator) | 16 |
| 1.1.20 selection.on(type[, listener[, capture]]) | 17 |
| 1.1.21 d3.event | 18 |
| 1.1.22 d3.mouse(container) | 19 |
| 1.1.23 selection.transition() | 19 |
| 1.1.24 selection.select(selector) | 19 |
| 1.1.25 selection.selectAll(selector) | 20 |
| 1.1.26 selection.each(function) | 21 |
| 1.1.27 selection.call(function[, arguments...]) | 22 |
| 1.1.28 selection.empty() | 23 |
| 1.1.29 selection.node() | 23 |
| 1.1.30 selection.size() | 23 |
| 1.2 过渡 | 23 |
| 1.2.1 transition.delay([delay]) | 23 |
| 1.2.2 transition.duration([duration]) | 24 |
| 1.2.3 transition.ease([value[, arguments]]) | 25 |

| | | |
|--------|--|----|
| 1.2.4 | transition.attr(name, value) | 25 |
| 1.2.5 | transition.attrTween(name, tween) | 26 |
| 1.2.6 | transition.style(name, value[, priority]) | 26 |
| 1.2.7 | transition.styleTween(name, tween[, priority]) | 27 |
| 1.2.8 | transition.text(value) | 27 |
| 1.2.9 | transition.tween(name, factory) | 28 |
| 1.2.10 | transition.remove() | 28 |
| 1.2.11 | transition.select(selector) | 29 |
| 1.2.12 | transition.selectAll(selector) | 29 |
| 1.2.13 | transition.filter(selector) | 30 |
| 1.2.14 | transition.transition() | 30 |
| 1.2.15 | transition.each([type,]listener) | 30 |
| 1.2.16 | transition.call(function[, arguments···]) | 31 |
| 1.2.17 | transition.empty() | 32 |
| 1.2.18 | transition.node() | 32 |
| 1.2.19 | transition.size() | 32 |
| 1.2.20 | d3.ease(type[, arguments···]) | 32 |
| 1.2.21 | ease(t) | 33 |
| 1.2.22 | d3.timer(function[, delay[, time]]) | 33 |
| 1.2.23 | d3.interpolate(a, b) | 34 |
| 1.2.24 | interpolate(t) | 34 |
| 1.2.25 | d3.interpolateNumber(a, b) | 35 |
| 1.2.26 | d3.interpolateRound(a, b) | 35 |
| 1.2.27 | d3.interpolateString(a, b) | 35 |
| 1.2.28 | d3.interpolateRgb(a, b) | 36 |
| 1.2.29 | d3.interpolateHsl(a, b) | 36 |
| 1.2.30 | d3.interpolateLab(a, b) | 36 |
| 1.2.31 | d3.interpolateHcl(a, b) | 36 |
| 1.2.32 | d3.interpolateArray(a, b) | 37 |
| 1.2.33 | d3.interpolateObject(a, b) | 37 |
| 1.2.34 | d3.interpolateTransform(a, b) | 37 |
| 1.2.35 | d3.interpolateZoom(a, b) | 38 |
| 1.3 | 数组 | 39 |
| 1.3.1 | d3.ascending(a, b) | 39 |
| 1.3.2 | d3.descending(a, b) | 40 |
| 1.3.3 | d3.min(array[, accessor]) | 40 |
| 1.3.4 | d3.max(array[, accessor]) | 41 |
| 1.3.5 | d3.extent(array[, accessor]) | 41 |
| 1.3.6 | d3.sum(array[, accessor]) | 41 |
| 1.3.7 | d3.mean(array[, accessor]) | 41 |
| 1.3.8 | d3.median(array[, accessor]) | 42 |
| 1.3.9 | d3.quantile(numbers, p) | 42 |
| 1.3.10 | d3.bisectLeft(array, x[, lo[, hi]]) | 42 |
| 1.3.11 | d3.bisectRight(array, x[, lo[, hi]]) | 42 |
| 1.3.12 | d3.bisect(array, x[, lo[, hi]]) | 43 |
| 1.3.13 | d3.bisector(accessor) | 43 |
| 1.3.14 | d3.shuffle(array) | 44 |
| 1.3.15 | d3.keys(object) | 44 |
| 1.3.16 | d3.values(object) | 44 |
| 1.3.17 | d3.entries(object) | 44 |
| 1.3.18 | d3.map([object]) | 44 |
| 1.3.19 | map.has(key) | 45 |
| 1.3.20 | map.get(key) | 45 |

| | | |
|--------|--|----|
| 1.3.21 | map.set(key, value) | 45 |
| 1.3.22 | map.remove(key) | 45 |
| 1.3.23 | map.keys() | 45 |
| 1.3.24 | map.values() | 45 |
| 1.3.25 | map.entries() | 46 |
| 1.3.26 | map.forEach(function) | 46 |
| 1.3.27 | map.empty() | 46 |
| 1.3.28 | map.size() | 46 |
| 1.3.29 | d3.set([array]) | 46 |
| 1.3.30 | set.has(value) | 47 |
| 1.3.31 | set.add(value) | 47 |
| 1.3.32 | set.remove(value) | 47 |
| 1.3.33 | set.values() | 47 |
| 1.3.34 | set.forEach(function) | 47 |
| 1.3.35 | set.empty() | 47 |
| 1.3.36 | set.size() | 47 |
| 1.3.37 | d3.merge(arrays) | 48 |
| 1.3.38 | d3.range([start,]stop[, step]) | 48 |
| 1.3.39 | d3.permute(array, indexes) | 48 |
| 1.3.40 | d3.zip(arrays···) | 48 |
| 1.3.41 | d3.transpose(matrix) | 49 |
| 1.3.42 | d3.pairs(array) | 49 |
| 1.3.43 | d3.nest() | 49 |
| 1.3.44 | nest.key(function) | 50 |
| 1.3.45 | nest.sortKeys(comparator) | 50 |
| 1.3.46 | nest.sortValues(comparator) | 51 |
| 1.3.47 | nest.rollup(function) | 51 |
| 1.3.48 | nest.map(array[, mapType]) | 52 |
| 1.3.49 | nest.entries(array) | 53 |
| 1.4 | 数学 | 54 |
| 1.4.1 | d3.random.normal([mean, [deviation]]) | 54 |
| 1.4.2 | d3.random.logNormal([mean, [deviation]]) | 54 |
| 1.4.3 | d3.random.bates(count) | 55 |
| 1.4.4 | d3.random.irwinHall(count) | 56 |
| 1.4.5 | d3.transform(string) | 56 |
| 1.4.6 | transform.rotate | 57 |
| 1.4.7 | transform.translate | 57 |
| 1.4.8 | transform.skew | 57 |
| 1.4.9 | transform.scale | 57 |
| 1.4.10 | transform.toString | 57 |
| 1.5 | 请求 | 57 |
| 1.5.1 | d3.xhr(url[, mimeType][, callback]) | 57 |
| 1.5.2 | xhr.header(name[, value]) | 58 |
| 1.5.3 | xhr.mimeType([type]) | 59 |
| 1.5.4 | xhr.responseType(type) | 60 |
| 1.5.5 | xhr.response(value) | 61 |
| 1.5.6 | xhr.get([callback]) | 61 |
| 1.5.7 | xhr.post([data][, callback]) | 61 |
| 1.5.8 | xhr.send(method[, data][, callback]) | 62 |
| 1.5.9 | xhr.abort() | 62 |
| 1.5.10 | xhr.on(type[, listener]) | 63 |
| 1.5.11 | d3.text(url[, mimeType][, callback]) | 63 |
| 1.5.12 | d3.json(url[, callback]) | 64 |
| 1.5.13 | d3.xml(url[, mimeType][, callback]) | 65 |

| | | |
|--------|--------------------------------------|----|
| 1.5.14 | d3.html(url[, callback]) | 66 |
| 1.5.15 | d3.csv(url[, accessor][, callback]) | 66 |
| 1.5.16 | d3.tsv(url[, accessor][, callback]) | 67 |
| 1.6 | 格式化 | 68 |
| 1.6.1 | d3.format(specifier) | 68 |
| 1.6.2 | d3.formatPrefix(value[, precision]) | 70 |
| 1.6.3 | d3.round(x[, n]) | 70 |
| 1.6.4 | d3.requote(string) | 71 |
| 1.7 | 本地化 | 71 |
| 1.7.1 | d3.locale(definition) | 71 |
| 1.7.2 | locale.numberFormat(specifier) | 72 |
| 1.7.3 | locale.timeFormat(specifier) | 72 |
| 1.7.4 | locale.timeFormat.utc(specifier) | 72 |
| 第2章 | 比例尺 (Scales) | 74 |
| 2.1 | 线性比例尺 | 74 |
| 2.1.1 | d3.scale.linear() | 75 |
| 2.1.2 | linear(x) | 75 |
| 2.1.3 | linear.invert(y) | 75 |
| 2.1.4 | linear.domain([numbers]) | 76 |
| 2.1.5 | linear.range([values]) | 76 |
| 2.1.6 | linear.rangeRound(values) | 77 |
| 2.1.7 | linear.interpolate([factory]) | 78 |
| 2.1.8 | linear.clamp([boolean]) | 78 |
| 2.1.9 | linear.nice([count]) | 79 |
| 2.1.10 | linear.ticks([count]) | 80 |
| 2.1.11 | linear.tickFormat(count, [format]) | 80 |
| 2.1.12 | linear.copy() | 80 |
| 2.2 | 恒等比例尺 | 81 |
| 2.2.1 | d3.scale.identity() | 82 |
| 2.2.2 | identity.invert(x) | 82 |
| 2.2.3 | identity.invert(y) | 82 |
| 2.2.4 | identity.domain([numbers]) | 82 |
| 2.2.5 | identity.range([numbers]) | 82 |
| 2.2.6 | identity.ticks([count]) | 83 |
| 2.2.7 | identity.tickFormat(count, [format]) | 83 |
| 2.2.8 | identity.copy() | 83 |
| 2.3 | 乘方比例尺 | 83 |
| 2.3.1 | d3.scale.sqrt() | 84 |
| 2.3.2 | d3.scale.pow() | 85 |
| 2.3.3 | pow(x) | 85 |
| 2.3.4 | pow.invert(y) | 85 |
| 2.3.5 | pow.domain([numbers]) | 85 |
| 2.3.6 | pow.range([values]) | 86 |
| 2.3.7 | pow.rangeRound(values) | 87 |
| 2.3.8 | pow.exponent([k]) | 87 |
| 2.3.9 | pow.interpolate([factory]) | 87 |
| 2.3.10 | pow.clamp([boolean]) | 88 |
| 2.3.11 | pow.nice([m]) | 88 |
| 2.3.12 | pow.ticks([count]) | 89 |
| 2.3.13 | pow.tickFormat([count, [format]]) | 89 |
| 2.3.14 | pow.copy() | 89 |
| 2.4 | 对数比例尺 | 90 |
| 2.4.1 | d3.scale.log() | 91 |
| 2.4.2 | log(x) | 91 |

| | | |
|--------|--|-----|
| 2.4.3 | log.invert(y) | 91 |
| 2.4.4 | log.domain([numbers]) | 91 |
| 2.4.5 | log.range([values]) | 92 |
| 2.4.6 | log.rangeRound(values) | 92 |
| 2.4.7 | log.interpolate([factory]) | 93 |
| 2.4.8 | log.clamp([boolean]) | 93 |
| 2.4.9 | log.nice() | 94 |
| 2.4.10 | log.ticks() | 94 |
| 2.4.11 | log.tickFormat([count, [format]]) | 94 |
| 2.4.12 | log.copy() | 95 |
| 2.5 | 量化比例尺 | 95 |
| 2.5.1 | d3.scale.quantize() | 96 |
| 2.5.2 | quantize(x) | 96 |
| 2.5.3 | quantize.invertExtent(y) | 96 |
| 2.5.4 | quantize.domain([numbers]) | 97 |
| 2.5.5 | quantize.range([values]) | 97 |
| 2.5.6 | quantize.copy() | 97 |
| 2.6 | 分位数比例尺 | 97 |
| 2.6.1 | d3.scale.quantile() | 99 |
| 2.6.2 | quantile(x) | 99 |
| 2.6.3 | quantile.invertExtent(y) | 99 |
| 2.6.4 | quantile.domain([numbers]) | 99 |
| 2.6.5 | quantile.range([values]) | 99 |
| 2.6.6 | quantile.quantiles() | 100 |
| 2.6.7 | quantile.copy() | 100 |
| 2.7 | 临界值比例尺 | 100 |
| 2.7.1 | d3.scale.threshold() | 101 |
| 2.7.2 | threshold(x) | 101 |
| 2.7.3 | threshold.invertExtent(y) | 101 |
| 2.7.4 | threshold.domain([domain]) | 102 |
| 2.7.5 | threshold.range([values]) | 102 |
| 2.7.6 | threshold.copy() | 102 |
| 2.8 | 序数比例尺 | 103 |
| 2.8.1 | d3.scale.ordinal() | 104 |
| 2.8.2 | ordinal(x) | 104 |
| 2.8.3 | ordinal.domain([values]) | 104 |
| 2.8.4 | ordinal.range([values]) | 105 |
| 2.8.5 | ordinal.rangePoints(interval[, padding]) | 105 |
| 2.8.6 | ordinal.rangeBands (interval[, padding[, outerPadding]]) | 106 |
| 2.8.7 | ordinal.rangeRoundBands (interval[, padding[, outerPadding]]) | 107 |
| 2.8.8 | ordinal.rangeBand() | 108 |
| 2.8.9 | ordinal.rangeExtent() | 108 |
| 2.8.10 | ordinal.copy() | 108 |
| 2.8.11 | d3.scale.category10() | 108 |
| 2.8.12 | d3.scale.category20() | 109 |
| 2.8.13 | d3.scale.category20b() | 110 |
| 2.8.14 | d3.scale.category20c() | 110 |
| 2.9 | 时间比例尺 | 111 |
| 2.9.1 | d3.time.scale() | 111 |
| 2.9.2 | d3.time.scale.utc() | 111 |
| 2.9.3 | scale(x) | 112 |
| 2.9.4 | scale.invert(y) | 112 |

| | |
|---|-----|
| 2.9.5 scale.domain([dates]) | 112 |
| 2.9.6 scale.nice([interval[, step]]) | 112 |
| 2.9.7 scale.nice([count]) | 113 |
| 2.9.8 scale.range([values]) | 114 |
| 2.9.9 scale.rangeRound([values]) | 114 |
| 2.9.10 scale.interpolate([factory]) | 114 |
| 2.9.11 scale.clamp([boolean]) | 115 |
| 2.9.12 scale.ticks([interval[, step]]) | 115 |
| 2.9.13 scale.ticks([count]) | 116 |
| 2.9.14 scale.copy() | 116 |
| 第3章 可缩放矢量图形 (SVG) | 117 |
| 3.1 SVG 元素 | 117 |
| 3.1.1 svg:rect x="0" y="0" width="0" height="0" rx="0" ry="0" | 117 |
| 3.1.2 svg:circle cx="0" cy="0" r="0" | 118 |
| 3.1.3 svg:ellipse cx="0" cy="0" rx="0" ry="0" | 119 |
| 3.1.4 svg:line x1="0" y1="0" x2="0" y2="0" | 119 |
| 3.1.5 svg:polyline points="" | 120 |
| 3.1.6 svg:polygon points="" | 120 |
| 3.1.7 svg:text x="0" y="0" dx="0" dy="0" text-anchor="start" | 121 |
| 3.1.8 svg:path d="" transform="" | 122 |
| 3.2 线生成器 | 123 |
| 3.2.1 d3.svg.line() | 123 |
| 3.2.2 line(data) | 123 |
| 3.2.3 line.x([x])和line.y([y]) | 124 |
| 3.2.4 line.interpolate([interpolate]) | 124 |
| 3.2.5 line.tension([tension]) | 125 |
| 3.2.6 line.defined([defined]) | 126 |
| 3.3 径向线生成器 | 126 |
| 3.3.1 d3.svg.line.radial() | 126 |
| 3.3.2 line(data) | 127 |
| 3.3.3 line.radius([radius]) | 127 |
| 3.3.4 line.angle([angle]) | 127 |
| 3.3.5 line.interpolate([interpolate]) | 127 |
| 3.3.6 line.tension([tension]) | 128 |
| 3.3.7 line.defined([defined]) | 128 |
| 3.4 面积生成器 | 128 |
| 3.4.1 d3.svg.area() | 128 |
| 3.4.2 area(data) | 129 |
| 3.4.3 area.x([x]) | 129 |
| 3.4.4 area.y0([y0]) | 130 |
| 3.4.5 area.y1([y1]) | 130 |
| 3.4.6 area.y, area.x0, area.x1 | 130 |
| 3.4.7 area.interpolate([interpolate]) | 131 |
| 3.4.8 area.tension([tension]) | 131 |
| 3.4.9 area.defined([defined]) | 131 |
| 3.5 径向面积生成器 | 131 |
| 3.5.1 d3.svg.area.radial() | 131 |
| 3.5.2 area(data) | 132 |
| 3.5.3 area.outerRadius([radius]) | 132 |
| 3.5.4 area.innerRadius([radius]) | 132 |
| 3.5.5 area.angle([angle]) | 132 |
| 3.6 弧生成器 | 133 |

| | | |
|---------|-----------------------------------|-----|
| 3.6.1 | d3.svg.arc() | 133 |
| 3.6.2 | arc(datum[, index]) | 134 |
| 3.6.3 | arc.innerRadius([radius]) | 134 |
| 3.6.4 | arc.outerRadius([radius]) | 134 |
| 3.6.5 | arc.startAngle([angle]) | 135 |
| 3.6.6 | arc.endAngle([angle]) | 135 |
| 3.6.7 | arc.centroid(arguments···) | 135 |
| 3.7 | 符号生成器 | 136 |
| 3.7.1 | d3.svg.symbol() | 136 |
| 3.7.2 | symbol.type([type]) | 137 |
| 3.7.3 | symbol.size([size]) | 138 |
| 3.7.4 | d3.svg.symbolTypes | 139 |
| 3.8 | 弦生成器 | 139 |
| 3.8.1 | d3.svg.chord() | 139 |
| 3.8.2 | chord(datum[, index]) | 140 |
| 3.8.3 | chord.source([source]) | 140 |
| 3.8.4 | chord.target([target]) | 140 |
| 3.8.5 | chord.startAngle([angle]) | 141 |
| 3.8.6 | chord.endAngle([angle]) | 141 |
| 3.8.7 | chord.radius([radius]) | 141 |
| 3.9 | 对角线生成器 | 142 |
| 3.9.1 | d3.svg.diagonal() | 142 |
| 3.9.2 | diagonal(datum[, index]) | 143 |
| 3.9.3 | diagonal.source([source]) | 143 |
| 3.9.4 | diagonal.target([target]) | 143 |
| 3.9.5 | diagonal.projection([projection]) | 143 |
| 3.9.6 | d3.svg.diagonal.radial() | 144 |
| 3.10 | 轴 | 145 |
| 3.10.1 | d3.svg.axis() | 145 |
| 3.10.2 | axis(selection) | 145 |
| 3.10.3 | axis.scale([scale]) | 145 |
| 3.10.4 | axis.orient([orientation]) | 146 |
| 3.10.5 | axis.ticks([arguments···]) | 147 |
| 3.10.6 | axis.tickValues([values]) | 147 |
| 3.10.7 | axis.tickSize([inner, outer]) | 148 |
| 3.10.8 | axis.innerTickSize([size]) | 148 |
| 3.10.9 | axis.outerTickSize([size]) | 149 |
| 3.10.10 | axis.tickPadding([padding]) | 149 |
| 3.10.11 | axis.tickFormat([format]) | 149 |
| 3.11 | 刷子 | 150 |
| 3.11.1 | d3.svg.brush() | 150 |
| 3.11.2 | brush(selection) | 151 |
| 3.11.3 | brush.x([scale]) | 151 |
| 3.11.4 | brush.y([scale]) | 152 |
| 3.11.5 | brush.extent([values]) | 152 |
| 3.11.6 | brush.clamp([clamp]) | 152 |
| 3.11.7 | brush.clear() | 153 |
| 3.11.8 | brush.empty() | 153 |
| 3.11.9 | brush.on(type[, listener]) | 153 |
| 第4章 | 时间 (Time) | 155 |
| 4.1 | 时间格式 | 155 |
| 4.1.1 | d3.time.format(specifier) | 155 |
| 4.1.2 | format(date) | 157 |
| 4.1.3 | format.parse(string) | 157 |

| | |
|---|-----|
| 4.1.4 d3.time.format.multi(formats) | 157 |
| 4.1.5 d3.time.format.utc(specifier) | 158 |
| 4.1.6 d3.time.format.iso | 158 |
| 4.2 时间间隔 | 159 |
| 4.2.1 d3.time.interval | 159 |
| 4.2.2 interval(date) | 160 |
| 4.2.3 interval.floor(date) | 160 |
| 4.2.4 interval.round(date) | 160 |
| 4.2.5 interval.ceil(date) | 160 |
| 4.2.6 interval.range(start, stop[, step]) | 161 |
| 4.2.7 interval.offset(date, step) | 162 |
| 4.2.8 interval.utc | 162 |
| 4.3 计数 | 162 |
| 第5章 布局 (Layouts) | 164 |
| 5.1 捆绑布局 | 164 |
| 5.1.1 d3.layout.bundle() | 167 |
| 5.1.2 bundle(links) | 167 |
| 5.2 弦布局 | 168 |
| 5.2.1 d3.layout.chord() | 169 |
| 5.2.2 chord.matrix([matrix]) | 169 |
| 5.2.3 chord.padding([padding]) | 169 |
| 5.2.4 chord.sortGroups([comparator]) | 170 |
| 5.2.5 chord.sortSubgroups([comparator]) | 171 |
| 5.2.6 chord.sortChords([comparator]) | 171 |
| 5.2.7 chord.chords() | 172 |
| 5.2.8 chord.groups() | 173 |
| 5.3 簇布局 | 173 |
| 5.3.1 d3.layout.cluster() | 175 |
| 5.3.2 cluster(root) | 175 |
| 5.3.3 cluster.nodes(root) | 175 |
| 5.3.4 cluster.links(nodes) | 176 |
| 5.3.5 cluster.children([children]) | 176 |
| 5.3.6 cluster.sort([comparator]) | 177 |
| 5.3.7 cluster.separation([separation]) | 177 |
| 5.3.8 cluster.size([size]) | 178 |
| 5.3.9 cluster.nodeSize([nodeSize]) | 179 |
| 5.4 力布局 | 180 |
| 5.4.1 d3.layout.force() | 182 |
| 5.4.2 force.size([size]) | 182 |
| 5.4.3 force.linkDistance([distance]) | 182 |
| 5.4.4 force.linkStrength([strength]) | 183 |
| 5.4.5 force.friction([friction]) | 184 |
| 5.4.6 force.charge([charge]) | 184 |
| 5.4.7 force.chargeDistance([distance]) | 185 |
| 5.4.8 force.theta([theta]) | 186 |
| 5.4.9 force.gravity([gravity]) | 186 |
| 5.4.10 force.nodes([nodes]) | 186 |
| 5.4.11 force.links([links]) | 187 |
| 5.4.12 force.start() | 188 |
| 5.4.13 force.alpha([value]) | 188 |
| 5.4.14 force.resume() | 188 |
| 5.4.15 force.stop() | 189 |
| 5.4.16 force.tick() | 189 |
| 5.4.17 force.on(type, listener) | 190 |

| | |
|--|-----|
| 5.4.18 force.drag() | 190 |
| 5.5 层次布局 | 191 |
| 5.6 直方图布局 | 191 |
| 5.6.1 d3.layout.histogram() | 193 |
| 5.6.2 histogram(values[, index]) | 193 |
| 5.6.3 histogram.value([accessor]) | 193 |
| 5.6.4 histogram.range([range]) | 194 |
| 5.6.5 histogram.bins() | 195 |
| 5.6.6 histogram.bins(count) | 195 |
| 5.6.7 histogram.bins(thresholds) | 196 |
| 5.6.8 histogram.bins(function) | 197 |
| 5.6.9 histogram.frequency([frequency]) | 197 |
| 5.7 包布局 | 198 |
| 5.7.1 d3.layout.pack() | 199 |
| 5.7.2 pack(root) | 199 |
| 5.7.3 pack.nodes(root) | 199 |
| 5.7.4 pack.links(nodes) | 199 |
| 5.7.5 pack.children([children]) | 200 |
| 5.7.6 pack.sort([comparator]) | 201 |
| 5.7.7 pack.value([value]) | 202 |
| 5.7.8 pack.size([size]) | 202 |
| 5.7.9 pack.radius([radius]) | 203 |
| 5.7.10 pack.padding([padding]) | 203 |
| 5.8 分区布局 | 204 |
| 5.8.1 d3.layout.partition() | 207 |
| 5.8.2 partition(root) | 207 |
| 5.8.3 partition.nodes(root) | 207 |
| 5.8.4 partition.links(nodes) | 208 |
| 5.8.5 partition.children([children]) | 209 |
| 5.8.6 partition.sort([comparator]) | 209 |
| 5.8.7 partition.value([value]) | 210 |
| 5.8.8 partition.size([size]) | 210 |
| 5.9 饼布局 | 211 |
| 5.9.1 d3.layout.pie() | 212 |
| 5.9.2 pie(values[, index]) | 212 |
| 5.9.3 pie.value([accessor]) | 213 |
| 5.9.4 pie.sort([comparator]) | 214 |
| 5.9.5 pie.startAngle([angle]) | 214 |
| 5.9.6 pie.endAngle([angle]) | 215 |
| 5.10 堆叠布局 | 216 |
| 5.10.1 d3.layout.stack() | 218 |
| 5.10.2 stack(layers[, index]) | 218 |
| 5.10.3 stack.values([accessor]) | 219 |
| 5.10.4 stack.offset([offset]) | 220 |
| 5.10.5 stack.order([order]) | 220 |
| 5.10.6 stack.y([accessor]) | 222 |
| 5.10.7 stack.out([setter]) | 223 |
| 5.11 树布局 | 223 |
| 5.11.1 d3.layout.tree() | 225 |
| 5.11.2 tree(root) | 225 |
| 5.11.3 tree.nodes(root) | 225 |
| 5.11.4 tree.links(nodes) | 226 |
| 5.11.5 tree.children([children]) | 226 |
| 5.11.6 tree.separation([separation]) | 227 |

| | |
|---|-----|
| 5.11.7 tree.size([size]) | 227 |
| 5.11.8 tree.nodeSize([nodeSize]) | 228 |
| 5.11.9 tree.sort([comparator]) | 229 |
| 5.12 矩形树布局 | 230 |
| 5.12.1 d3.layout.treemap()..... | 232 |
| 5.12.2 treemap(root) | 232 |
| 5.12.3 treemap.nodes(root) | 232 |
| 5.12.4 treemap.links(nodes) | 233 |
| 5.12.5 treemap.children([children]) | 233 |
| 5.12.6 treemap.sort([comparator]) | 234 |
| 5.12.7 treemap.value([value]) | 235 |
| 5.12.8 treemap.size([size]) | 235 |
| 5.12.9 treemap.padding([padding]) | 235 |
| 5.12.10 treemap.round([round]) | 236 |
| 5.12.11 treemap.sticky([sticky]) | 236 |
| 5.12.12 treemap.mode([mode]) | 238 |
| 第6章 地理 (Geo) | 240 |
| 6.1 地理路径 | 240 |
| 6.1.1 d3.geo.path() | 241 |
| 6.1.2 path(feature[, index]) | 241 |
| 6.1.3 path.projection([projection]) | 242 |
| 6.1.4 path.context([context]) | 243 |
| 6.1.5 path.centroid(feature) | 244 |
| 6.1.6 path.area(feature) | 245 |
| 6.1.7 path.bounds(feature) | 246 |
| 6.2 经纬网生成器 | 246 |
| 6.2.1 d3.geo.graticule | 247 |
| 6.2.2 graticule() | 247 |
| 6.2.3 graticule.lines() | 247 |
| 6.2.4 graticule.outline() | 248 |
| 6.2.5 graticule.extent(extent) | 249 |
| 6.2.6 graticule.majorExtent(extent)..... | 249 |
| 6.2.7 graticule.minorExtent(extent) | 250 |
| 6.2.8 graticule.step(step) | 250 |
| 6.2.9 graticule.majorStep(step) | 251 |
| 6.2.10 graticule.minorStep(step) | 251 |
| 6.3 球面数学运算 | 252 |
| 6.3.1 d3.geo.area(feature) | 252 |
| 6.3.2 d3.geo.centroid(feature) | 252 |
| 6.3.3 d3.geo.bounds(feature) | 252 |
| 6.3.4 d3.geo.distance(a, b) | 252 |
| 6.3.5 d3.geo.length(feature) | 253 |
| 6.3.6 d3.geo.interpolate(a, b) | 253 |
| 6.4 标准抽象投影 | 254 |
| 6.4.1 d3.geo.projection(raw) | 254 |
| 6.4.2 projection(location) | 254 |
| 6.4.3 projection.invert(point) | 254 |
| 6.4.4 projection.rotate([rotation]) | 254 |
| 6.4.5 projection.center([location]) | 255 |
| 6.4.6 projection.translate([point]) | 256 |
| 6.4.7 projection.scale([scale]) | 256 |
| 6.4.8 projection.clipAngle(angle) | 257 |
| 6.4.9 projection.clipExtent(extent) | 258 |
| 6.5 标准投影 | 258 |

| | | |
|--------|---|-----|
| 6.5.1 | d3.geo.azimuthalEqualArea() | 258 |
| 6.5.2 | d3.geo.azimuthalEquidistant() | 259 |
| 6.5.3 | d3.geo.conicConformal() | 259 |
| 6.5.4 | conicConformal.parallels([parallels]) | 259 |
| 6.5.5 | d3.geo.conicEqualArea() | 260 |
| 6.5.6 | conicEqualArea.parallels([parallels]) | 261 |
| 6.5.7 | d3.geo.conicEquidistant() | 261 |
| 6.5.8 | conicEquidistant.parallels([parallels]) | 261 |
| 6.5.9 | d3.geo.equirectangular() | 262 |
| 6.5.10 | d3.geo.gnomonic() | 262 |
| 6.5.11 | d3.geo.mercator() | 263 |
| 6.5.12 | d3.geo.orthographic() | 264 |
| 6.5.13 | d3.geo.stereographic() | 264 |
| 6.5.14 | d3.geo.transverseMercator() | 264 |
| 6.6 | 流 | 265 |
| 6.6.1 | d3.geo.stream(object, listener) | 265 |
| 6.6.2 | listener.point(x, y[, z]) | 266 |
| 6.6.3 | listener.lineStart() | 266 |
| 6.6.4 | listener.lineEnd() | 266 |
| 6.6.5 | listener.polygonStart() | 266 |
| 6.6.6 | listener.polygonEnd() | 266 |
| 6.6.7 | listener.sphere() | 267 |
| 6.6.8 | d3.geo.transform(methods) | 267 |
| 6.6.9 | transform.stream(listener) | 268 |
| 6.6.10 | d3.geo.clipExtent() | 269 |
| 6.6.11 | clipExtent.extent([extent]) | 269 |
| | 第7章 几何 (Geometry) | 270 |
| 7.1 | 四叉树 | 270 |
| 7.1.1 | d3.geom.quadtree() | 272 |
| 7.1.2 | quadtree(points) | 272 |
| 7.1.3 | root.add(point) | 273 |
| 7.1.4 | root.visit(callback) | 274 |
| 7.1.5 | quadtree.x([x]) | 274 |
| 7.1.6 | quadtree.y([y]) | 275 |
| 7.2 | 凸包 | 275 |
| 7.2.1 | d3.geom.hull() | 276 |
| 7.2.2 | hull(vertices) | 276 |
| 7.2.3 | hull.x([x]) | 276 |
| 7.2.4 | hull.y([y]) | 277 |
| 7.3 | 多边形 | 277 |
| 7.3.1 | d3.geom.polygon(vertices) | 278 |
| 7.3.2 | polygon.area() | 278 |
| 7.3.3 | polygon.centroid() | 278 |
| 7.3.4 | polygon.clip(subject) | 278 |
| 7.4 | 泰森多边形 | 279 |
| 7.4.1 | d3.geom.voronoi() | 280 |
| 7.4.2 | voronoi(data) | 280 |
| 7.4.3 | voronoi.x([x]) | 280 |
| 7.4.4 | voronoi.y([y]) | 281 |
| 7.4.5 | voronoi.clipExtent([extent]) | 282 |
| 7.4.6 | voronoi.links(data) | 283 |
| 7.4.7 | voronoi.triangles(data) | 284 |
| | 第8章 行为 (Behaviors) | 285 |
| 8.1 | 拖曳 | 285 |

| | |
|-----------------------------------|----------------------|
| 8.1.1 d3.behavior.drag() | 286 |
| 8.1.2 drag.on(type[, listener]) | 286 |
| 8.1.3 drag.origin([origin]) | 287 |
| 8.2 缩放 | 287 |
| 8.2.1 d3.behavior.zoom() | 288 |
| 8.2.2 zoom(selection) | 288 |
| 8.2.3 zoom.translate([translate]) | 289 |
| 8.2.4 zoom.scale([scale]) | 289 |
| 8.2.5 zoom.scaleExtent([extent]) | 290 |
| 8.2.6 zoom.center([center]) | 290 |
| 8.2.7 zoom.size([size]) | 290 |
| 8.2.8 zoom.x([x]) | 290 |
| 8.2.9 zoom.y([y]) | 290 |
| 8.2.10 zoom.on(type, listener) | 291 |
| 本书参考资料 | 291 |
| • • • • • | (收起) |

[D3 API详解 \(全彩\) 下载链接1](#)

标签

论D3的重要性

数据可视化

D3js

D3

前端开发

评论

基本翻译一遍，可当参考书查阅。

之前看的，纯文档翻译，不如直接看文档

api的堆砌，适合当字典用。

[D3 API详解（全彩） 下载链接1](#)

书评

[D3 API详解（全彩） 下载链接1](#)