

[← jQuery UI 实例 – 定位 \( Position \)](#)

## jQuery UI 实例 - 部件库 ( Widget Factory )

使用与所有 jQuery UI 小部件相同的抽象化来创建有状态的 jQuery 插件。

如需了解更多有关部件库 ( Widget Factory ) 的细节，请查看 API 文档 [部件库 \( Widget Factory \)](#)。

### 默认功能

该演示展示了一个简单的使用部件库 ( jquery.ui.widget.js ) 创建的自定义的小部件。

三个区块是以不同的方式初始化的。点击改变他们的背景颜色。查看源码及注释理解工作原理。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery UI 部件库 (Widget Factory) - 默认功能</title>
  <link rel="stylesheet" href="//code.jquery.com/ui/1.10.4/themes/smoothness/jquery-ui.css">
  <script src="//code.jquery.com/jquery-1.9.1.js"></script>
  <script src="//code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
  <link rel="stylesheet" href="http://jqueryui.com/resources/demos/style.css">
  <style>
    .custom-colorize {
      font-size: 20px;
      position: relative;
      width: 75px;
      height: 75px;
    }
    .custom-colorize-changer {
      font-size: 10px;
      position: absolute;
      right: 0;
      bottom: 0;
    }
  </style>
  <script>
    $(function() {
      // 部件定义，其中 "custom" 是命名空间，"colorize" 是部件名称
      $.widget( "custom.colorize", {
        // 默认选项
        options: {
          red: 255,
          green: 0,
          blue: 0,
```

```
// 回调
change: null,
random: null
},

// 构造函数
_create: function() {
    this.element
        // 添加一个主题化的 class
        .addClass( "custom-colorize" )
        // 防止双击来选择文本
        .disableSelection();

    this.changer = $( "<button>", {
        text: "改变",
        "class": "custom-colorize-changer"
    })
    .appendTo( this.element )
    .button();

    // 绑定 changer 按钮上的 click 事件到 random 方法
    this._on( this.changer, {
        // 当小部件被禁用时, _on 不会调用 random
        click: "random"
    });
    this._refresh();
},

// 当创建及之后改变选项时调用
_refresh: function() {
    this.element.css( "background-color", "rgb(" +
        this.options.red + "," +
        this.options.green + "," +
        this.options.blue + ")"
    );

    // 触发一个回调/事件
    this._trigger( "change" );
},

// 一个改变颜色为随机值的公共方法
// 可通过 .colorize( "random" ) 直接调用
random: function( event ) {
    var colors = {
        red: Math.floor( Math.random() * 256 ),
        green: Math.floor( Math.random() * 256 ),
        blue: Math.floor( Math.random() * 256 )
    };
};
```

```
// 触发一个事件, 检查是否已取消
if ( this._trigger( "random", event, colors ) !== false ) {
    this.option( colors );
}
},

// 自动移除通过 _on 绑定的事件
// 在这里重置其他的修改
_destroy: function() {
    // 移除生成的元素
    this.changer.remove();

    this.element
        .removeClass( "custom-colorize" )
        .enableSelection()
        .css( "background-color", "transparent" );
},

// _setOptions 是通过一个带有所有改变的选项的哈希来调用的
// 当改变选项时总是刷新
_setOptions: function() {
    // _super 和 _superApply
    this._superApply( arguments );
    this._refresh();
},

// _setOption 是为每个独立的改变的选项调用的
_setOption: function( key, value ) {
    // 防止无效的颜色值
    if ( /red|green|blue/.test(key) && (value < 0 || value > 255) ) {
        return;
    }
    this._super( key, value );
}
});

// 通过默认选项进行初始化
$( "#my-widget1" ).colorize();

// 通过两个自定义的选项进行初始化
$( "#my-widget2" ).colorize({
    red: 60,
    blue: 60
});

// 通过自定义的 green 值和一个只允许颜色足够绿的随机的回调进行初始化
$( "#my-widget3" ).colorize( {
    green: 128,
    random: function( event, ui ) {
```

```
        return ui.green > 128;
    }
});

// 点击切换 enabled/disabled
$( "#disable" ).click(function() {
    // 为每个小部件使用自定义的选择器来找到所有的实例
    // 所有的实例一起切换，所以我们可以从第一个开始检查状态
    if ( $( ":custom-colorize" ).colorize( "option", "disabled" ) ) {
        $( ":custom-colorize" ).colorize( "enable" );
    } else {
        $( ":custom-colorize" ).colorize( "disable" );
    }
});

// 在初始化后，点击设置选项
$( "#black" ).click( function() {
    $( ":custom-colorize" ).colorize( "option", {
        red: 0,
        green: 0,
        blue: 0
    });
});
});
</script>
</head>
<body>

<div>
    <div id="my-widget1">改变颜色</div>
    <div id="my-widget2">改变颜色</div>
    <div id="my-widget3">改变颜色</div>
    <button id="disable">切换 disabled 选项</button>
    <button id="black">改为黑色</button>
</div>

</body>
</html>
```

[查看演示](#)[← jQuery UI 实例 – 定位 \( Position \)](#)[✎ 点我分享笔记](#)

