



链滴

## 绘画板 06——铅笔与自定义鼠标图标

作者: [crick77](#)

原文链接: <https://ld246.com/article/1474711147009>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

github地址: <https://github.com/wangyuheng/painter>

DEMO地址: <http://painter.crick.wang/>

## 自定义鼠标图标

可以通过css设置cursor的方式指定鼠标图标样式

- | default | 默认光标 (通常是一个箭头) |
- | auto | 默认。浏览器设置的光标。 |
- | crosshair | 光标呈现为十字线。 |
- | pointer | 光标呈现为指示链接的指针 (一只手) |
- | move | 此光标指示某对象可被移动。 |
- | e-resize | 此光标指示矩形框的边缘可被向右 (东) 移动。 |
- | ne-resize | 此光标指示矩形框的边缘可被向上及向右移动 (北/东) 。 |
- | nw-resize | 此光标指示矩形框的边缘可被向上及向左移动 (北/西) 。 |
- | n-resize | 此光标指示矩形框的边缘可被向上 (北) 移动。 |
- | se-resize | 此光标指示矩形框的边缘可被向下及向右移动 (南/东) 。 |
- | sw-resize | 此光标指示矩形框的边缘可被向下及向左移动 (南/西) 。 |
- | s-resize | 此光标指示矩形框的边缘可被向下移动 (南) 。 |
- | w-resize | 此光标指示矩形框的边缘可被向左移动 (西) 。 |
- | text | 此光标指示文本。 |
- | wait | 此光标指示程序正忙 (通常是一只表或沙漏) 。 |
- | help | 此光标指示可用的帮助 (通常是一个问号或一个气球) 。 |

也可以通过url指定自定义的图标样式, 通常为.cur文件。

```
$("#svgPanel").css("cursor", "url(style/img/cur/tool_pencil.cur), auto");
```

auto为指定图标不存在时, 鼠标显示的样式。

## 铅笔

通过svg的path属性, 可以绘制出鼠标移动轨迹, 实现铅笔画图效果。svgjs提供了path()函数, 值为

- M = moveto
- L = lineto
- H = horizontal lineto
- V = vertical lineto
- C = curveto
- S = smooth curveto
- Q = quadratic Belzier curve

- T = smooth quadratic Belzier curveto
- A = elliptical Arc
- Z = closepath

所以mousedown时做 M+坐标点，mousemove时，为L+坐标点，可以指定不同的路径数据以达到不同的绘制效果，所以在构造函数中增加了一个prefix字段。

pencil.js 代码如下

```
(function() {

    var parent = null;
    var drawing = false;
    var element = null;
    var startPoint = null;

    var plot = null;
    var plotPrefix = null;
    var defaultPlotPrefix = 'L';

    function mousedown(event) {
        console.log('pencil mousedown');
        drawing = true;
        startPoint = svgDoc.transformPoint(event);
        plot = 'M' + startPoint.x + ' ' + startPoint.y;
        element = parent.path(plot).fill(GlobalStatus.getFillColor()).style("fill-opacity", GlobalStat
s.getFillOpacity()).stroke({
            width: GlobalStatus.getLineSize(),
            color: GlobalStatus.getFontColor()
        });
        return false;
    }

    function mousemove(event) {
        console.log('pencil mousemove');
        if (drawing) {
            var startPoint = svgDoc.transformPoint(event);
            console.log(plot);
            plot += plotPrefix + startPoint.x + ' ' + startPoint.y;
            element.plot(plot);
        }
        return false;
    };

    function mouseup(event) {
        console.log('pencil mouseup ' + element);
        drawing = false;
        if (element.attr("d").split(plotPrefix).length > 2) {
            element.pickable();
        }
        return false;
    }
}
```

```

var listener = {
  mousedown: mousedown,
  mousemove: mousemove,
  mouseup: mouseup,
};

var Pencil = function(parentEle, prefix) {
  parent = parentEle;
  svgDoc = parent.doc();
  DrawTool.init(svgDoc, listener);
  plotPrefix = prefix || defaultPlotPrefix;
  this.stop = function() {
    DrawTool.stop(svgDoc, listener);
  };
};

this.DrawTool.Pencil = Pencil;

})();

```

在首页监听铅笔的点击事件，并在页面加载完成后默认调用 `$("#tool_pencil").click()`;

```

$("#tool_pencil").on("click", function() {
  resetCurrentDrawTool();
  currentDrawTool = new DrawTool.Pencil(svgDoc);
  $("#svgPanel").css("cursor", "url(style/img/cur/tool_pencil.cur), auto");
});

```

加了一个彩蛋，监听铅笔按钮的双击事件，可以绘制映射线。

```

$("#tool_pencil").on("dblclick", function() {
  resetCurrentDrawTool();
  currentDrawTool = new DrawTool.Pencil(svgDoc, 'T');
  $("#svgPanel").css("cursor", "url(style/img/cur/tool_pencil.cur), auto");
});

```

## 附

在Tools的listener方法中，都加了一个return false; 避免事件向上传递。

在模拟的下拉列表lineSize中，通过属性data-line-size记录width值，在GlobalStatus.getLineSize()调用。

将已选择元素抽象到在GlobalStatus中，并提供管理方法。

```

pickedElementList: [],
pushPicked: function(o){
  return this.pickedElementList.push(o);
},

```

```
removePicked: function(o) {  
    return this.pickedElementList.remove(o);  
},  
getPicked: function(){  
    return this.pickedElementList;  
}
```