



D3 API 详解

张天旭 魏飞◎编著

“一图胜千言”数据可视化使人们对海量复杂数据的阐释和理解变得事半功倍。D3是当今非常流行的可视化利器。

本书用大量简洁直观的案例详细介绍了D3的使用。全书图文并茂，全彩印刷，力图使读者对D3有深入理解和整体把握。

D3 API 详解

张天旭 魏飞◎编著



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

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。

本书基于官方 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。本书涵盖了 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。

本书适合所有想使用 D3 实现数据可视化方案的人使用。读者朋友可以将本书作为入门参考，也可以作为速查手册。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目 (CIP) 数据

D3 API 详解 / 张天旭, 魏飞编著. —北京: 电子工业出版社, 2016.3
ISBN 978-7-121-27899-0

I. ①D… II. ①张… ②魏… III. ①JAVA 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2015)第 307582 号

策划编辑: 付 睿

责任编辑: 徐津平

特约编辑: 顾慧芳

印 刷: 中国电影出版社印刷厂

装 订: 三河市华成印务有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×980 1/16 印张: 19.5 字数: 446 千字

版 次: 2016 年 3 月第 1 版

印 次: 2016 年 3 月第 1 次印刷

印 数: 3000 册 定价: 89.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前言

数据可视化是用图形图像等方式表现数据内涵的技术。俗话说“一图胜千言”，数据可视化最重要的意义在于让数据变得简单直观。它有效利用了人眼的感知能力和大脑的处理能力，让数据中蕴含的规律一目了然。尤其是在大数据时代，数据可视化已成为一座桥梁，跨越了数据和使用者的鸿沟，减少了在数据分析、交流和传播中的障碍。它使得人们对海量、复杂数据的阐释和理解变得事半功倍。在很多情况下，数据可视化是理解和表达数据的有效手段，有时甚至是唯一的手段。

数据可视化的强大魅力还源于它是美而生动的。只有一堆数据难免枯燥乏味，可视化技术给数据穿上了华丽的外衣，丰富的交互功能增添了数据的灵动性。数据可视化绽放了数据之美。它让数据变得生动鲜活，有效地吸引了人们的注意力，能大大激发人们的好奇心和求知欲。让人们在与数据的交互过程中不仅探索到了知识还收获了乐趣。正因为如此，数据可视化技术蓬勃发展，在互联网、生物、金融、医学、建筑、教育、国防等需要展现数据的行业都有广泛的应用。

工欲善其事必先利其器。D3 是当今最流行的数据可视化工具之一。D3 是一个 JavaScript 库，擅长绘制交互式矢量图表，底层使用了 SVG 绘图技术并通过数据驱动创建文档。D3 提供的 API 主要包括对 HTML 和 SVG 文档的操作、处理多种格式的数据、为特定的可视化布局生成数据，以及提供丰富的交互和动画功能等。通过这些编程接口，我们既不用太关心底层实现，也不会因为封装程度太高而很难修改，所以 D3 比 Processing¹这样的底层绘图库更简单，比 Echarts²这样高度封装的图表库更自由，是一个灵活轻量的前端数据可视化库。而且 D3 不依赖其他的库，你可以充分发挥想象，仅用 D3 就可以创造绚丽多彩的数据可视化作品。当然，到

1 <https://processing.org/>.

2 <http://echarts.baidu.com/index.html>.

底要不要使用 D3 还需要根据具体情况进行选择，表 0.1 可以作为一个技术选型的参考。

表 0.1 D3 的适用范围

考虑因素	适用范围
开源	D3 基于开源协议 BSD-3-Clause ³ ，可以免费用于商业项目。源码托管在 GitHub 上 ⁴ ，截至 2015 年 9 月 26 日 star 数已达 41869 次，fork 数达 11021 次，有大量用户和丰富友好的案例
兼容性	支持 Firefox、Chrome、Safari、Opera、IE9+ 等现代浏览器。支持 Android 和 iOS 平台
功能	D3 擅长于绘制二维 Web 数据可视化矢量图表，支持丰富的交互功能，支持创建自定义可视化方案。但不支持 3D 可视化，并且加载到浏览器的数据量不能太大（10MB 以内较合适）。D3 不是现成的图表库，相比于封装好的图表库学习和开发成本都较高

目前 D3 的官方文档已经十分全面了，涵盖了 600 多个函数，全文接近 6 万个单词。然而，D3 参考文档中的专业术语较多，很多地方表述十分晦涩艰深，原文中的示例代码较少，况且对于初学者来说直接看英文也很吃力。本书笔者为了方便 D3 学习者更好地理解和使用 D3，对 D3 官方 API（应用程序编程接口）⁵进行了详细介绍，还为大部分函数都编写了简单的示例代码。下面，我们先来看看本书涉及的 D3 官方文档主要包含哪些内容，D3 API 总览如表 0.2 所示。

表 0.2 D3 API 总览

内容	简介
核心 (Core)	包括选择器、过渡、数据处理、本地化、颜色等
比例尺 (Scales)	在数据编码和视觉编码之间转换
可缩放矢量图形 (SVG)	提供用于创建可伸缩矢量图形的实用工具
时间 (Time)	解析或格式化时间，计算日历的时间间隔等
布局 (Layouts)	推导定位元素的辅助数据
地理 (Geography)	球面坐标，经纬度运算
几何 (Geometry)	提供绘制 2D 几何图形的实用工具
行为 (Behaviors)	可重用的交互行为

本书覆盖了绝大部分的 API，为方便使用，本书尽量保持与官方文档结构一致，对表 0.2 的每个部分单独成章介绍。本书所用的 D3 版本是 3.5.5，全文案例在 Chrome 浏览器中调试通过。一些简单的函数直接使用浏览器的控制台演示，少部分案例需要使用服务器运行，大部分案例都可以直接打开 HTML 文档来查看效果。对于本书的使用，强烈建议读者朋友们边看文字表述边参考示例代码动手实践。本书的随书源码可在 <https://github.com/tianxuzhang/>

3 <http://opensource.org/licenses/BSD-3-Clause>.

4 <https://github.com/mbostock/d3>.

5 <https://github.com/mbostock/d3/wiki/API-Reference>.

d3-api-demo/archive/master.zip 和 <http://www.broadview.com.cn/27899> 下载使用,通过量的积累最终一定会有质的改变。大家要保持对新技术的兴趣和对理想的热情,在自学的同时多分享多讨论。D3 的学习会是很有成就感的,希望读者可以通过本书感受到数据可视化的魅力,能从中收获快乐。

致谢

首先要感谢章成志教授的启蒙教育,是他让我了解到 D3 这门技术。并且,在学习和生活中给予了我足够的信任和支持。可以说,章老师是我真正的良师益友。

其次,感谢 Mike Bostock 以及其他 D3 贡献者,是他们开发了 D3 这个优秀的数据库,并且提供了大量简单实用的案例。正因为如此,才能让我们有机会领略数据可视化的奇妙。

还要感谢 CSDN 这个平台,CSDN 是中国程序员最常使用的技术网站。通过 CSDN 让我能够与更多数据可视化爱好者相识并共同探讨 D3 相关技术。

感谢电子工业出版社的付睿编辑,没有伯乐就没有千里马,谢谢她给予我这个机会编写本书。感谢顾慧芳编辑,谢谢她为本书辛勤校对和编辑,这有效地保障了本书的质量。

最重要的是要感谢参与 D3 参考文档翻译以及对本书提供帮助的人,他们是:崔博敏、何凯琳、张玉杰、季国亮、张烁、边城、卫学士、翟琰琦、路明非、何丽、谁浮、马语者、现明涟漪、Carry on、陶凜然、李琛婧、赵荣和宋墩江以及其他关心和支持我的人,和你们的讨论和交流是我最温暖的回忆。

最后,也是最应该感谢的是一直与我并肩奋斗的魏飞,感谢那些安静得只剩下键盘声的夜晚,每一个奋斗的日子都值得深深铭记。

由于作者水平有限,书中难免有错误之处,我们渴望得到您的反馈,如果发现问题请发送邮件至 zhang_tianxu@sina.com,或者在新浪微博上@D3 数据可视化给我们批评指正。

张天旭

2015 年 9 月

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	1.3.9	d3.quantile(numbers, p).....	42
1.2.13	transition.filter(selector).....	30	1.3.10	d3.bisectLeft(array, x[, lo[, hi]])	42
1.2.14	transition.transition().....	30	1.3.11	d3.bisectRight(array, x[, lo[, hi]])	42
1.2.15	transition.each([type,]listener)	30	1.3.12	d3.bisect(array, x[, lo[, hi]]).....	43
1.2.16	transition.call(function[, arguments...]).....	31	1.3.13	d3.bisector(accessor)	43
1.2.17	transition.empty().....	32	1.3.14	d3.shuffle(array)	44
1.2.18	transition.node().....	32	1.3.15	d3.keys(object)	44
1.2.19	transition.size()	32	1.3.16	d3.values(object)	44
1.2.20	d3.ease(type[, arguments...]).....	32	1.3.17	d3.entries(object).....	44
1.2.21	ease(t)	33	1.3.18	d3.map([object])	44
1.2.22	d3.timer(function[, delay[, time]])	33	1.3.19	map.has(key)	45
1.2.23	d3.interpolate(a, b).....	34	1.3.20	map.get(key).....	45
1.2.24	interpolate(t)	34	1.3.21	map.set(key, value).....	45
1.2.25	d3.interpolateNumber(a, b).....	35	1.3.22	map.remove(key).....	45
1.2.26	d3.interpolateRound(a, b)	35	1.3.23	map.keys()	45
1.2.27	d3.interpolateString(a, b)	35	1.3.24	map.values()	45
1.2.28	d3.interpolateRgb(a, b)	36	1.3.25	map.entries().....	46
1.2.29	d3.interpolateHsl(a, b)	36	1.3.26	map.forEach(function).....	46
1.2.30	d3.interpolateLab(a, b).....	36	1.3.27	map.empty()	46
1.2.31	d3.interpolateHcl(a, b)	36	1.3.28	map.size()	46
1.2.32	d3.interpolateArray(a, b).....	37	1.3.29	d3.set([array])	46
1.2.33	d3.interpolateObject(a, b)	37	1.3.30	set.has(value).....	47
1.2.34	d3.interpolateTransform(a, b)	37	1.3.31	set.add(value)	47
1.2.35	d3.interpolateZoom(a, b).....	38	1.3.32	set.remove(value)	47
1.3	数组	39	1.3.33	set.values().....	47
1.3.1	d3.ascending(a, b).....	39	1.3.34	set.forEach(function)	47
1.3.2	d3.descending(a, b).....	40	1.3.35	set.empty().....	47
1.3.3	d3.min(array[, accessor])	40	1.3.36	set.size()	47
1.3.4	d3.max(array[, accessor]).....	41	1.3.37	d3.merge(arrays).....	48
1.3.5	d3.extent(array[, accessor]).....	41	1.3.38	d3.range([start,]stop[, step])	48
1.3.6	d3.sum(array[, accessor]).....	41	1.3.39	d3.permute(array, indexes)	48
1.3.7	d3.mean(array[, accessor]).....	41	1.3.40	d3.zip(arrays...).....	48
1.3.8	d3.median(array[, accessor]).....	42	1.3.41	d3.transpose(matrix)	49

1.3.42	d3.pairs(array).....	49	1.5.14	d3.html(url[, callback]).....	66
1.3.43	d3.nest().....	49	1.5.15	d3.csv(url[, accessor][, callback]).....	66
1.3.44	nest.key(function).....	50	1.5.16	d3.tsv(url[, accessor][, callback]).....	67
1.3.45	nest.sortKeys(comparator).....	50	1.6 格式化.....	68	
1.3.46	nest.sortValues(comparator).....	51	1.6.1	d3.format(specifier).....	68
1.3.47	nest.rollup(function).....	51	1.6.2	d3.formatPrefix(value[, precision]).....	70
1.3.48	nest.map(array[, mapType]).....	52	1.6.3	d3.round(x[, n]).....	70
1.3.49	nest.entries(array).....	53	1.6.4	d3.requote(string).....	71
1.4 数学.....	54		1.7 本地化.....	71	
1.4.1	d3.random.normal([mean, [deviation]]).....	54	1.7.1	d3.locale(definition).....	71
1.4.2	d3.random.logNormal([mean, [deviation]]).....	54	1.7.2	locale.numberFormat(specifier).....	72
1.4.3	d3.random.bates(count).....	55	1.7.3	locale.timeFormat(specifier).....	72
1.4.4	d3.random.irwinHall(count).....	56	1.7.4	locale.timeFormat.utc(specifier).....	72
1.4.5	d3.transform(string).....	56	第 2 章 比例尺 (Scales).....	74	
1.4.6	transform.rotate.....	57	2.1 线性比例尺.....	74	
1.4.7	transform.translate.....	57	2.1.1	d3.scale.linear().....	75
1.4.8	transform.skew.....	57	2.1.2	linear(x).....	75
1.4.9	transform.scale.....	57	2.1.3	linear.invert(y).....	75
1.4.10	transform.toString.....	57	2.1.4	linear.domain([numbers]).....	76
1.5 请求.....	57		2.1.5	linear.range([values]).....	76
1.5.1	d3.xhr(url[, mimeType][, callback]).....	57	2.1.6	linear.rangeRound(values).....	77
1.5.2	xhr.header(name[, value]).....	58	2.1.7	linear.interpolate([factory]).....	78
1.5.3	xhr.mimeType([type]).....	59	2.1.8	linear.clamp([boolean]).....	78
1.5.4	xhr.responseType(type).....	60	2.1.9	linear.nice([count]).....	79
1.5.5	xhr.response(value).....	61	2.1.10	linear.ticks([count]).....	80
1.5.6	xhr.get([callback]).....	61	2.1.11	linear.tickFormat(count, [format]).....	80
1.5.7	xhr.post([data][, callback]).....	61	2.1.12	linear.copy().....	80
1.5.8	xhr.send(method[, data][, callback]).....	62	2.2 恒等比例尺.....	81	
1.5.9	xhr.abort().....	62	2.2.1	d3.scale.identity().....	82
1.5.10	xhr.on(type[, listener]).....	63	2.2.2	identity.invert(x).....	82
1.5.11	d3.text(url[, mimeType][, callback]).....	63	2.2.3	identity.invert(y).....	82
1.5.12	d3.json(url[, callback]).....	64			
1.5.13	d3.xml(url[, mimeType][, callback]).....	65			

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	text-anchor="start"	121	
2.8.9	ordinal.rangeExtent()	108	3.1.8	svg:path d="" transform=""	122
2.8.10	ordinal.copy()	108	3.2 线生成器	123	
2.8.11	d3.scale.category10()	108	3.2.1	d3.svg.line()	123
2.8.12	d3.scale.category20()	109	3.2.2	line(data)	123
2.8.13	d3.scale.category20b()	110	3.2.3	line.x([x])和 line.y([y])	124
2.8.14	d3.scale.category20c()	110	3.2.4	line.interpolate([interpolate])	124
2.9 时间比例尺	111		3.2.5	line.tension([tension])	125
2.9.1	d3.time.scale()	111	3.2.6	line.defined([defined])	126
2.9.2	d3.time.scale.utc()	111	3.3 径向线生成器	126	
2.9.3	scale(x)	112	3.3.1	d3.svg.line.radial()	126
2.9.4	scale.invert(y)	112	3.3.2	line(data)	127
2.9.5	scale.domain([dates])	112	3.3.3	line.radius([radius])	127
2.9.6	scale.nice([interval[, step]])	112	3.3.4	line.angle([angle])	127
2.9.7	scale.nice([count])	113	3.3.5	line.interpolate([interpolate])	127
2.9.8	scale.range([values])	114	3.3.6	line.tension([tension])	128
2.9.9	scale.rangeRound([values])	114	3.3.7	line.defined([defined])	128
2.9.10	scale.interpolate([factory])	114	3.4 面积生成器	128	
2.9.11	scale.clamp([boolean])	115	3.4.1	d3.svg.area()	128
2.9.12	scale.ticks([interval[, step]])	115	3.4.2	area(data)	129
2.9.13	scale.ticks([count])	116	3.4.3	area.x([x])	129
2.9.14	scale.copy()	116	3.4.4	area.y0([y0])	130
第3章 可缩放矢量图形 (SVG) 117			3.4.5	area.y1([y1])	130
3.1 SVG 元素	117		3.4.6	area.y, area.x0, area.x1	130
3.1.1	svg:rect x="0" y="0" width="0" height="0" rx="0" ry="0"	117	3.4.7	area.interpolate([interpolate])	131
3.1.2	svg:circle cx="0" cy="0" r="0"	118	3.4.8	area.tension([tension])	131
3.1.3	svg:ellipse cx="0" cy="0" rx="0" ry="0"	119	3.4.9	area.defined([defined])	131
3.1.4	svg:line x1="0" y1="0" x2="0" y2="0"	119	3.5 径向面积生成器	131	
3.1.5	svg:polyline points=""	120	3.5.1	d3.svg.area.radial()	131
3.1.6	svg:polygon points=""	120	3.5.2	area(data)	132
3.1.7	svg:text x="0" y="0" dx="0" dy="0"		3.5.3	area.outerRadius([radius])	132
			3.5.4	area.innerRadius([radius])	132

3.5.5	area.angle([angle])	132	3.10.3	axis.scale([scale])	145
3.6	弧生成器	133	3.10.4	axis.orient([orientation])	146
3.6.1	d3.svg.arc()	133	3.10.5	axis.ticks([arguments...])	147
3.6.2	arc(datum[, index])	134	3.10.6	axis.tickValues([values])	147
3.6.3	arc.innerRadius([radius])	134	3.10.7	axis.tickSize([inner, outer])	148
3.6.4	arc.outerRadius([radius])	134	3.10.8	axis.innerTickSize([size])	148
3.6.5	arc.startAngle([angle])	135	3.10.9	axis.outerTickSize([size])	149
3.6.6	arc.endAngle([angle])	135	3.10.10	axis.tickPadding([padding])	149
3.6.7	arc.centroid(arguments...)	135	3.10.11	axis.tickFormat([format])	149
3.7	符号生成器	136	3.11	刷子	150
3.7.1	d3.svg.symbol()	136	3.11.1	d3.svg.brush()	150
3.7.2	symbol.type([type])	137	3.11.2	brush(selection)	151
3.7.3	symbol.size([size])	138	3.11.3	brush.x([scale])	151
3.7.4	d3.svg.symbolTypes	139	3.11.4	brush.y([scale])	152
3.8	弦生成器	139	3.11.5	brush.extent([values])	152
3.8.1	d3.svg.chord()	139	3.11.6	brush.clamp([clamp])	152
3.8.2	chord(datum[, index])	140	3.11.7	brush.clear()	153
3.8.3	chord.source([source])	140	3.11.8	brush.empty()	153
3.8.4	chord.target([target])	140	3.11.9	brush.on(type[, listener])	153
3.8.5	chord.startAngle([angle])	141	第 4 章 时间 (Time)	155	
3.8.6	chord.endAngle([angle])	141	4.1 时间格式	155	
3.8.7	chord.radius([radius])	141	4.1.1	d3.time.format(specifier)	155
3.9	对角线生成器	142	4.1.2	format(date)	157
3.9.1	d3.svg.diagonal()	142	4.1.3	format.parse(string)	157
3.9.2	diagonal(datum[, index])	143	4.1.4	d3.time.format.multi(formats)	157
3.9.3	diagonal.source([source])	143	4.1.5	d3.time.format.utc(specifier)	158
3.9.4	diagonal.target([target])	143	4.1.6	d3.time.format.iso	158
3.9.5	diagonal.projection([projection])	143	4.2 时间间隔	159	
3.9.6	d3.svg.diagonal.radial()	144	4.2.1	d3.time.interval	159
3.10	轴	145	4.2.2	interval(date)	160
3.10.1	d3.svg.axis()	145	4.2.3	interval.floor(date)	160
3.10.2	axis(selection)	145			

4.2.4	interval.round(date).....	160	5.4.2	force.size([size]).....	182
4.2.5	interval.ceil(date).....	160	5.4.3	force.linkDistance([distance]).....	182
4.2.6	interval.range(start, stop[, step]).....	161	5.4.4	force.linkStrength([strength]).....	183
4.2.7	interval.offset(date, step).....	162	5.4.5	force.friction([friction]).....	184
4.2.8	interval.utc.....	162	5.4.6	force.charge([charge]).....	184
4.3	计数.....	162	5.4.7	force.chargeDistance([distance]).....	185
第5章	布局 (Layouts).....	164	5.4.8	force.theta([theta]).....	186
5.1	捆绑布局.....	164	5.4.9	force.gravity([gravity]).....	186
5.1.1	d3.layout.bundle().....	167	5.4.10	force.nodes([nodes]).....	186
5.1.2	bundle(links).....	167	5.4.11	force.links([links]).....	187
5.2	弦布局.....	168	5.4.12	force.start().....	188
5.2.1	d3.layout.chord().....	169	5.4.13	force.alpha([value]).....	188
5.2.2	chord.matrix([matrix]).....	169	5.4.14	force.resume().....	188
5.2.3	chord.padding([padding]).....	169	5.4.15	force.stop().....	189
5.2.4	chord.sortGroups([comparator]).....	170	5.4.16	force.tick().....	189
5.2.5	chord.sortSubgroups([comparator]).....	171	5.4.17	force.on(type, listener).....	190
5.2.6	chord.sortChords([comparator]).....	171	5.4.18	force.drag().....	190
5.2.7	chord.chords().....	172	5.5	层次布局.....	191
5.2.8	chord.groups().....	173	5.6	直方图布局.....	191
5.3	簇布局.....	173	5.6.1	d3.layout.histogram().....	193
5.3.1	d3.layout.cluster().....	175	5.6.2	histogram(values[, index]).....	193
5.3.2	cluster(root).....	175	5.6.3	histogram.value([accessor]).....	193
5.3.3	cluster.nodes(root).....	175	5.6.4	histogram.range([range]).....	194
5.3.4	cluster.links(nodes).....	176	5.6.5	histogram.bins().....	195
5.3.5	cluster.children([children]).....	176	5.6.6	histogram.bins(count).....	195
5.3.6	cluster.sort([comparator]).....	177	5.6.7	histogram.bins(thresholds).....	196
5.3.7	cluster.separation([separation]).....	177	5.6.8	histogram.bins(function).....	197
5.3.8	cluster.size([size]).....	178	5.6.9	histogram.frequency([frequency]).....	197
5.3.9	cluster.nodeSize([nodeSize]).....	179	5.7	包布局.....	198
5.4	力布局.....	180	5.7.1	d3.layout.pack().....	199
5.4.1	d3.layout.force().....	182	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	5.11.2	tree(root)	225
5.7.6	pack.sort([comparator])	201	5.11.3	tree.nodes(root)	225
5.7.7	pack.value([value])	202	5.11.4	tree.links(nodes)	226
5.7.8	pack.size([size])	202	5.11.5	tree.children([children])	226
5.7.9	pack.radius([radius])	203	5.11.6	tree.separation([separation])	227
5.7.10	pack.padding([padding])	203	5.11.7	tree.size([size])	227
5.8	分区布局	204	5.11.8	tree.nodeSize([nodeSize])	228
5.8.1	d3.layout.partition()	207	5.11.9	tree.sort([comparator])	229
5.8.2	partition(root)	207	5.12	矩形树布局	230
5.8.3	partition.nodes(root)	207	5.12.1	d3.layout.treemap()	232
5.8.4	partition.links(nodes)	208	5.12.2	treemap(root)	232
5.8.5	partition.children([children])	209	5.12.3	treemap.nodes(root)	232
5.8.6	partition.sort([comparator])	209	5.12.4	treemap.links(nodes)	233
5.8.7	partition.value([value])	210	5.12.5	treemap.children([children])	233
5.8.8	partition.size([size])	210	5.12.6	treemap.sort([comparator])	234
5.9	饼布局	211	5.12.7	treemap.value([value])	235
5.9.1	d3.layout.pie()	212	5.12.8	treemap.size([size])	235
5.9.2	pie(values[, index])	212	5.12.9	treemap.padding([padding])	235
5.9.3	pie.value([accessor])	213	5.12.10	treemap.round([round])	236
5.9.4	pie.sort([comparator])	214	5.12.11	treemap.sticky([sticky])	236
5.9.5	pie.startAngle([angle])	214	5.12.12	treemap.mode([mode])	238
5.9.6	pie.endAngle([angle])	215	第 6 章	地理 (Geo)	240
5.10	堆叠布局	216	6.1	地理路径	240
5.10.1	d3.layout.stack()	218	6.1.1	d3.geo.path()	241
5.10.2	stack(layers[, index])	218	6.1.2	path(feature[, index])	241
5.10.3	stack.values([accessor])	219	6.1.3	path.projection([projection])	242
5.10.4	stack.offset([offset])	220	6.1.4	path.context([context])	243
5.10.5	stack.order([order])	220	6.1.5	path.centroid(feature)	244
5.10.6	stack.y([accessor])	222	6.1.6	path.area(feature)	245
5.10.7	stack.out([setter])	223	6.1.7	path.bounds(feature)	246
5.11	树布局	223	6.2	经纬网生成器	246
5.11.1	d3.layout.tree()	225			

6.2.1	d3.geo.graticule.....	247	6.5.5	d3.geo.conicEqualArea().....	260
6.2.2	graticule().....	247	6.5.6	conicEqualArea.parallels([parallels])	261
6.2.3	graticule.lines()	247	6.5.7	d3.geo.conicEquidistant().....	261
6.2.4	graticule.outline().....	248	6.5.8	conicEquidistant.parallels([parallels])	261
6.2.5	graticule.extent(extent)	249	6.5.9	d3.geo.equirectangular().....	262
6.2.6	graticule.majorExtent(extent).....	249	6.5.10	d3.geo.gnomonic().....	262
6.2.7	graticule.minorExtent(extent)	250	6.5.11	d3.geo.mercator()	263
6.2.8	graticule.step(step)	250	6.5.12	d3.geo.orthographic()	264
6.2.9	graticule.majorStep(step)	251	6.5.13	d3.geo.stereographic()	264
6.2.10	graticule.minorStep(step)	251	6.5.14	d3.geo.transverseMercator()	264
6.3	球面数学运算	252	6.6	流	265
6.3.1	d3.geo.area(feature)	252	6.6.1	d3.geo.stream(object, listener).....	265
6.3.2	d3.geo.centroid(feature)	252	6.6.2	listener.point(x, y[, z])	266
6.3.3	d3.geo.bounds(feature)	252	6.6.3	listener.lineStart()	266
6.3.4	d3.geo.distance(a, b)	252	6.6.4	listener.lineEnd()	266
6.3.5	d3.geo.length(feature)	253	6.6.5	listener.polygonStart()	266
6.3.6	d3.geo.interpolate(a, b)	253	6.6.6	listener.polygonEnd()	266
6.4	标准抽象投影	254	6.6.7	listener.sphere()	267
6.4.1	d3.geo.projection(raw)	254	6.6.8	d3.geo.transform(methods).....	267
6.4.2	projection(location).....	254	6.6.9	transform.stream(listener).....	268
6.4.3	projection.invert(point)	254	6.6.10	d3.geo.clipExtent()	269
6.4.4	projection.rotate([rotation]).....	254	6.6.11	clipExtent.extent([extent])	269
6.4.5	projection.center([location])	255	第 7 章 几何 (Geometry)	270	
6.4.6	projection.translate([point])	256	7.1	四叉树	270
6.4.7	projection.scale([scale])	256	7.1.1	d3.geom.quadtree().....	272
6.4.8	projection.clipAngle(angle)	257	7.1.2	quadtree(points).....	272
6.4.9	projection.clipExtent(extent).....	258	7.1.3	root.add(point)	273
6.5	标准投影	258	7.1.4	root.visit(callback).....	274
6.5.1	d3.geo.azimuthalEqualArea()	258	7.1.5	quadtree.x([x]).....	274
6.5.2	d3.geo.azimuthalEquidistant()	259	7.1.6	quadtree.y([y]).....	275
6.5.3	d3.geo.conicConformal()	259			
6.5.4	conicConformal.parallels([parallels])	259			

7.2 凸包	275	第 8 章 行为 (Behaviors)	285
7.2.1 d3.geom.hull().....	276	8.1 拖曳	285
7.2.2 hull(vertices).....	276	8.1.1 d3.behavior.drag()	286
7.2.3 hull.x([x]).....	276	8.1.2 drag.on(type[, listener])	286
7.2.4 hull.y([y]).....	277	8.1.3 drag.origin([origin])	287
7.3 多边形	277	8.2 缩放	287
7.3.1 d3.geom.polygon(vertices).....	278	8.2.1 d3.behavior.zoom().....	288
7.3.2 polygon.area().....	278	8.2.2 zoom(selection)	288
7.3.3 polygon.centroid().....	278	8.2.3 zoom.translate([translate]).....	289
7.3.4 polygon.clip(subject)	278	8.2.4 zoom.scale([scale]).....	289
7.4 泰森多边形	279	8.2.5 zoom.scaleExtent([extent]).....	290
7.4.1 d3.geom.voronoi ()	280	8.2.6 zoom.center([center])	290
7.4.2 voronoi(data).....	280	8.2.7 zoom.size([size])	290
7.4.3 voronoi.x([x]).....	280	8.2.8 zoom.x([x]).....	290
7.4.4 voronoi.y([y]).....	281	8.2.9 zoom.y([y]).....	290
7.4.5 voronoi.clipExtent([extent]).....	282	8.2.10 zoom.on(type, listener)	291
7.4.6 voronoi.links(data)	283	本书参考资料	293
7.4.7 voronoi.triangles(data)	284		

第 1 章

核心 (Core)

本章介绍 D3 的核心 API 函数¹。D3 的核心 API 主要包含一些常用的文档操作和数据处理等方面的函数。在文档操作方面，D3 的核心函数可以选取 DOM 元素，改变元素的属性、样式和内容。还可以追加、插入和删除元素。通过数据绑定可以创建数据驱动的文档。对元素使用过渡函数可以编写生动的动画效果。另一方面，很多针对数据的函数便于获取外部数据，对数据进行格式化、本地化等处理。有一些函数可以用来生成随机数，操作二维仿射变换。还有大量函数便于操作数组、映射、集合、嵌套等不同结构的数据。

1.1 选择

一个选择就是从当前文档中选中的一组元素。D3 使用 CSS3²选择器来选择页面元素。如果浏览器不支持 CSS3 选择器，则可以在 D3 前引入 Sizzle³来向后兼容。

1.1.1 d3.select(selector)

选择第一个与指定字符串 selector 相匹配的元素，选择器字符串通常是 CSS 选择器。如果没有匹配的元素，则返回一个空的选择对象。

例如，对如下结构的两个表格：

```
<table>
```

1 <https://github.com/mbostock/d3/wiki/Core>.

2 <http://www.w3.org/TR/css3-selectors/>.

3 <http://sizzlejs.com/>.