D3.js入门指南

D3是指数据驱动文档(Data-Driven Documents)。  
D3.js是一个JavaScript库，它可以通过数据来操作文档。它通过使用HTML、SVG和CSS把数据展现出来。D3不依赖SVG，使用SVG只是因为它适合画图。最新版本的D3也是支持Canvas的。

jQuery通过封装方便了我们操作DOM树，D3也通过自己的封装方便了我们操作SVG。

D3零依赖，函数式编程风格。可以参考下面示例理解D3的编程风格。

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>demo01</title>

<script src="../build/d3.js"></script>

</head>

<body>

<script>

function SimpleWidget(spec) {

// 定义私有变量

var instance = {};

var headline;

var description;

instance.render = function () {

var div = d3.select('body').append('div');

div.append('h3').text(headline);

div.attr('class', 'box')

.attr('style', 'color:' + spec.color)

.append('p')

.text(description);

return instance;

}

instance.headline = function (h) {

if (!arguments.length) {

return headline;

}

headline = h;

return instance;

}

instance.description = function (d) {

if (!arguments.length) {

return description;

}

description = d;

return instance;

}

return instance;

}

var widget = SimpleWidget({

color: '#6495ed'

})

.headline('hello world') // setter

.description('This is helloworld.');

console.log(widget.headline()); //getter

widget.render();

</script>

</body>

</html>

该示例很简单，就是在页面显示一下hello world。但D3的写作风格却可一窥究竟。  
**首先是链式风格：**  
instance.method1(…).method2(…).method3(…)，这种风格大家应该不陌生，jQuery也很早就有这种写法；  
**其次是可变参数：**  
setter注释行，当调用方法有参数时，即如methodName(params)时，可以理解为相应的setter方法，赋值之后返回一个实例；  
getter注释行，当调用方法没有参数时，即如methodName()时，可以理解为相应的getter方法，获取当前实例属性的值。

源码地址如下：  
<https://github.com/hankaibo/d3-demos/blob/master/demo01/index.html>