

[文章](#) [问答](#) [论坛](#) [东西](#) [休息室](#) [?](#)

Search for articles, questions,

手表



# SVG 世界地图



马塞洛·里卡多·德·奥利维拉

2011 年 10 月 5 日 警察

评价我: 4.92/5 (60 票)

使用 SVG 和 jQuery 的交互式世界地图。

[下载世界地图 - 1.31 MB](#)

## 目录

- [介绍](#)
- [系统要求](#)
- [SVG 案例](#)
- [jQuery SVG 插件](#)
- [对原始世界地图 SVG 文件进行逆向工程](#)
- [与 SVG 交互](#)
- [一些示例代码](#)
- [最后的考虑](#)
- [历史](#)

# 介绍

鉴于技术领域的最新发展（亲爱的未来读者：这是 2010 年 9 月），我们已经充斥着新闻和事实，表明 Web 开发将从现在开始受到更多关注。越来越多的移动设备（智能手机/平板电脑）和便携式设备（笔记本电脑/上网本）可以访问可靠和稳定的宽带互联网，这证明了这种趋势是合理的。

此外，一些重要的移动浏览器（例如 iPad、iPhone 中的 Safari）和 Windows 8 Metro 中的 IE10（请注意，这里我说的是 Metro 界面，专为平板电脑设计，而不是用于台式机/笔记本电脑的 Windows 8）。不/将不支持 Flash 和 Silverlight 等插件。这意味着，如果您是使用 Flash 或 Silverlight 的 Web 开发人员，如果您想将 Web 应用程序交付到此类设备，您将需要重新考虑您的策略。这不是 Android 平台的情况，但在这里我们已经（再次）达到了您作为 Web 开发人员必须确保您的应用程序在所有预期平台上运行的地步。Flash（以及最近的 Silverlight）已经普及多年，这对使用这些技术的开发人员来说是一件好事。在许多方面，

在玩了几个月的 HTML5 基础之后，我一直在等待合适的时机用 SVG 做点什么。虽然这篇文章是将我学到的一些 SVG 理论付诸实践的借口，但幸运的是，我最终创造了一些（希望）对我们 Web 开发人员有用的东西。

## 系统要求

为了运行本文附带的 SVG World Map 示例，您必须有一个支持 HTML5 SVG 的浏览器：



- Internet Explorer 8 或更高版本
- 火狐
- 苹果浏览器
- 铬合金

## SVG 案例



由于我有一些 WPF (Windows Presentation Foundation) 和 JavaScript 背景，我必须承认我对 SVG“语言”（类似于 WPF 几何元素）和操作 Web 元素的方法（当然是 jQuery）感到相当满意。当您可以使用以前的知识来学习新事物时，这很好，不是吗？

在这个项目中，我使用了 [jQuery SVG](#)，这是 **Keith Wood** 创建的一个很棒的 jQuery 插件，它允许您以编程方式与 SVG Canvas 进行交互。

例如，下面的 CIA Factbook 网站提供了一个非常漂亮的基于 Flash 的世界地图（免责声明：我与 CIA 没有任何联系）。当您将鼠标移到一个国家上时，它的领土会突出显示，并显示一个带有国家名称的小框。这是我想使用原生 HTML5/SVG/JavaScript 功能创建的应用程序类型，而不依赖于 Flash/Silverlight 等插件基础架构。



为了以编程方式创建一个新的 SVG，我首先从 Wikipedia ([http://upload.wikimedia.org/wikipedia/commons/a/ad/World\\_map\\_blank\\_with\\_blue\\_sea.svg](http://upload.wikimedia.org/wikipedia/commons/a/ad/World_map_blank_with_blue_sea.svg)) 下载一个 SVG 文件并将其拆解。有可能是一个 SVG 里面的许多元素，但为了简便起见，我们假设特定的 SVG 文件，我们使用有三个基本要素：**SVG**，**g** 和 **path**，**SVG** 作为顶级元素，而 **g** 代表的“团”内部元素，并且 **path** 是保存国家边界复杂几何形状的元素。

## jQuery SVG 插件

**jQuery SVG** 是由 [Keith Wood](#) 创建的一个很棒的插件。通常，使用标准的 jQuery 框架，您无法对 SVG 做很多事情。但幸运的是，jQuery SVG 让您更轻松。如果您熟悉 jQuery，那么您就会知道如何使用选择器访问 DOM 元素：

JavaScript

复制代码

```
var myDiv = $('#myDiv');
```

现在假设您要创建一个全新的 SVG 并将其附加到您的 **div**。您需要做的就是：

JavaScript

复制代码

```
var mySVG = $('#myDiv').svg();
```

从现在开始，您可能需要参考新创建的 SVG。jQuery SVG 允许您使用这样的选择器：

JavaScript

复制代码

```
var svg = $('#myDiv').svg('get');
```

最后，您可能想要绘制一个带有 3 像素粗红色笔划的黄色圆圈。无需深入研究 HTML，因为您可以通过编程方式完成：

JavaScript

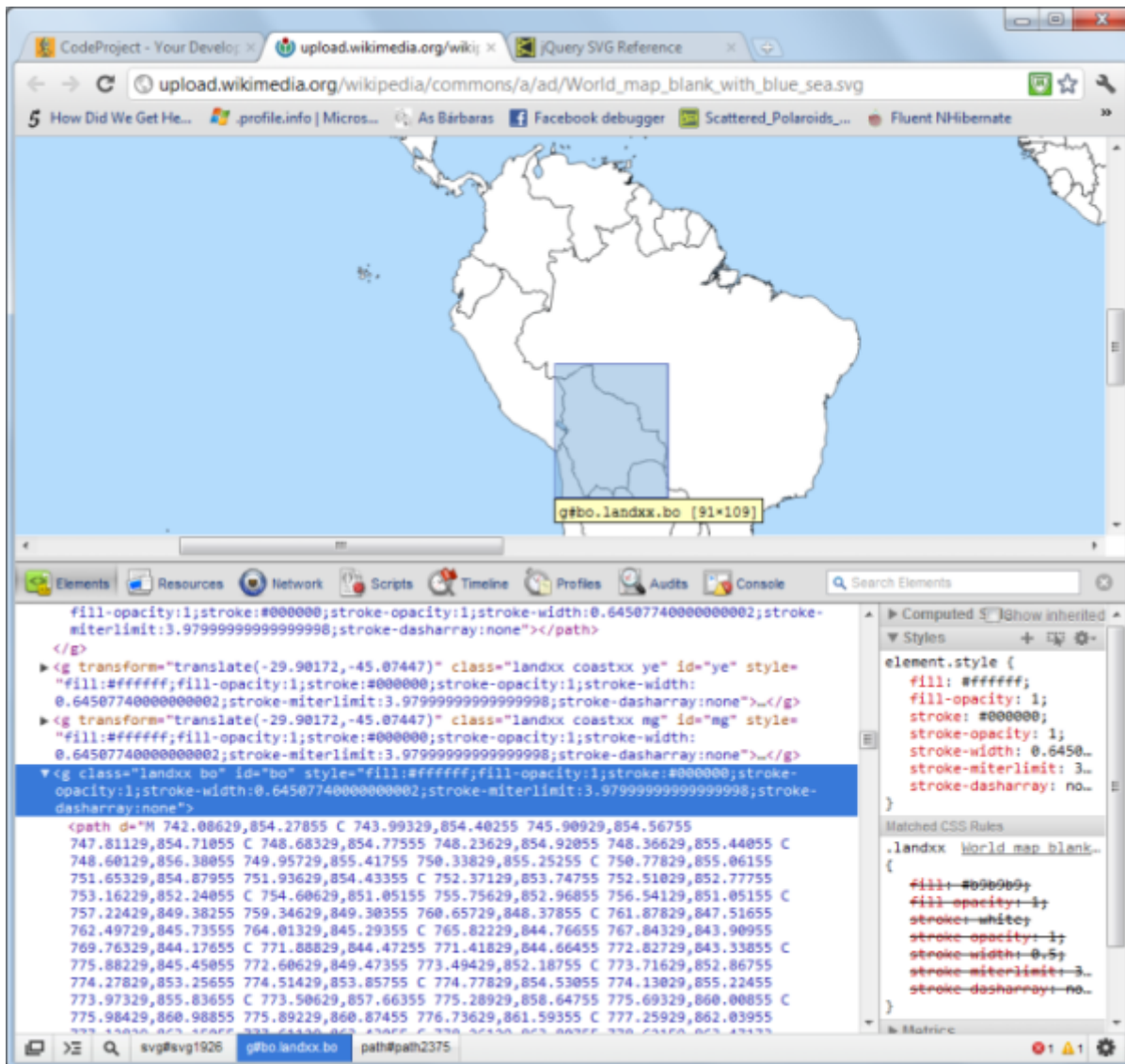
复制代码

```
svg.circle(100, 50, 50, {fill: 'yellow', stroke: 'red', strokeWidth: 3});
```

## 对原始世界地图 SVG 文件进行逆向工程

这是该项目真正困难的部分：我将原始.svg文件拆分为多个字符串数组（稍后存储在 JavaScript 文件中），每个国家一组。我称之为“逆向工程 SVG”。我都是手动完成的，有些国家没有包括在内，所以如果你的国家没有在地图中找到，请原谅我（顺便说一下，我可以将它们包括在文章的将来版本中）。

例如，下图显示了原始.svg文件中构成玻利维亚的元素。可以注意到 `g`element 和 `path`element 有很多信息：



HTML

复制代码

```
<g xmlns="http://www.w3.org/2000/svg" class="landxx bo" id="bo"
  style="fill:#ffffff;fill-opacity:1;stroke:#000000;
    stroke-opacity:1;stroke-width:0.6450774000000002;
    stroke-miterlimit:3.9799999999999998;stroke-dasharray:none">
  <path d="M 742.08629,854.27855 C 743.99329,854.40255 745.90929,854.56755 747.81129,
    854.71055 C 748.68329,854.77555 748.23629,854.92055 748.36629,855.44055
    C 748.60129,856.38055 749.95729,855.41755 750.33829,855.25255 C 750.77829,
    855.06155 751.65329,854.87955 751.93629,854.43355 C 752.37129,853.74755 752.51029,
    852.77755 753.16229,852.24055 C 754.60629,851.05155 755.75629,852.96855 756.54129,
    851.05155 C 757.22429,849.38255 759.34629,849.30355...">
  <path d="M 750.24716,899.89622 C 750.26405,899.89762 749.7766,
    899.86692 749.13517,899.68022 C 748.83492,
    899.59283 749.36535,898.78795 749.64135,898.64095...">
</g>
```

上面的 SVG 代码以字符串数组的形式转换为 JavaScript，如下所示：

JavaScript

复制代码

```
p = [];
p[p.length] = 'M 742.08629,854.27855 C 743.99329,854.40255 745.90929,
```

```

854.56755 747.81129,854.71055 C 748.68329,854.77555 748.23629,854.92055...
p[p.length] = 'M 750.24716,899.89622 C 750.26405,899.89762 749.7766,
899.86692 749.13517,899.68022 C 748.83492,899.59283 749.36535,898.78795...

var bo = { pathCollection: p, id: 'bo', translate: [29.90172, 45.07447] };

```

其中“bo”是保存国家“玻利维亚”数据的匿名对象。稍后在 JavaScript 文件中，当我们将一组国家/地区传递给drawCountries函数时，将创建南美洲大陆：

JavaScript

复制代码

```

var countries = [ar, bo, br, cl, co, ec, gf, gu, py, pe, sr, uy, ve, gy];
drawCountries(svg, scale, countries);

```

当我们运行drawCountries函数时，“逆向工程”就完成了。注意path.move, path.curveC和path.line方法，即移动，绘制曲线，并绘制SVG路径内的线，分别。

JavaScript

缩小▲ 复制代码

```

function drawCountries(svg, config, countries, translate) {
    //Let's draw one country at a time
    for (var c = 0; c < countries.length; c++) {
        var country = countries[c];
        var g;

        //code removed for sake of brevity

        //Each country has a collection of path. Let's draw them
        for (var i = 0; i < country.pathCollection.length; i++) {
            var splitted = country.pathCollection[i].split(' ');

            var path = svg.createPath();

            var index = 0;

            //Here we parse each path
            while (index < splitted.length) {
                var command = splitted[index];

                switch (command) {
                    //M x,y = "Move to point (x,y)", that is, the path
                    // will start at this absolute position (x,y)
                    case 'M':
                        var moveconfig1 = splitted[index + 1].split(',');
                        path.move(moveconfig1[0], moveconfig1[1]);
                        index += 2;
                        break;
                    //C x1,y1 x2,y2 x3,y3 = "Curve (x1,y1,x2,y2,x3,y3)",
                    // and draws a bézier curve using 3 control points (xn,yn)
                    case 'C':
                        var curveCconfig1 = splitted[index + 1].split(',');
                        var curveCconfig2 = splitted[index + 2].split(',');
                        var curveCconfig3 = splitted[index + 3].split(',');
                        path.curveC(curveCconfig1[0], curveCconfig1[1],
                                curveCconfig2[0], curveCconfig2[1],
                                curveCconfig3[0], curveCconfig3[1]
                                );
                        index += 4;
                        break;
                    //L x,y = "Straight Line (x,y)", a line segment starting
                    //at the current position and ending in (x,y) point
                    case 'L':
                        var lineconfig1 = splitted[index + 1].split(',');
                        path.line(lineconfig1[0], lineconfig1[1]);
                        index += 2;
                        break;
                }
            }
        }
    }
}

```



```

        svg.path(g, path, { id: country.id, countryId: country.id });
    }

    //code removed for sake of brevity
}
}

```

此时，我们已经重新创建了 SVG。那是困难的部分。现在是乐趣开始的地方，我们可以与 SVG 及其组件进行交互。

## 与 SVG 交互

为了监听 SVG 触发的鼠标事件，我们必须绑定适当的事件。在这个例子中，我们将绑定 **mousemove**，**mouseenter**，**mouseout**，和 **click**。当然，我们可以使用更多事件，但这些对于我们项目所需的功能来说已经足够了：

JavaScript

缩小▲ 复制代码

```

$(svg.root()).bind('mousemove',
function (path) {
    var offset = $(config.selector).position();
    $('#box').attr('transform',
        'translate(' + path.pageX + ' ' + path.pageY + ')');
});

$('#' + country.id, svg.root()).bind('mousemove',
function (path) {
    var g = path.target.parentNode;

    if (countryBoxFadeOut) {

        if (config.showCountryBoxOnMouseEnter &&
            (lastPoint[0] != path.pageX &&
            lastPoint[1] != path.pageY)) {
            showCountryBox(svg);
        }

        timer = setTimeout(function () {
            if (lastCountryId == currentCountryId) {
                clearTimeout(timer);
                hideCountryBox(svg);
            }
        }, 1000);
    }

    lastPoint = [path.pageX, path.pageY];
});

$('#' + country.id, svg.root()).bind('mouseenter',
function (path) {

    $('#box').attr('transform', 'translate(' + path.pageX + ' ' + path.pageY + ')');

    var g = path.target.parentNode;

    $(g).attr('opacity', config.activeCountryOpacity);
    $(g).attr('fill', config.activeCountryFill);
    $(g).attr('stroke', config.activeCountryStroke);
    $(g).attr('strokeWidth', config.activeCountryStrokeWidth);

    config.countryId = path.target.id;
    config.pos = [path.pageX, path.pageY];

    lastCountryId = currentCountryId;

    currentCountryId = config.countryId;

```

```

    if (config.showCountryBoxOnMouserEnter) {
        showCountryBox(svg);
        var box = $('#box', svg.root());
        $(box).stop();
        $(box).animate({ svgOpacity: 1.0 }, 100);
        var txt = $('#txtBox', svg.root());
        var name = getCountryName(config.countryId).split(',')[0];
        txt[0].textContent = name.toUpperCase();
    }

    timer = setTimeout(function () {
        if (lastCountryId == currentCountryId) {
            clearTimeout(timer);

            if (config.showCountryBoxOnMouserEnter)
                hideCountryBox(svg);
        }
    }, 1000);

    if (config.onmouseenter) {
        config.onmouseenter(config);
    }
});

$('#' + country.id, svg.root()).bind('mouseout',
function (path) {
    var g = path.target.parentNode;
    $(g).attr('fill', config.inactiveCountryFill);
    $(g).attr('opacity', config.inactiveCountryOpacity);
    $(g).attr('stroke', config.inactiveCountryStroke);
    $(g).attr('strokeWidth', config.inactiveCountryStrokeWidth);
    $('#box').stop();
});

$('#' + country.id, svg.root()).bind('click',
function (path) {
    if (config.onclick) {
        var g = path.target.parentNode;

        config.onclick(getCountryName(g.id));
    }
});

```

## 一些示例代码

在继续示例代码之前，让我们看一下 SVG 世界地图所需的一些配置：

### 配置选项

- **id**: (string) 地图的编程 ID
- **selector**: (string) 将附加地图 SVG 元素的 CSS 选择器
- **scale**: (数字) 用于渲染地图的比例。
- **margin**: (string) 地图 SVG 元素的 CSS 边距
- **top**: (string) 地图 SVG 元素的 CSS 上边距
- **height**: (number) 地图 SVG 元素的 CSS 高度
- **width**: (number) 地图 SVG 元素的 CSS 宽度
- **inactiveCountryFill**: (string) 用于填充非活动国家的 CSS 填充
- **inactiveCountryStroke**: (string) 用于勾勒不活跃国家的 CSS 笔画
- **inactiveCountryStrokeWidth**: (数字) 用于勾勒非活动国家的笔画粗细
- **activeCountryFill**: (string) 用于填充活跃国家的 CSS 填充
- **activeCountryStroke**: (字符串) 用于勾勒活动国家/地区的 CSS 笔划
- **activeCountryStrokeWidth**: (数字) 用于勾勒活动国家的笔画粗细

- **showCountryBoxOnMouserEnter**: (bool) 确定用户将鼠标移到国家上时是否必须显示名称框
- **drawNorthAmerica**: (bool) 决定是否应该渲染北美
- **drawCentralAmerica**: (bool) 确定是否应呈现中美洲
- **drawSouthAmerica**: (bool) 决定是否应该渲染南美洲
- **drawEurope**: (bool) 决定是否应该渲染欧洲
- **drawAfrica**: (bool) 决定是否应该渲染非洲
- **drawAsia**: (bool) 决定是否应该渲染亚洲
- **drawOceania**: (bool) 确定是否应该渲染大洋洲
- **drawAntarctic**: (bool) 决定是否应该渲染南极洲

## 活动

- ⚡ **onCountryMouseEnter**: 当鼠标进入一个国家时触发
- ⚡ **onCountryMouseMove**: 当鼠标移过一个国家时触发
- ⚡ **onCountryMouseOut**: 当鼠标离开一个国家时触发
- ⚡ **onCountryMouseClicked**: 当鼠标点击一个国家时触发



JavaScript

缩小▲ 复制代码

```
var wm1 = WorldMap({
  id: 'map1',
  selector: '#svgWorldMap1',
  scale: 0.2,
  margin: '0',
  top: '50',
  height: '300',
  width: '550',
  inactiveCountryFill: '#4af',
  inactiveCountryStroke: '#fff',
  inactiveCountryStrokeWidth: 6,
  showCountryBoxOnMouserEnter: true,
  drawNorthAmerica: true,
  drawCentralAmerica: true,
  drawSouthAmerica: true,
  drawEurope: true,
  drawAfrica: true,
  drawAsia: true,
  drawOceania: true,
  drawAntarctic: true,
  onCountryMouseEnter: function (config) {
    var id = config.countryId;
  },
  onCountryMouseMove: function (config) {
```



```

        var id = config.countryId;
    },
    onCountryMouseOut: function (config) {
        var id = config.countryId;
    },
    onCountryMouseClicked: function (countryId) {
        var id = countryId;
    }
});

```



JavaScript

缩小▲ 复制代码

```

var wm2 = WorldMap({
    id: 'map2',
    selector: '#svgWorldMap2',
    scale: 0.2,
    margin: '0',
    top: '50',
    height: '300',
    width: '550',
    inactiveCountryFill: 'transparent',
    inactiveCountryStroke: '#ccc',
    inactiveCountryStrokeWidth: 4,
    activeCountryFill: 'orange',
    activeCountryStroke: '#ccc',
    activeCountryStrokeWidth: 0,
    showCountryBoxOnMouserEnter: false,
    drawNorthAmerica: true,
    drawCentralAmerica: true,
    drawSouthAmerica: true,
    drawEurope: true,
    drawAfrica: true,
    drawAsia: true,
    drawOceania: true,
    drawAntarctic: true,
    onCountryMouseEnter: function (config) {
        var id = config.countryId;
    },
    onCountryMouseMove: function (config) {
        var id = config.countryId;
    },
    onCountryMouseOut: function (config) {
        var id = config.countryId;
    },
    onCountryMouseClicked: function (countryId) {
        var id = countryId;
    }
});

```

```

    }
  });

```



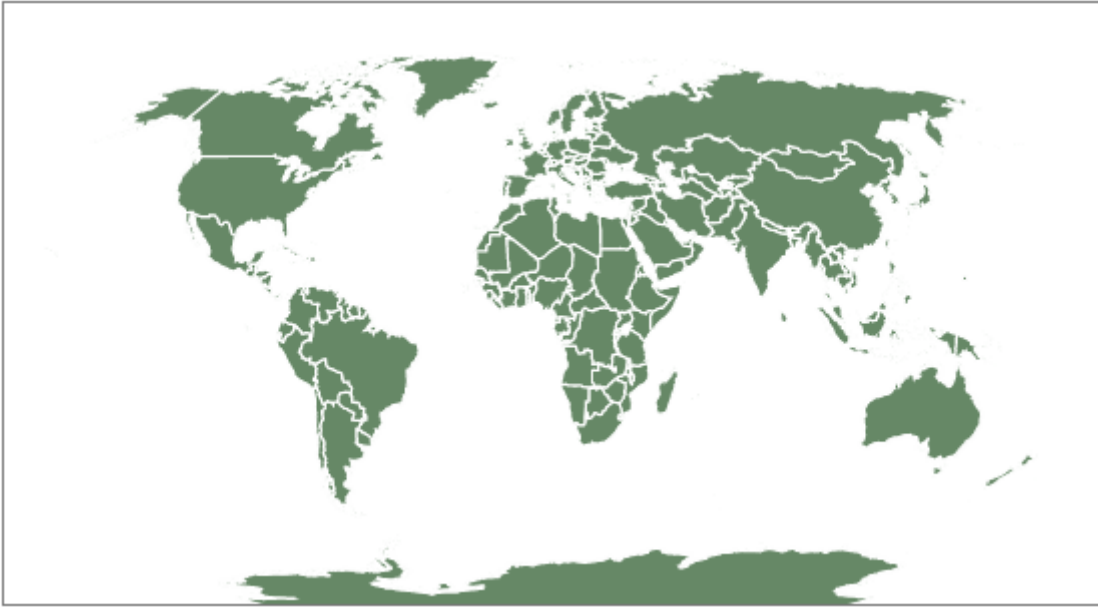
JavaScript

缩小▲ 复制代码

```

var wm3 = WorldMap({
  id: 'map3',
  selector: '#svgWorldMap3',
  scale: 0.2,
  margin: '0',
  top: '50',
  height: '300',
  width: '550',
  inactiveCountryFill: '#ccc',
  inactiveCountryStroke: 'gray',
  inactiveCountryStrokeWidth: 6,
  activeCountryFill: 'orange',
  activeCountryStroke: 'gray',
  activeCountryStrokeWidth: 6,
  showCountryBoxOnMouseEnter: true,
  drawNorthAmerica: true,
  drawCentralAmerica: true,
  drawSouthAmerica: true,
  drawEurope: true,
  drawAfrica: true,
  drawAsia: true,
  drawOceania: true,
  drawAntarctic: true,
  onCountryMouseEnter: function (config) {
    var id = config.countryId;
  },
  onCountryMouseMove: function (config) {
    var id = config.countryId;
  },
  onCountryMouseOut: function (config) {
    var id = config.countryId;
  },
  onCountryMouseClicked: function (countryId) {
    var id = countryId;
  }
});

```



JavaScript

缩小▲ 复制代码

```
var wm4 = WorldMap({
  id: 'map4',
  selector: '#svgWorldMap4',
  scale: 0.2,
  margin: '0',
  top: '50',
  height: '300',
  width: '550',
  inactiveCountryFill: '#686',
  inactiveCountryStroke: '#fff',
  inactiveCountryStrokeWidth: 6,
  showCountryBoxOnMouseEnter: false,
  drawNorthAmerica: true,
  drawCentralAmerica: true,
  drawSouthAmerica: true,
  drawEurope: true,
  drawAfrica: true,
  drawAsia: true,
  drawOceania: true,
  drawAntarctic: true,
  onCountryMouseEnter: function (config) {
    var id = config.countryId;
  },
  onCountryMouseMove: function (config) {
    var id = config.countryId;
  },
  onCountryMouseOut: function (config) {
    var id = config.countryId;
  },
  onCountryMouseClicked: function (countryId) {
    var id = countryId;
  }
});
```

## 最后的考虑

在这里，我们使用**HTML5 SVG**完成了我们的小项目。请记住，SVG 是您的朋友，它背后蕴藏着巨大的潜力，由您来探索它。

代码不是 100% 完善的，当然还有许多有趣的改进空间。随意使用它，如果你这样做，我会很高兴。

如果您到达了这条线，那么非常感谢您的耐心等待。我希望这篇文章在某种程度上是有用的，或者至少是有趣的。

## 历史

- 2011-09-30: 初始版本。

## 执照

本文以及任何相关的源代码和文件均根据[The Code Project Open License \(CPOl\)](#)获得许可

## 分享

## 关于作者



### 马塞洛·里卡多·德·奥利维拉

讲师/培训师 Alura Cursos Online  
巴西 🇧🇷

手表  
该会员

Marcelo Ricardo de Oliveira 是一名高级自由软件开发人员，与他可爱的妻子 Luciana 和他的小伙伴兼继子 Kauê 住在巴西瓜鲁柳斯，是巴西电视指南[电视地图](#)的联合创始人，目前为[Alura Cursos Online](#) 工作。

他经常从事严肃的企业项目，尽管在业余时间他尝试编写有趣的代码项目文章.....

[展示更多](#)

## 评论和讨论

添加评论或问题



Email Alerts

Search Comments



First Prev Next

**Easier alternative** 🌟

Anna Ivanov 8-Jul-20 20:44

**Microsoft JScript runtime error** 🌟

Member 9601343 19-Feb-13 14:35

**MuitObrigado** 🌟

Joan Blanq 9-Feb-13 2:51

**Excellent work...** 📌📌

ian sexton 2-Jan-13 2:30

---

**Great !!** 📌📌

Member 9427491 13-Sep-12 20:08

---

**My vote of 5** 📌📌

Raj20164 29-Jul-12 21:39

---

**Great Job** 📌📌

Member 1965191 19-Mar-12 10:00

---

Re: Great Job 📌📌

Marcelo Ricardo de Oliveira 20-Mar-12 22:34

---

**My vote of 5** 📌📌

Akram El Assas 10-Mar-12 18:14

---

Re: My vote of 5 📌📌

Marcelo Ricardo de Oliveira 20-Mar-12 22:33

---

**My vote of 5** 📌📌

Paulo Bueno 12-Dec-11 12:51

---

Re: My vote of 5 📌📌

Marcelo Ricardo de Oliveira 2-Mar-12 9:45

---

**Works nice. Future request.** 📌📌

@dri@n 23-Nov-11 20:10

---

Re: Works nice. Future request. 📌📌

Marcelo Ricardo de Oliveira 2-Mar-12 9:45

---

**sweden** 📌📌

Seçkin 25-Oct-11 13:08

---

Re: sweden 📌📌

Marcelo Ricardo de Oliveira 2-Mar-12 9:44

---

**My vote of 5** 📌📌

jawed.ace 20-Oct-11 20:06

---

Re: My vote of 5 📌📌

Marcelo Ricardo de Oliveira 2-Mar-12 9:44

---

**My vote of 5** 📌📌

User 6255464 18-Oct-11 19:52

---

Re: My vote of 5 📌📌

Marcelo Ricardo de Oliveira 18-Oct-11 22:38

---

**My vote of 5** 📌📌

Abhijit Jana 14-Oct-11 21:11

---

回复: 我的5票 📌📌

Marcelo Ricardo de Oliveira 15-Oct-11 22:16

---

**投票 5** 📌📌

AditSheth 13-Oct-11 12:16

---

回复: 投票 5

Marcelo Ricardo de Oliveira 14-Oct-11 6:08

我的5票

Ahsan Murshed 12-Oct-11 16:44

刷新

123 下一页

一般

新闻

建议

问题

错误

答案

笑话

赞美

咆哮

管理员

使用Ctrl+Left/Right 切换消息，Ctrl+Up/Down 切换主题，Ctrl+Shift+Left/Right 切换页面。

永久链接  
广告  
隐私  
Cookie  
使用条款

布局: 固定 | 体液

文章版权所有 2011 年 Marcelo Ricardo de Oliveira  
其他所有内容版权所有 © CodeProject ,  
1999-2021 Web03 2.8.20210930.1