目录

[后续要补充的一些知识： 1](#_Toc69491956)

[第一章 1](#_Toc69491957)

[为什么要做交互式的图像？ 2](#_Toc69491958)

[为什么可视化要在web上 2](#_Toc69491959)

[第二章 Introduction D3 3](#_Toc69491960)

[第三章： 5](#_Toc69491961)

[SVG基础 9](#_Toc69491962)

[第四章 D3的安装 11](#_Toc69491963)

[第五章 数据 13](#_Toc69491964)

[第六章 Drawing Data 16](#_Toc69491965)

[第七章：Scales 17](#_Toc69491966)

[scaleTime 19](#_Toc69491967)

[把时间转化为像素个数的scale 19](#_Toc69491968)

[第八章 Axes 19](#_Toc69491969)

[第九章 Updates,Transitions, and Motion 23](#_Toc69491970)

[第十章 交互性 30](#_Toc69491971)

[第十一章 使用路径 31](#_Toc69491972)

[第十二章 selection的理解和最佳实践 34](#_Toc69491973)

[第十三章 Layouts 36](#_Toc69491974)

[第十四章 Geomapping 39](#_Toc69491975)

# 后续要补充的一些知识：

1. 后续有时间结合抓包工具再学习一遍计算机网络
2. Css还是完整的过一遍( [CSS-Tricks (css-tricks.com)](https://css-tricks.com/)
3. DevTools看一下

# 第一章

介绍下什么是数据可视化，为什么需要数据可视化以及本书则么使用。

## 为什么要做交互式的图像？

1.让你自己去探索数据 比静态的展示能发现更多的东西

2．交互式可视化提供了数据的概述以及“深入”细节的工具，可以同时完成许多任务，解决不同受众的不同问题。 无论是 对数据非常熟悉的可视化专家还是对数据集不懂的人。

3.会动的更有意思

### 为什么可视化要在web上

1. 什么设备都有浏览器，让尽可能多的人能够上手这个可视化

2.

#### 这本书的主题：

data visualization interactive design web development

D3是一个强大的基于网络的自定义可视化工具

##### 这本书需要的前置技能：

使用的d3的前置技能：

HTML, CSS, JavaScript, and SVG

这本书会在第三章介绍d3所需要的如上的基础知识

###### 这本书不能做的事情

1.问题的解决方法只给作者觉得最简单的（可能还有其他的）

2. 主要介绍基本概念和方法，具体的例子去D3的官网去做结合例子去理姐

使用实例代码

[Releases · scotthmurray/d3-book (github.com)](https://github.com/scotthmurray/d3-book/releases)

# 第二章 Introduction D3

什么是D3？

生成和修改带有数据的文档

**内在过程：**

1. 吧数据加载到浏览器的内存
2. 将数据绑定到文档中的元素，根据需要创建新元素
3. Transforming those elements by interpreting each element’s bound datum and

setting its visual properties accordingly

1. Transitioning elements between states in response to user input（interactive

D3不能做的是：

1. 没有预设图表（又是个绘图工具，干）
2. 不支持旧的浏览器
3. 主要搞SVG图片和JSON数据

最基础的D3不能处理位图（有个插件可以d3-tile plug-in）

bitmap map tiles, such as those provided by

Google Maps or Mapbox？

1. 需要被可视化的数据必须被发送到客户端（要求使用可以共享的数据）

替代方案：安装一些插件 比如 Flash 或者展示静态图（服务器端渲染）

起源和背景：

在2005年世界上第一个可视化软件是用java写的prefuse,它作为浏览器的插件使用，能让非专业编程人员进行可视化

在2007年第二个可视化软件Flare诞生，同样依赖浏览器插件，是用Action‐

Script 写的。可以在Flash播放器中播放。

在2009年第三个可视化软件Protovis诞生，这是一个基于JavaScript的可视化工具包，完全依赖于本地浏览器技术。完全不需要编程，但是带来了调试的麻烦？？

在2011年的时候，D3诞生，直接在网络文档上操作，更加容易调试，实验，缺点是难学的一批。

论文推荐: D3创始人 在 InfoVis 上的paper

其他可替代的选择：

如果只是需要快速生成简单的图表可以用以下的工具：

Echarts, antv,DataV

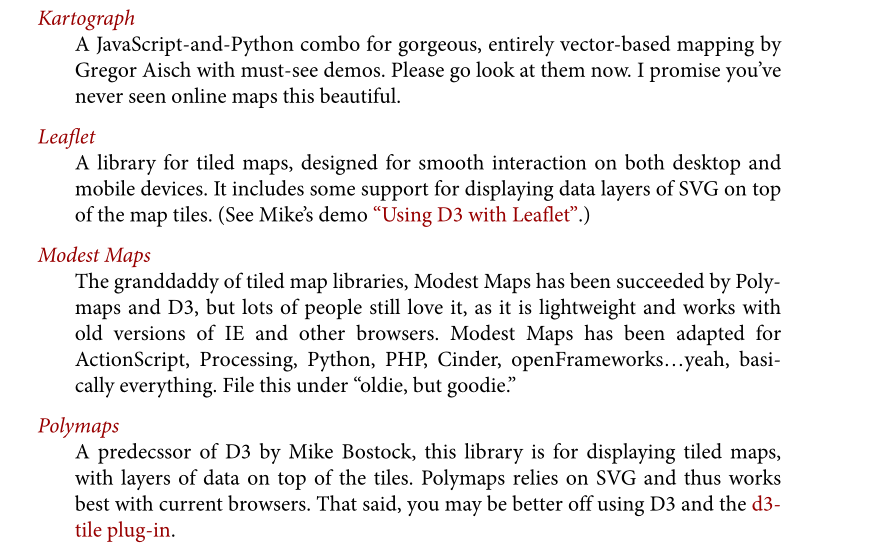
Graph Visualizations

Arbor.js

Cytoscape.js

Sigma.js

Geomapping



Almost from Scratch

P5.js

Paper.js

Raphaël

Snap.svg

Two.js

Three-Dimensional

D3 is not the best at 3D, simply because web browsers are historically two-dimensional beasts. But with increased support for WebGL, there are now more opportunities for 3D web experiences

PhiloGL

Three.js

查看D3的支持库（10-16）

# 第三章：

D3的预备知识（只用看不会的和不熟悉的

Comments：注释

D3有 css风格的选择器

Css:

[CSS reference - CSS: Cascading Style Sheets | MDN (mozilla.org)](https://developer.mozilla.org/en-US/docs/Web/CSS/Reference)

JSON：

1. JSON官网其实解释的很清楚, JSON采用完全独立于语言的文本格式, 因为易读, 易写, 易解析的特性成为理想的数据交换语言  
2. 要搞清楚个这个问题, 就要明白JSON可以有哪些值, 主要有三种类型的值:  
简单值(字符串, 数字, 布尔, null), 对象, 数组  
所以, "null"是合法的JSON值, "1"也是合法的JSON值, 要测试也很简单, JSON.parse("null")和JSON.parse("1")都可以正确返回结果  
3. JSON和JS对象应该没有什么比较性而言吧, 一个是文本格式, 一个是对象, 问题应该是JSON中的对象和JS对象的区别吧, 我姑且就按JSON中的对象和JS对象的区别回答  
主要区别是,  
1. JSON中的对象中属性名必须使用双引号  
2. 属性值不能除了简单值, 对象, 数组以外的值

GeoJSON（针对地理数据优化的JSON



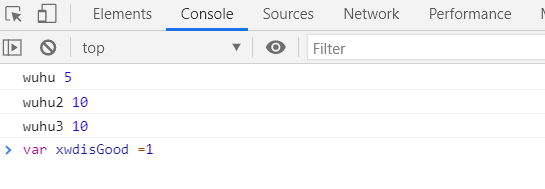
Note: coordinates 前面是经度后面是纬度

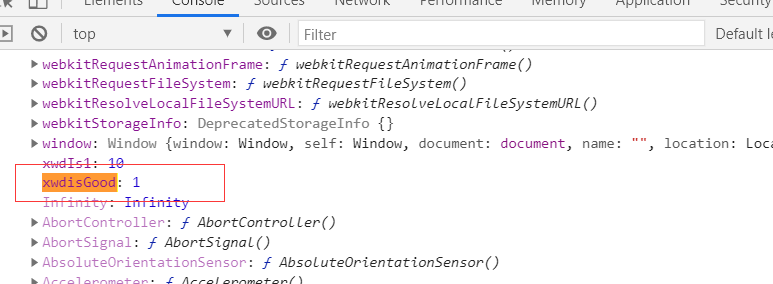
Javascript 弱类型语言，用typeof 检查类型

直接声明的

**全局命名空间问题**

在js文件中定义的全局变量默认会添加到window对象中。



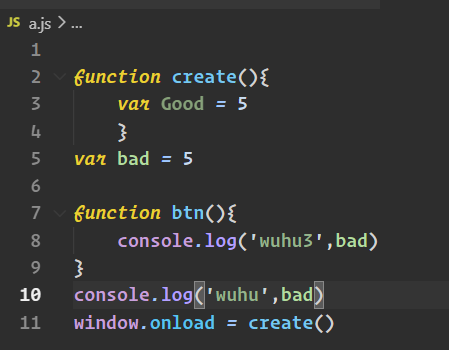


这样会带来一个问题，如果我有的html文件引入了多个js对象，它们有同名的全局变量。就会出现多个文件引用同一个window下的变量（污染命名空间）

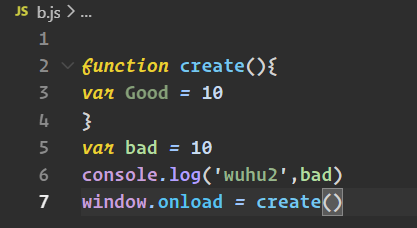
比如我引入a.js,b.js



在a.js中定义bad=5



在 b.js中定义 bad = 10

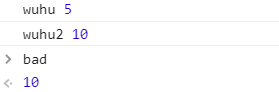


打开html页面，看下window对象下bad这个变量的值是多少

可以看到首先控制台输出了我们在js中定义的log输出

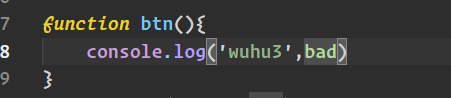


表明 js文件是按我们放入html的顺序加载的。接下来我们查看下window对象中bad的值是多少。

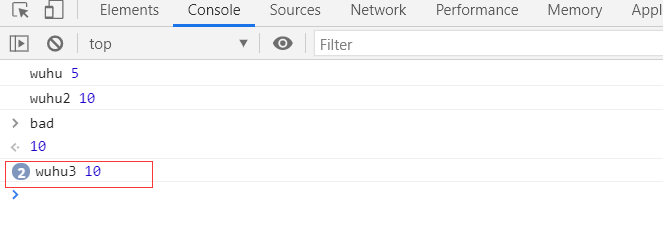


**Note：** 因为js文件中的全局变量对象是window管的所以直接输入变量名就可以了。

1. js中添加一个处理按钮的函数，再运行中点击看下两个文件是否是用的是同一个对象



可以看到确实。



解决方法，给js文件单独定义一个对象作为全局变量的容器。避免命名冲突导致的全局变量的污染

没有用var声明的变量也会污染全局命名空间

## SVG基础

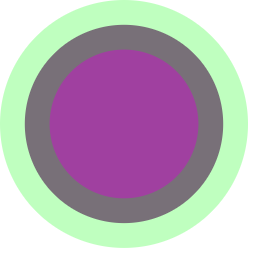
SVG =>text

基线=> 相当于字体的大小+水平的位置

**SVG设置样式的最佳实践**

把样式属性抽出来以css格式去做,(为了区分真正的css和SVG 可以在样式前面加上SVG的标识)

以绘制一个圆为例子来演示如何以css的方式来设置SVG的样式。



原来的面向对象的写法：

<circle cx="25" cy="25" r="20" fill="rgba(128, 0, 128, 0.75)"

stroke="rgba(0, 255, 0, 0.25)" stroke-width="10"/>

css写法

吧需要用到的css样式写到defs里面，方便复用

<defs>

<style type="text/css">

<![CDATA[

svg #cir {

stroke: rgba(0, 255, 0, 0.25);

fill: rgba(128, 0, 128, 0.75);

stroke-width:10

}]]>

</style>

</defs>

<circle cx="25" cy="25" r="20" id="cir"/>

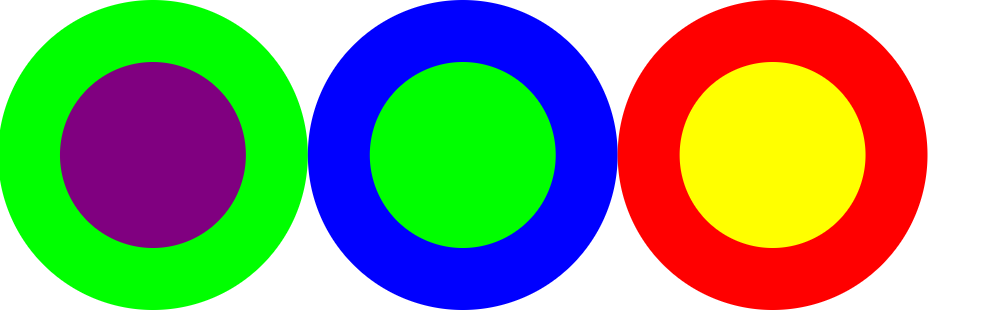
<![CDATA[]]>:这个是引入css标签的外框

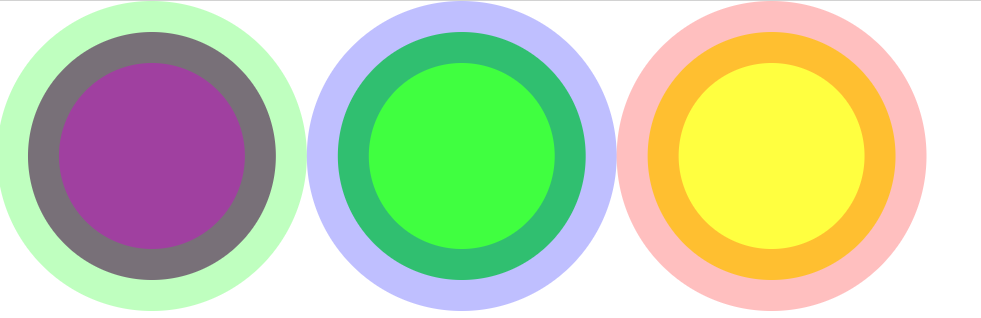
svg #cir：这个是要应用样式的circle元素的id,为了区分真正的css样式和SVG的类css样式最好在前面加一个svg名字来区分下。

SVG图像叠放的顺序和引入的顺序一致

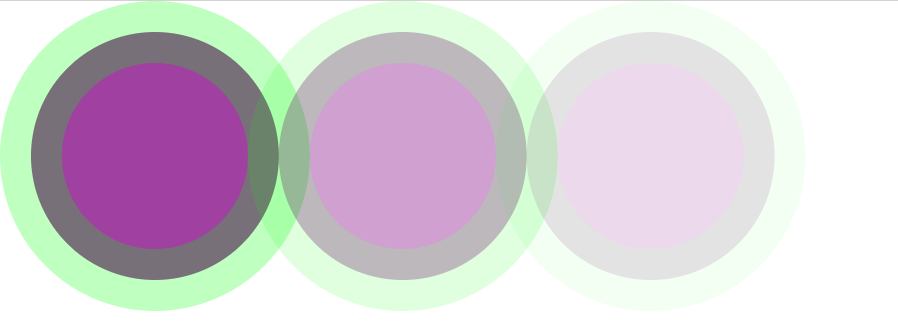
**SVG透明度**：

1. 如果设置两个图片都是具有一定透明性，交界处会是两个颜色的叠加后面的图片压在前面的图片上面。





1. Opacity设置的透明度会和rgba设置的透明度会成倍的叠加



Note:IE8及之前的浏览器不支持SVG

**基础变形:**

translate

你能把元素移动一段距离

rotate

旋转一个元素是相当常见的任务

scale

scale()变形改变了元素的尺寸

# 第四章 D3的安装

1. 去官网下载压缩包<https://d3js.org>

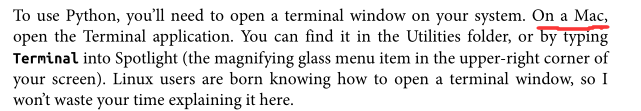
解压开来里面有两个js文件

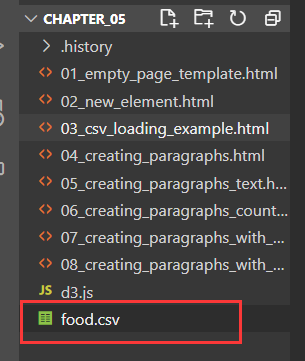


在编写代码的时候用d3.js 方便调试找问题。在实际运行的时候用d3.min.js 节省空间。

1. 在自己的html文件上引用该js文件
2. 运行在服务器上

因为，部分浏览器存在限制从本地加载 js文件和从不信任的网站加载CSV,JSON等。出于这个原因，让我们的页面运行在web server上是更好的选择。首选本地的server(不用钱而且更快)

比如我从本地抓取一个CSV文件



<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>D3: Loading data from a CSV file</title>

<script type="text/javascript" src="d3.js"></script>

</head>

<body>

<script type="text/javascript">

d3.csv("food.csv", function(data) {

console.log(data);

});

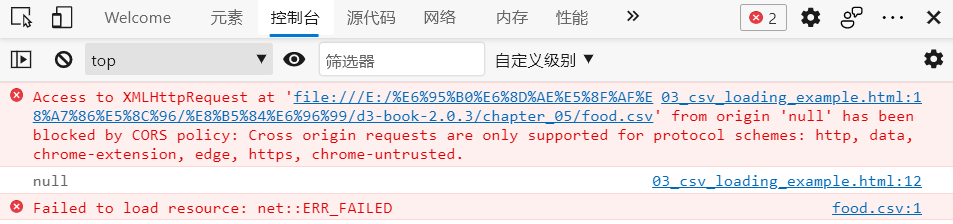
</script>

</body>

</html>

运行查看控制台信息，被跨域政策阻止了。让我们来看下跨域政策都允许那些格式

http,data,chrome-extension,edge,https,chrome-untrusted





???

本地的：

1. Python -m http.server 8888
2. 用Apache+WampServer（可以拖拽式安装包）

[WampServer, la plate-forme de développement Web sous Windows - Apache, MySQL, PHP](https://www.wampserver.com/en/)

[Download XAMPP (apachefriends.org)](https://www.apachefriends.org/zh_cn/download.html)

# 第五章 数据

什么是数据

not just text, but bits and bytes representing images,audio, video, databases, streams, models, archives, and anything else.

D3中指的的数据一般是 json,csv,txt

D3的语法：

链式语法

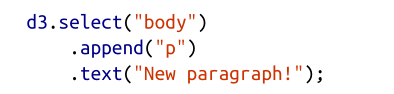


D3:是引用的D3对象

Select:是css选择器，会返回选中的第一个元素，如果想要全部用SelectAll

链式方法核心就是每调用一个方法基本上会返回一个引用对象(有时候不会返回引用对象)

Note:建议一个方法一行不要写到一行里。



D3的API列表：

[d3/API.md at master · d3/d3 (github.com)](https://github.com/d3/d3/blob/master/API.md)

**D3加载数据**

加载CSV数据

D3.csv(“path”,function(data){

Process on data …

})

Note:

1. D3读取csv默认第一行是列名

2.听说会记录列名信息，但是我没看到= =可能是api更新了。

3. csv读入的数据都为string类型

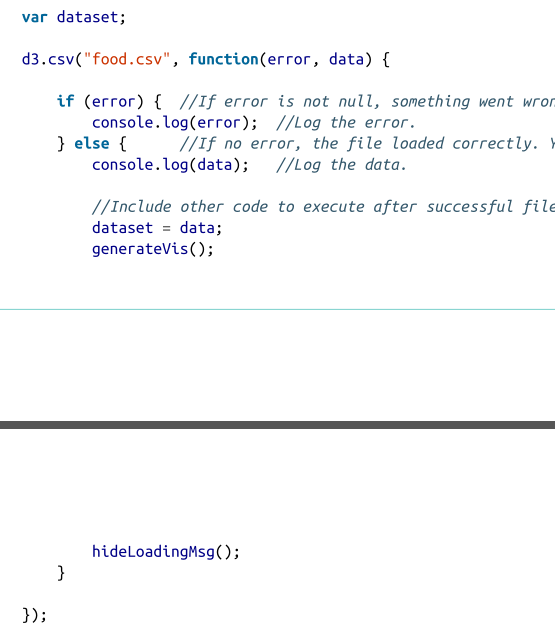
解决方法指定一个行转换函数。

目前新的api和书上的不一样了用dsv，来代替csv函数。

D是分隔符的意思



3. 因为无论成功与否第一个回调函数都会执行，而有时候会因为数据的加载或者网络的问题，导致得不到预期的数据。所以要对错误进行一个捕获。给原来的function前面加个error



一个可以吧excel转化为其他格式的工具[Mr. Data Converter (shancarter.github.io)](https://shancarter.github.io/mr-data-converter/)

d3.select("body").selectAll("p")

.data(dataset)

.enter()

.append('p')

.text("New paragraph!");

Note:

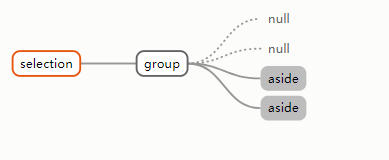
1. 为啥enter().append() 会连之前的p标签也选中，

[How Selections Work (ocks.org)](https://bost.ocks.org/mike/selection/)

因为.append() 会把enter的element和 update element 都选中返回。

什么样的叫update的

原来存在的 <p> 没有\_\_data\_\_属性。

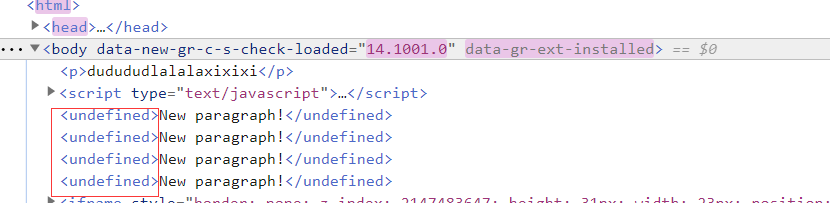


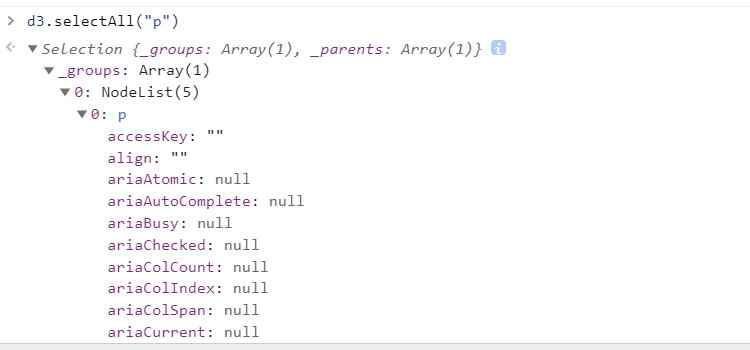
这个null可以保留位置信息

如何将数据和元素绑定呢？

首选选择元素（selectAll）以及加载数据（data）.接着做数据绑定（enter）

如果元素和数据个数不一致enter函数会添加空占位符





没找到绑定的数据

加载进来的数据如何使用？

D3的链式方法可以以函数作为参数。通过传入一个只有一个形参的匿名函数可以获取到绑定到元素上的数据

Text(function(d){

return

})

D3还能加样式

attr() 方法控制DOM属性值

style()方法直接对元素应用css样式

为啥

# 第六章 Drawing Data

**本质：**

1. 将数据绑定到 Dom结点上。

2. 动态改变节点的样式来呈现出画图的效果

Attr 和style的区别?

DOM的属性

Style 其实也是Dom的属性

只不过它很大且很常用所以细分出来了

**D3 api的内在原理**

Style方法的本质是在dom标签上加style属性然后往里面给值（要给完整的字符串）

d3.select("body")

.selectAll("div")

.data(dataset)

.enter()

.append("div")

.attr('class','bar')

.style('height',function(d){

return d+'px'

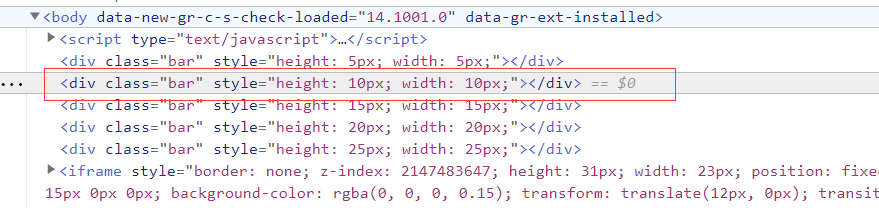
})

.style('width',function(d){

return d+'px'

})

加了两个style 还设置了一个class



Note：这些api的本质还是往它们的标签上加话，如果value无效照样咩用。



给SVG增加class属性该用法是否有问题？

Note:class的属性会覆盖svg本身的值

**散点图**

做数据映射的时候不要吧数据映射到半径上，这样会扭曲关系，请映射到区域面积上。

# 第七章：Scales

Scales are functions that map from an input domain to an output range for visualization purposes.

**Scales的本质**：(LinearScale)



domain:原始域

range:目标域

例子：

输入点 10

输出（10-0）/250\*（1-0）+0=0.04

去做适应屏幕的缩放用的

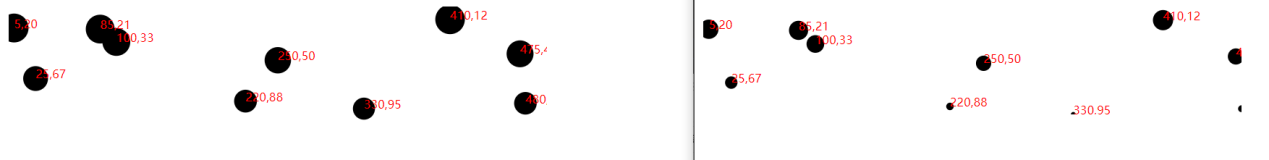
scale和axes是不一样的

var scale = d3.scaleLinear()

.domain([100, 500])

.range([10, 350])

为了可视化需要做的比例尺。



很明显显示效果更好

官方推荐的设置margin的方式：

[Margin Convention / D3 / Observable (observablehq.com)](https://observablehq.com/@d3/margin-convention)

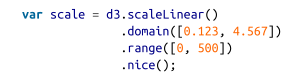
**一些辅助scale的函数**：

nice():给数做四舍五入的 [0.201479…, 0.996679…] =>[0.2,1.0]

rangeRound():只保留整数部分，可以有效防止毛边。

Clamp():吧输出超过range的压缩到边界上

这些函数的用法 ：



其余的scale:

1. scaleSqrt:

2. scalePow

3. scaleLog

4. scaleQuantize:映射到离散值，分类用的

5. scaleQuantile：输入不连续的时候用的

6. scaleOrdinal: 输出为非数值 比如 输入[1,2] 输出[‘apple’,’orange’]

7. schemeCategory10, schemeCategory20, schemeCategory20b, and

schemeCategory20c:预设分类类别，输出各个类别的颜色

8. scaleTIme:???

Note:

1. scale给范围都是传数组的！！！！

2. scaleSqrt的映射还不是很明白

UTC协调世界时

**TimeScale:**

Moment.js

[70 | Rocket Science with Rachel Binx – Data Stories](https://datastori.es/70-rocket-science-with-rachel-binx/" \l "t=30:28)（关于做UTC相关的任务，有空再看）

**D3时间格式输入的说明：**

[d3/d3-time-format: Parse and format times, inspired by strptime and strftime. (github.com)](https://github.com/d3/d3-time-format" \l "locale_format)



var parseTime = d3.timeParse("%m/%d/%y")

**CSV和Excel的区别？**

|  |  |  |  |
| --- | --- | --- | --- |
|  | 格式 | 能做的事情 |  |
| csv | 文本文件 | 数据交互不能记录公式 |  |
| excel | 二进制文件 | 可以搞公式，冻结列之类的操作 |  |

## scaleTime

## 把时间转化为像素个数的scale

D3.scaleTime()

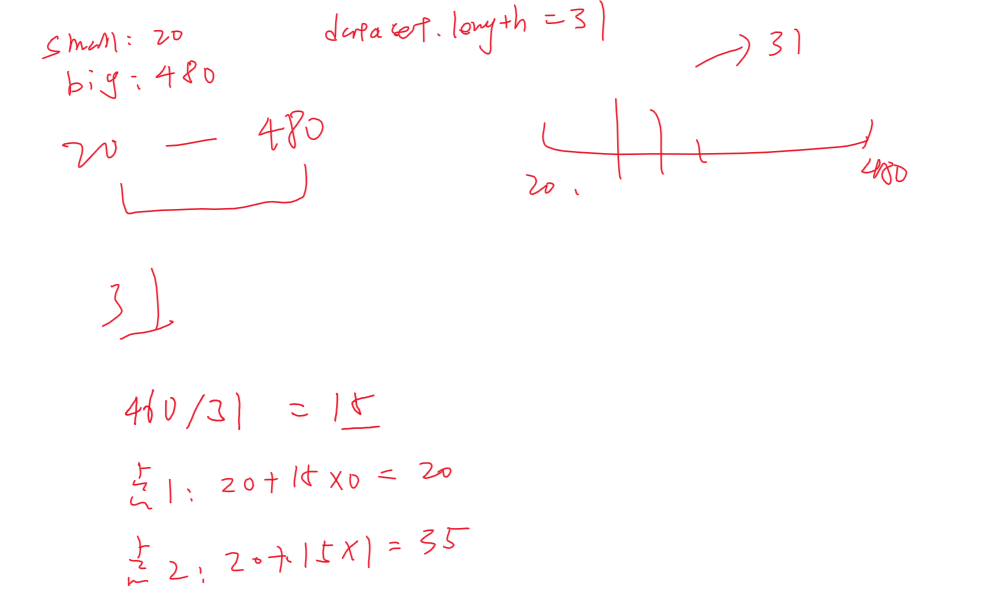
.domain(Min(dataSet),Max(dataSet))

.range(small,big)

domain:传入原始的时间

range:目标的范围

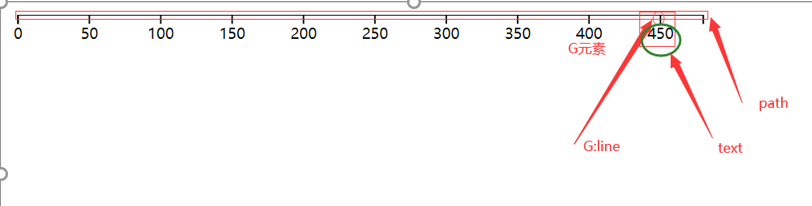
本质相当于给时间排序，然后把range的范围切成dataset.length个对应的值就是时间映射过去的值。

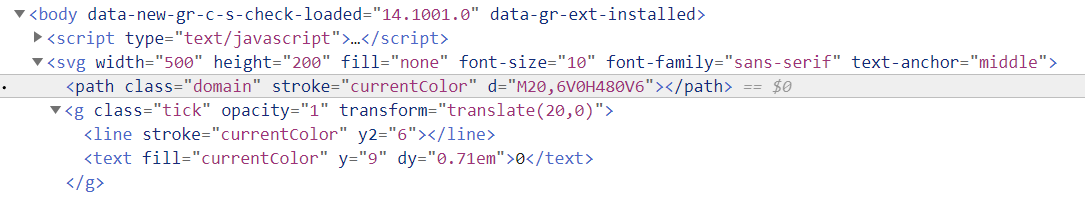


# 第八章 Axes

Axes :

1. 坐标轴
2. 是一个没有返回值的函数（相当于vue中组件），是要画到DOM里头的。





g ：群组元素,管理群组里的元素 比如组织共同行为（div）

**call**

什么是call函数

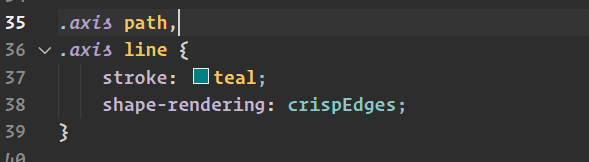
感觉就是调用Axis的组件去渲染

移动坐标轴：

在 g元素上添加变形 移动坐标轴

如何给Axes添加css样式?(这是一个和给vue组件添加样式的类似的问题）

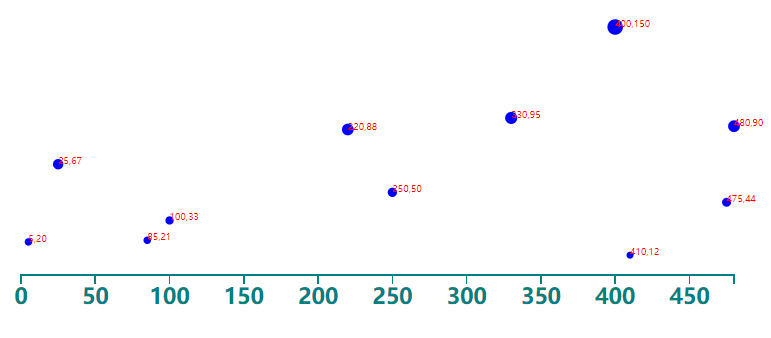
用子孙选择器：选中组件中的基础元素

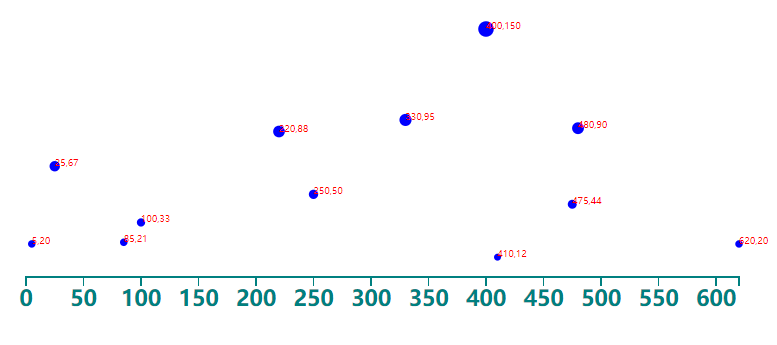


Why?

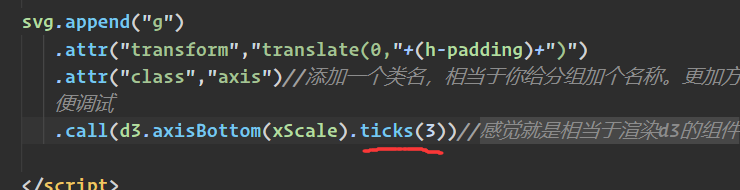
**shape-render :**画抗锯齿的线条

D3的Axes 会随着你的标签自动去改



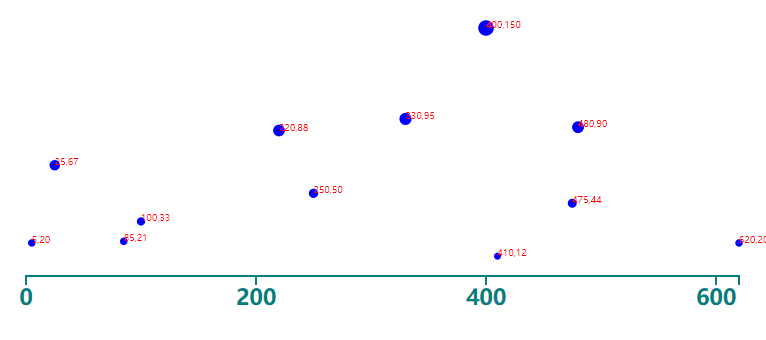


可以设置Axis的tick数

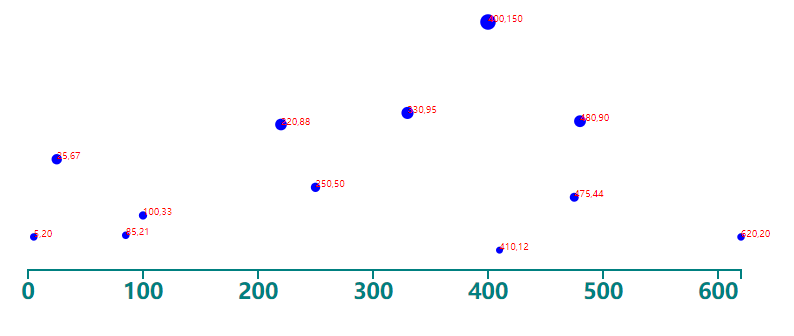


最终显示的ticks 和传入的值不一定相同。D3会使用他认为最清楚可读性最好的方式去划分

传入3=>5个点



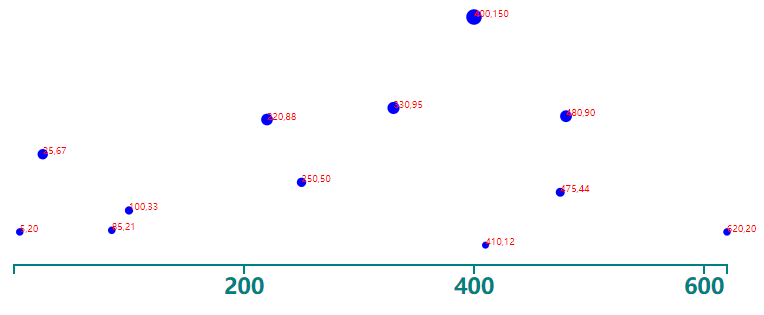
传入8个点=> 8个点



想要和以为这样可以的差距 买家秀 卖家秀

但是他不一定准，想要完全控制咋办？

TickValues([]) 传入完整的ticks数组



Wordwrap

之前因为 vscode识别html页面为

D3 tick format

[d3/d3-format: Format numbers for human consumption. (github.com)](https://github.com/d3/d3-format" \l "locale_format)

# 第九章 Updates,Transitions, and Motion

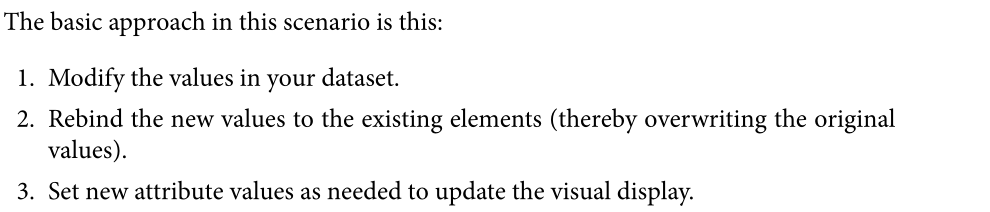
1. 数据值变，总数不变

**Updates**

全部update

部分

具体的看12章就行了



**Transition:**

添加过度，原来点击update是一下子重新画了一页。

**Transition（）**

会插值画出很多中间过渡的图形。比如 现在高度为100 目标高度为200 100

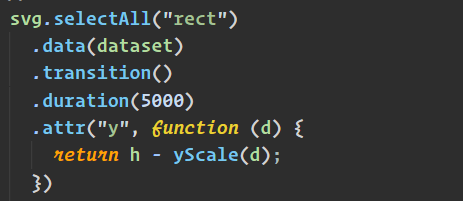
110 .。。190 200 .

要在元素存在的情况下调用，加载想要变化的效果前面

初始值



要变化的：



**duration ( num milliseconds)（指定插多少）**

param : milliseconds

用这个方法指定，默认是250毫秒或者 1000毫秒重绘一次插值后的图像

最佳实践：

1. 次要的 150毫秒(例如悬停在元素上)
2. 主要的1000毫秒（一个视图切换到另一个视图）

**Ease()（自己选过渡特效）**

用Ease 方法可以指定transition() 变化的动画指定transition() 变化的动画方式，加在 transition 后面 attr前面（在duration前后都可）

param fun: 给一个过渡特效函数，根据时间计算位置

**常用的ease方法**

本质是传入一个函数，根据时间动态计算位置

**d3.easeCircleIn**

Gradual ease in and acceleration until elements snap into place.

**d3.easeElasticOut**

The best way to describe this one is “sproingy. ”

**d3.easeBounceOut（这个效果没啥实际意义）**

Like a ball bouncing, then coming to rest.

**d3.easeLinear**

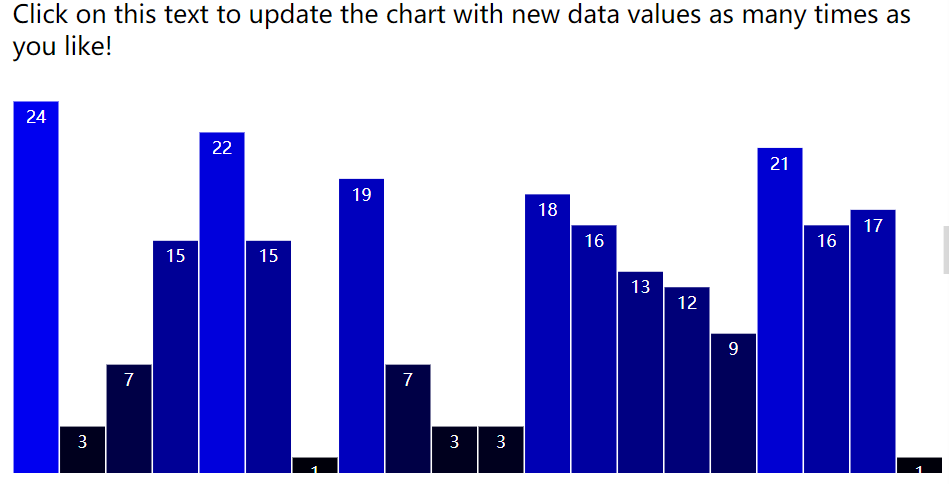
**所有效果**

[d3/d3-ease: Easing functions for smooth animation. (github.com)](https://github.com/d3/d3-ease)

**delay( milliseconds )**

指定过渡开始的时间

Param milliseconds: 以毫秒为单位，多久开始



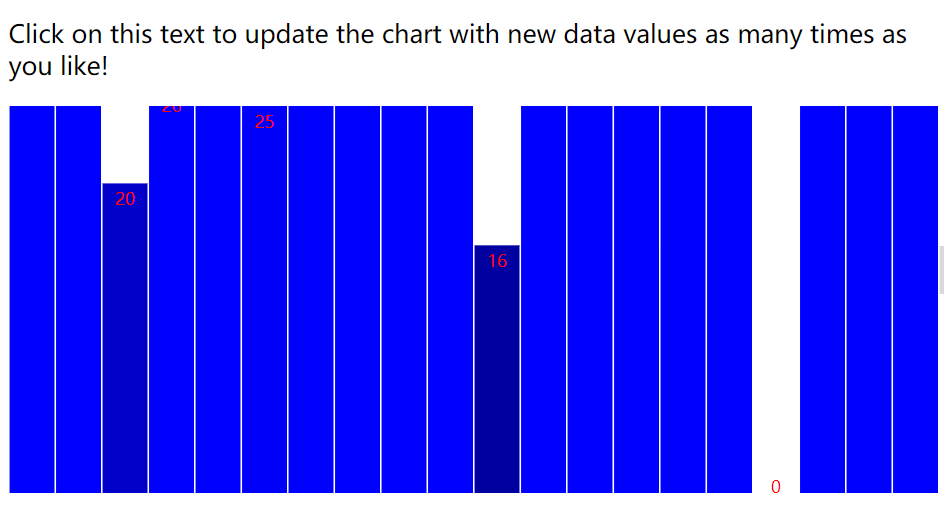
当柱子变小有的标签遮住了。咋么做？

判断，当d<1 把标签上移到柱子上面

**过渡中要注意的问题**

1. **Update scale**

随机生成柱子的图中，如果不改变scale会出现柱子伸出Svg画布的现象。



1. **Update Axes**

Axes 一开始会根据数据点的值来设置自己的刻度。但是，在绘制完之后不会再自动重绘了。要在xscale改变之后再次重绘。

**On(type,function(){})**

设置在Transition 开始的时候或者结束的时候的一些操作

type: 传字符串，“start” or “end”

function(){}:要执行的动作，最好不要在里面加上transition会停掉前面设置的transition。

**Note：**

1. 当有多个transition 作用在一个元素上，会停止之前的选择最新的（就是直接选择最新的） 这是D3故意设计的
2. 最好不要用on来做过渡效果。

url啥意思

[<url> - CSS（层叠样式表） | MDN (mozilla.org)](https://developer.mozilla.org/zh-CN/docs/Web/CSS/url()) 搜不到，感觉是引用一组Svg中的元素。作为属性

**SVG clipPath**

[clipPath - SVG | MDN (mozilla.org)](https://developer.mozilla.org/zh-CN/docs/Web/SVG/Element/clipPath)

定义范围和裁剪的形状，然后给其他元素用

剪切路径限制了图形的可见范围。从概念上来说，如果图形超出了当前剪切路径所包围的区域，那么超出部分将不会绘制。

**Note**:1.注意属性名字和标签名字不同！！！

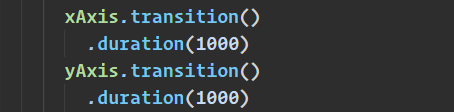


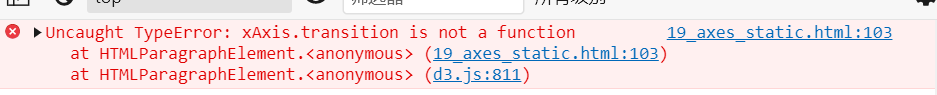
**问题：**

1. 试下自己加个基础变形的效果

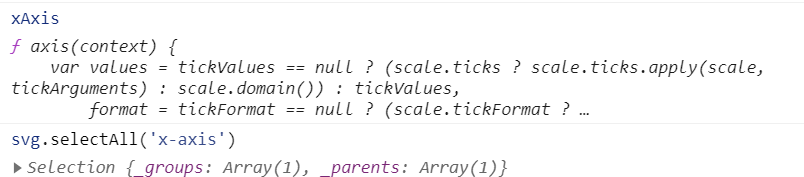
Transform=》translate, rotate, scale

2．异想天开直接用报错了





估计是调用对象不对，



Transition 是Selection的方法而xAxis甚至连返回值都无

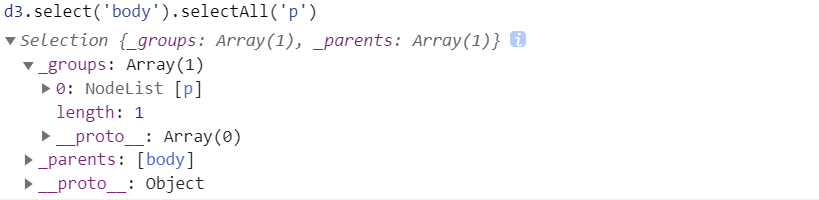
Scale能直接改是因为 scale 是映射函数，而Axis本质是Dom节点，要用操作Dom的方式去改变

为什么 Axes 会跟着 circle的值变？

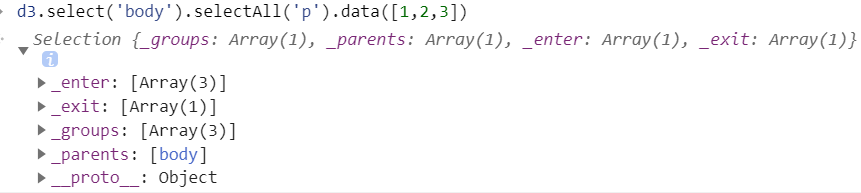
深入理解：D3 select

NodeList是Dom节点的集合不是数组

[NodeList - Web API 接口参考 | MDN (mozilla.org)](https://developer.mozilla.org/zh-CN/docs/Web/API/NodeList)

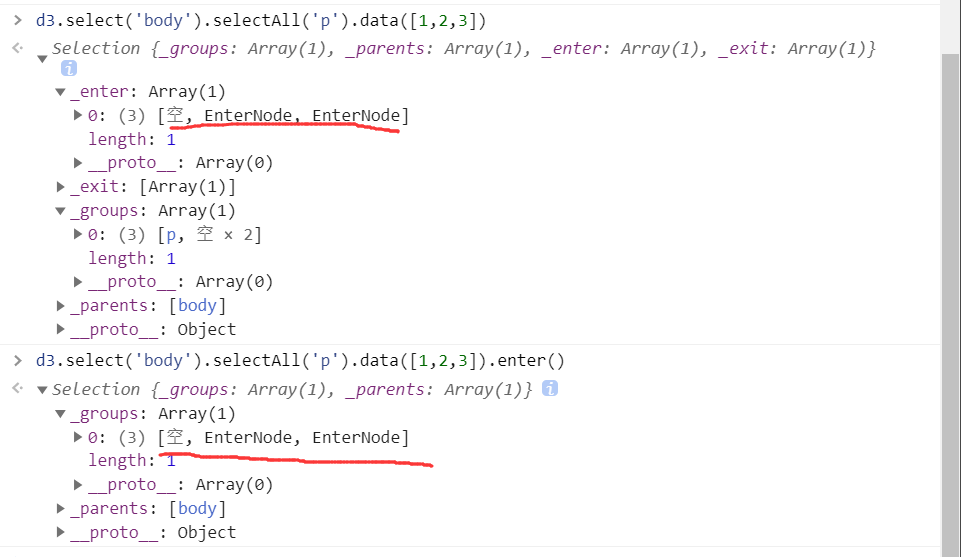


.data() 返回的是Update Selection，是一个维护随着数据变化的一个状态数组，比如在这个例子中显示，如果后面调enter就是再添加两个节点，如果exit 就是删除目前有的那个节点。

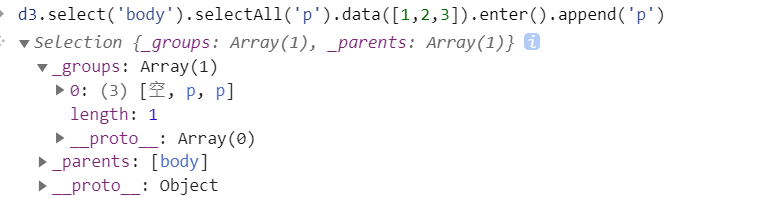


.enter() /.exit() 返回的是 update 之后的状态

比如选择 enter 把 \_enter 数组中的元素放入 \_groups中

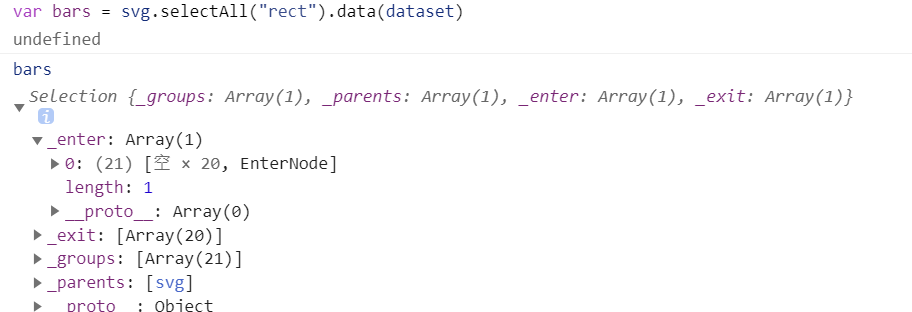


点击 append 把append的元素换掉enterNode

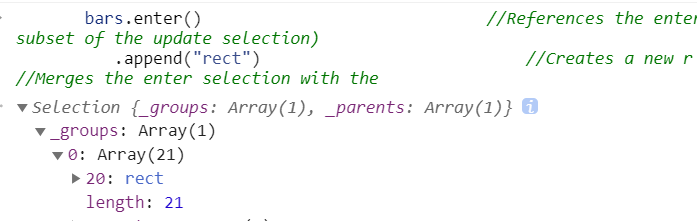


理解添加一个元素到数组

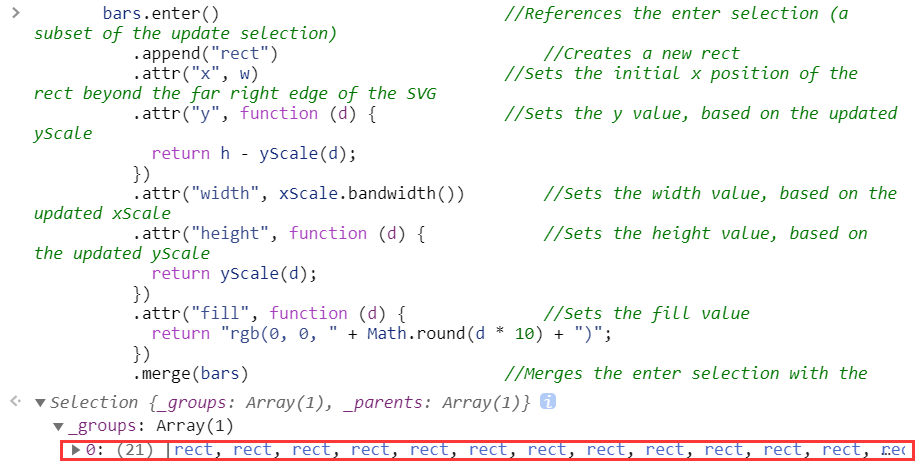
添加一个新元素到dataSet,.data 会添加一个新的元素进去。



.enter().append（） 后返回的是新增加的内个节点的选择器引用，长度是21，但是返回的实际元素只有第20个rect 相当于 Array[length-1] 这个元素相当于属于Array(21)的元素



到这里标明新的元素已经加进来了，但是旧的元素也得改变位置。我们需要对所有的元素进行操作。用merge(bars) 把原来选中前20个元素 +现在选中第21个元素 并起来，



这个解释好像有点问题，之前一共20个创建19个也行

**删除元素特效：**

1. 更新dataset
2. 重新绑定data到DOM上
3. .enter 选中还会存在的所有元素，改变这些元素的位置
4. .exit()选中要退出的元素，设置他退出的效果.transition().duration(1000).remove()

remove() 删除被选中的DOM元素。

**按key更新数据：**

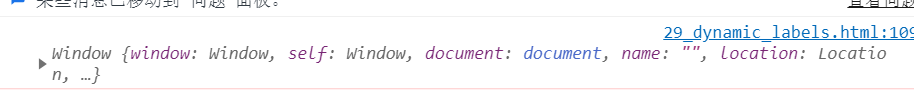
原来rebind 数据是比较dataset的长度，直接换掉的。现在按key更新是更新dataset中的某些元素。 相当于之前是给DOM节点重新赋值\_\_data属性，现在是按key找，更新某个dom节点而不是直接删。

**综合例子：添加和删除**

Function ()}{} 匿名函数的this是什么？

匿名函数最经典的总结：就是谁调用匿名函数，匿名函数中的this就指向谁；匿名函数是有执行上下文，只是执行上下文是执行的时候传递过来，箭头函数中的执行上下文是父级的执行上下文， ... this是JavaScript中的一个关键字，在函数运行期间自动生成的对象，只可以在函数内部使用，表示函数在运行时候的环境对象。

箭头函数的返回值：



匿名函数的this的返回值



研究下箭头函数和匿名函数的区别

明天起来做一个第九章综合的例子： 用keyupdate 的增加删除（后面想做个交互式的排序动画）

前十章是基础

# 第十章 交互性

[Event compatibility tables (quirksmode.org)](https://www.quirksmode.org/dom/events/)

D3支持的事件都是js的标准事件（自定义的不可以

监听事件一般是绑定所有元素上的

Css3新样式能做更多的事情

同样一个事情用css实现好还是用js实现好，比如鼠标悬停?

This 选中的是谁？

the current element upon which we are acting

在重叠元素上的指针事件问题:

指针事件只会被处于图片最上层的元素所捕获，而有的元素不响应指针事件，导致，这个事件失效。

事件冒泡回头再看下

**pointer-events: none;**

除了指示该元素不是鼠标事件的目标之外，值none表示鼠标事件“穿透”该元素并且指定该元素“下面”的任何东西。

用g 分组包括 一组 rect 和text 然后将监听事件放在g元素上，就不用担心 overlap element.

当点击和mouseover同时出现，会出现transition 冲突两个办法

1. Mouseover + Mouseout用css的hover实现，
2. 给其中一个transition起名字，使得两次事件调用的不是一个transition

Note：起了名字之后可以给transition 调用interrupt手动停止事件。

**工具提示：**

创建与销毁

显示与隐藏

代替标签展示数据

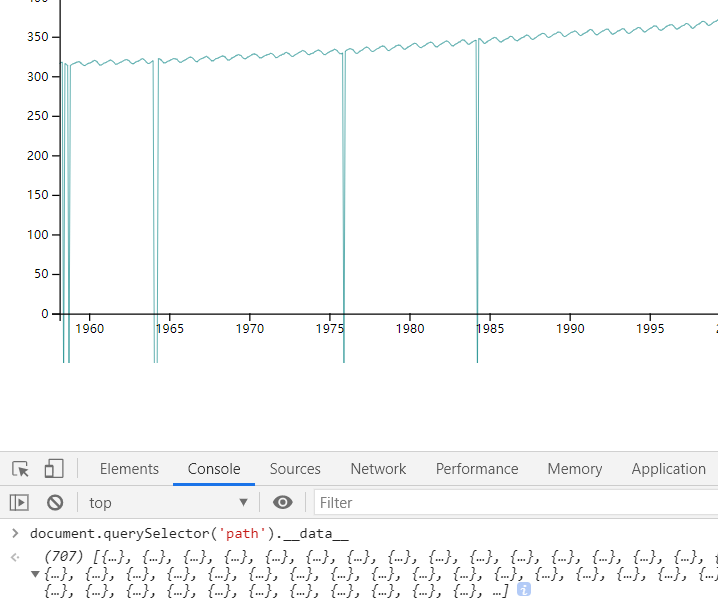
1. 浏览器工具提示（完全无法修改效果）
2. SVG
3. 用js +css 这一套去实现（最灵活也最复杂）

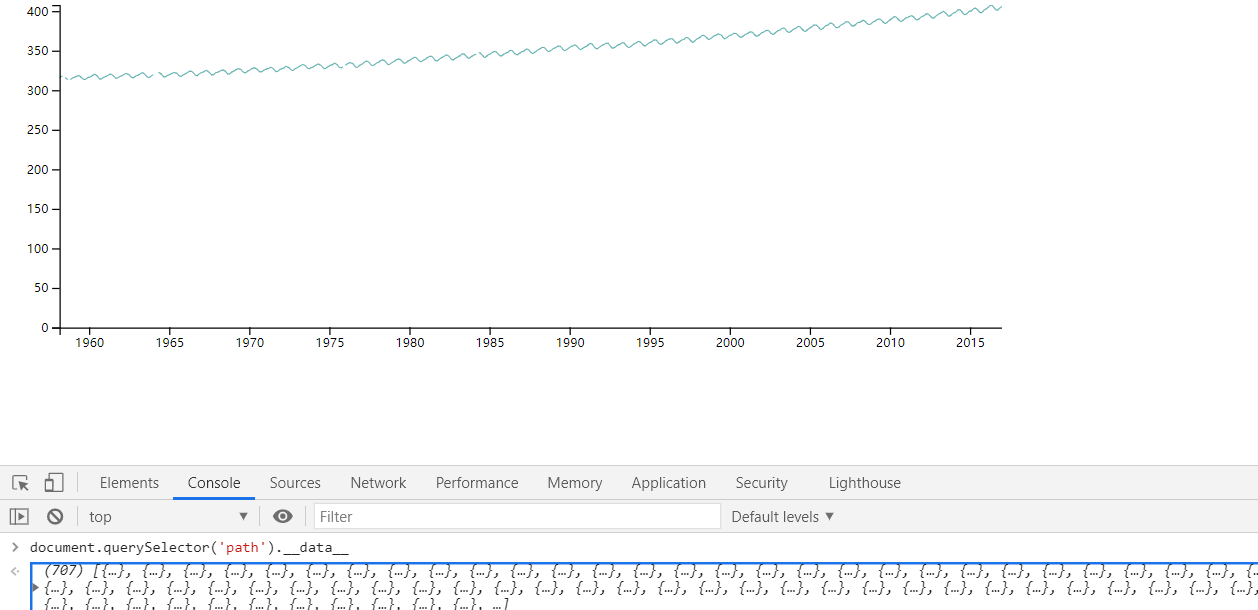
# 第十一章 使用路径

折线图和Area 图都是借助 svg中的path来实现效果的

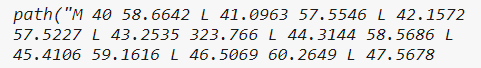
通常步骤：

1. 设置一个生成函数（生成path的d命令）
2. 将数据和样式以及生成函数绑定到path元素上
3. 如果出现缺值 图会断掉一块





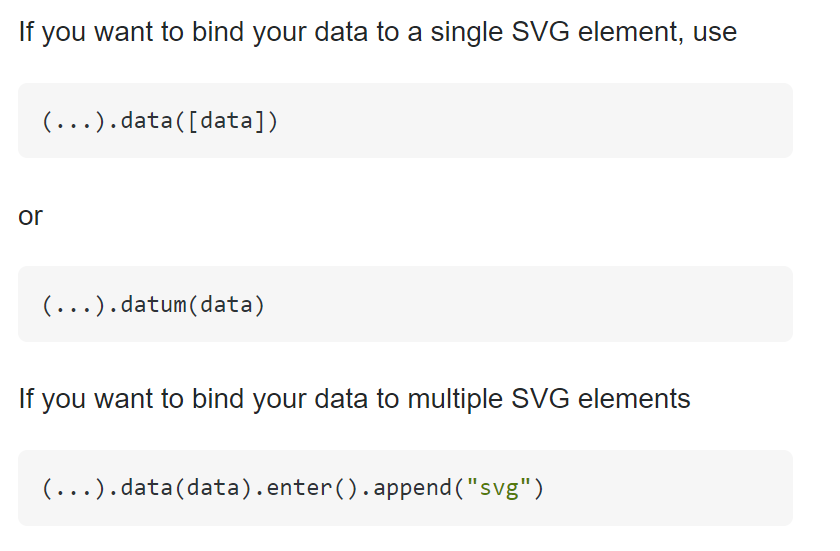




Define去掉了不行的点并且，下次就是把画笔拿起来重新画了所以会出现断的线。

转换格式到csv的工具

Data() 和 datum的区别？



Console.table可以显示表格

A graphical mark = A SVG element

Deal missing data:

**defined(function(){})**

过滤出要渲染的数据

**Param function(){}:** 传过滤规则

**Note：**

1. **被过滤过得数据画图上可能会出现断的线，因为 他跳过一个数据会重新用M 而不是LineTo。而实际绑定到dom上的元素不会受define影响，define只影响d 命令**

用 line.define()

生成path命令的时候跳过缺失数据

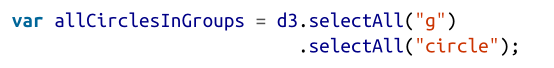
当时实际上被绑定上去的data是没变的

折线如何做出一点点绘出来的动画效果

# 第十二章 selection的理解和最佳实践

Selection 是d3创建和维护的一个js对象。

连续选择的简写





要注意的问题什么时候切换selection了

注意书写



新增两格缩进，修改四格

D3 selection的相关说明（[selection](https://github.com/d3/d3-selection/blob/master/README.md)

Selections are very flexible; I recommend you make and store them in whatever

sequence makes the most sense to you, for your project, coding style, and way of

working

enter() 方法是获取了绑定data之后的enter选择器（exit同理

merge后的返回值叫 update selection

基于数据过滤选择器

**filiter(function(d){**

**process on…..**

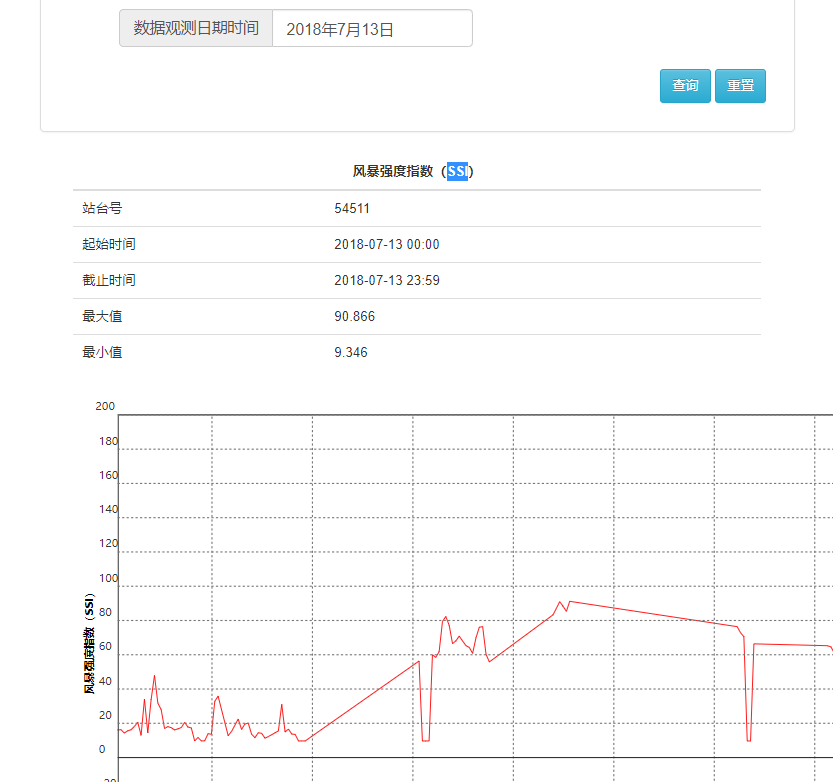
**})**

基于数据过滤选择器

**Param function(){}**:过滤方法

**Note：**

**要问的问题：**

1. 问下老师，之前说用ES6的形式则么实现的
2. 添加柱形，点快了scale会计算错误。
3. D3 没有很好的语法补全和提示
4. 变量英语命名缩写都咋么写的（搜一些一些大厂的命名规范来看）
5. 

SVG画的图能自适应嘛？

Webpack 可以加入本地文件



这个name= “filterPrest”是 选中其他的清除当前用的

d3*.*schemeCategory10

d3预设的一组颜色

  -webkit-appearance: slider-vertical;

改变外观用的

**filter(function(){})**

按条件筛选相应的selection

**Param function(){}**:要执行的动作。

**Note：**

1. 主要对某类型的选择器进行操作用的

**遇到的问题**

**each(function(){})**

each() to run an arbitrary function once for each element in

a selection.

**Param function(){}**:要执行的动作。

**Note：**

1. 主要是便于维护

**遇到的问题**

# 第十三章 Layouts

与顾名思义相反，D3布局实际上并没有在屏幕上为您展示任何东西。布局方法没有直接的视觉输出。相反，D3布局采用您提供的数据，并对其进行重新映射或转换，从而生成更便于特定可视化任务的新数据。还是要靠你来获取新的数据，并从中产生视觉效果。

其他的布局

[Gallery · d3/d3 Wiki (github.com)](https://github.com/d3/d3/wiki/Gallery)

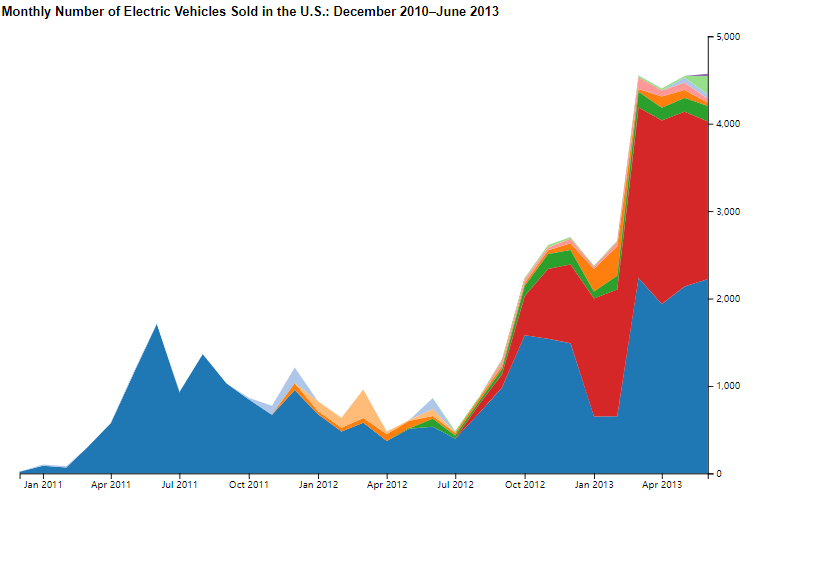
**Firfox 开发版 html svg xml 错误也会检查 nice.**

**Stack Layout**

d3.stack()将二维数据转换为“堆叠”数据；它为每个数据计算一个基线值，这样你就可以把数据一层一层地“堆叠”起来。这可以用来生成堆积条形图、堆积面积图，甚至是流向图(它们只是堆积面积图，但没有严格的起始基线值零)

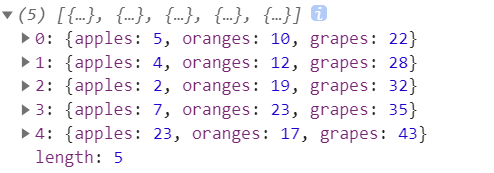
**特点**：Stack图的好处是既能看到整体的变化又能看到 局部相对的比较

但是其实局部的比较是非常不直观的（因为没有一个统一的基线）最好是做一个交互在点击的时候突出局部的比较把它放在最下面。

****

**Stack layout 的过程**

1. **给一个对象数组**

****

1. **设置stack 方法**

**const stack = d3.stack()**

**.keys(["apples", "bananas", "cherries", "dates"])**

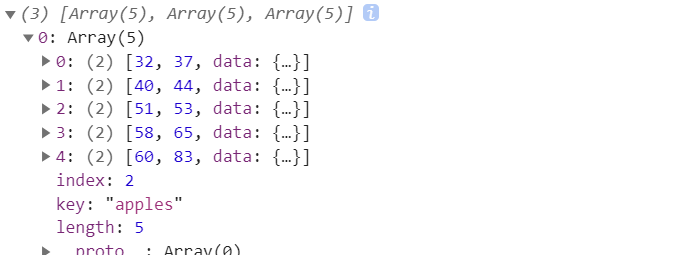
**.order(d3.stackOrderNone)**

**Keys:设置要分类的属性名，顺序也是按keys里的顺序来的**

**Order:指定**

1. **将 原始dataset传入stack**

**Stack(dataset),结构如下：**

****

**外面的Array表示有三个分类，**

**里头有三个数组**

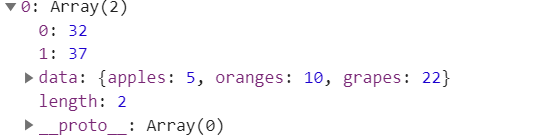
**每个数组里是一个类数组对象**

**Index:是keys中该分类的位置**

**Key:是对应的key的名字**

**Length:该key下的长度**

**类数组对象里面每一个项**

****

**[baseLine,topLine]**

**Data:原始的对象**

**Length:**

1. **将生成的stack数据绑定到目标的元素上去**

**这个柱状图还是有点懵回头再试下**

**SVG:**

**Path**

**Dasharray**

**Dashoffset**

**这俩画虚线的，Dasharray是指定线长和间隔**

**Dashoffset 指定线的偏移 正是向左偏移 负是向右偏移。**

# 第十四章 Geomapping

GEoJSON的结构

GeoJSON 的结构是什么？ 给出一般性的描述

GeoJSON

投影干嘛用的？

Well, as an astute observer, you have also noticed that the globe

is round, not flat. Round things are three-dimensional, and don’t take well to being

represented on two-dimensional surfaces. A projection is an algorithm of compro‐

mise; it is the method by which 3D space is “projected” onto a 2D plane.

lon/lat becomes x/y.

关于D3投影的官方说明文档：

<https://github.com/d3/d3-geo-projection>

演示各个投影效果的一个可视化demo：

<https://bl.ocks.org/mbostock/3711652>

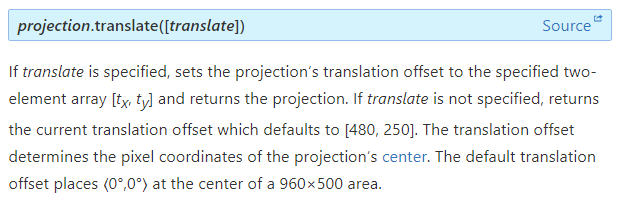
等值线图的缺点是什么？

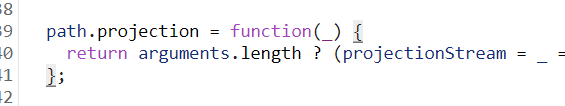
将要观测的数据和地理信息结合起来有几种方法？

什么是 projection.translate?

移动 projection中的 中心点

设定translation offset





Projection 是 path对象的一个属性，传的是引用改变 projection 不用重新在path里设置。

用.html()输入html元素

 north*.*append('text')

*.*attr("x", w */* 2)

*.*attr("y", 15)

*.*text("&uarr;")

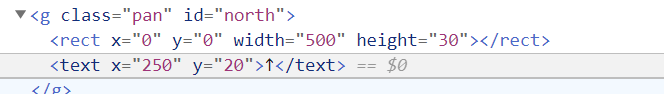


    north*.*append("text")

*.*attr("x", w*/*2)

*.*attr("y", 20)

*.*html("&uarr;");



如何阻止text的浏览器默认行为？

用 js 和css都可以

[(13条消息) 取消双击等选中文字事件\_u011200562的博客-CSDN博客](https://blog.csdn.net/u011200562/article/details/105659057)

[(13条消息) js禁止双击选中文本\_qq\_42256562的博客-CSDN博客](https://blog.csdn.net/qq_42256562/article/details/103635887)

在d3里用js 尝试还没成功过。

D3拖拽

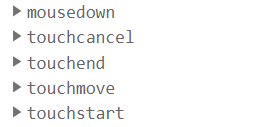
**一般步骤：**

1. Group all “pan-able” elements into a single SVG g group, for simplicity.

2. Bind a drag behavior onto that group.

3. Tell D3 what to do when dragging occurs.

**实际实现的几个事件：**



Touch相关的是触摸屏用的

Drag的一般用法

//Define what to do when dragging

var dragging = function (d) {

//Log out d3.event, so you can see all the goodies inside

//console.log(d3.event);

//Get the current (pre-dragging) translation offset

var offset = projection.translate();

//Augment the offset, following the mouse movement

offset[0] += d3.event.dx;

offset[1] += d3.event.dy;

console.log(d3.event)

//Update projection with new offset

projection.translate(offset);

//Update all paths and circles

svg.selectAll("path")

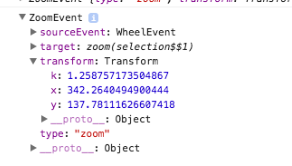
.attr("d", path);

}

d3.event:返回当前发生的事件

## Zoom

是什么?



sourceEvent: 描述触发ZoomEvent的事件 wheelEvent是滚轮

transform:

k: 比例因子

x: 做变换的x坐标

y: 做变换的y坐标

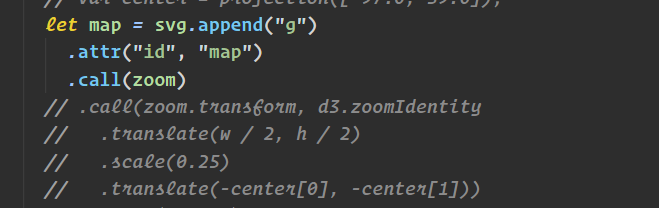
This approach to updating the map view is called reprojection. In this way, we “move”

the map by changing the parameters of the projection, and then “reprojecting” all the

map elements.

为什么zoomEvent 要把变换值绑定到目标dom元素上？

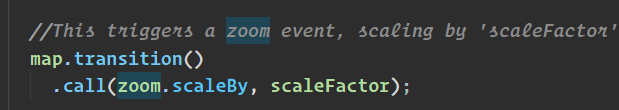
Selection.node（） 会返回dom结点



为了让经纬度和坐标能对应上

这个call有什么用？

加一个变换，这个变换会把 zoom.transform的数据绑定到 g上。



加了缩放之后手动pan的按钮坏了，因为zoom

ZOOM和drag区别：

1. Zoom除了能drag还可以缩放
2. Zoom是直接设置tranform 而且 会把tranform直接绑定到dom元素上（好处是不同的元素可以不同程度的缩放），drag是给偏移。

给zoom加上限制;

scaleExtent([min\_scaleFactor, max\_scaleFactor])

translateExtent():[[x0,y0],[x1,y1]]

Note:1. 这些限制会被变换取代。

preset views：

预先设置的视图

Path.centroid() 返回路径的形心坐标

可以用来设置label在路径中心，但是在形状不规则的情况下效果不是很好。

一个寻找多边形中心的算法:

[A new algorithm for finding a visual center of a polygon | by Mapbox | maps for developers](https://blog.mapbox.com/a-new-algorithm-for-finding-a-visual-center-of-a-polygon-7c77e6492fbc)

**如何获取和准备原始的地理数据：**

1. **Find shapefiles**

**是什么？**

**Shapefiles 是一个起源于使用GIS软件的地理学家，制图师和科学家。**

**open source QGIS**

**一般的shapefile采用WGS84投影**

**泽么找？**

**<http://www.naturalearthdata.com/>**

[bit.ly](http://bit.ly/2tNyl8h)

只有图的

**<https://openvectormaps.com/>**

1. **Choose a resolution**

**一般单位有两种**

**英寸：**1in=2.54cm

**米**

**用途：**

**Low-detail**

**1:110m**

**show detailed outlines of a specific state**

**1:10m**

**such as a specific city or even neighbor‐**

**hood, then you’ll need much higher-resolution data**

1. **Simplify the shapes**

**当找不到自己想要的分辨率shapefile,但是有更高的。**

[mapshaper](https://mapshaper.org/)

[TopoJSON - 简书 (jianshu.com)](https://www.jianshu.com/p/351fbc010412?from=singlemessage)

1. **Convert to GeoJSON**

[mapshaper](https://mapshaper.org/)

[ogr2ogr：矢量数据转换大杀器 · 语雀 (yuque.com)](https://www.yuque.com/geoway-vision/vision/ogr2ogr)

**好处是可以处理很多种投影的shapefile 但是安装麻烦**

[Ogre - ogr2ogr web client (adc4gis.com)](http://ogre.adc4gis.com/)

基于org2ogr的基于网页的客户端，功能弱一些但是够用了

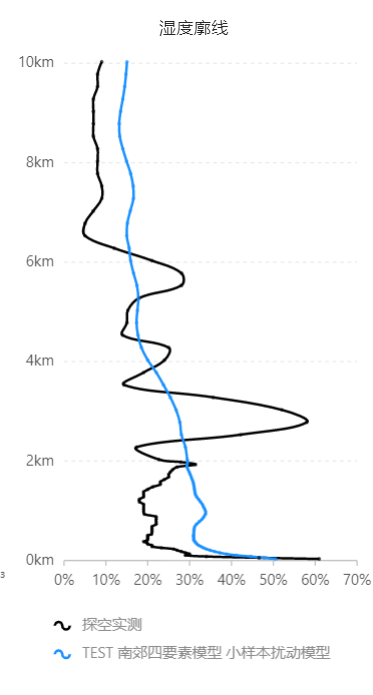
**5. Choose a projection**

如何选择投影？  
[Choosing the Right Map Projection - Learning - Source: An OpenNews project](https://source.opennews.org/articles/choosing-right-map-projection/)

**Note：**

1. **此外可能还有 flawed data obscure file format projections.等问题**

# 廓线图绘制：



为什么本地化系统里这个缩放会动？和vue有关？

为什么老师画的线咋么光滑？

CustomAxis函数里有几行代码没明白

感觉写起来很零碎