

## jQuery UI 小部件 ( Widget ) 方法调用

小部件 ( Widget ) 是通过 [部件库 \( Widget Factory \)](#) 使用方法来改变他们初始化后的状态和执行动作而被创建的。有两种调用小部件方法的方式 - 通过部件库 ( Widget Factory ) 创建的插件，或者通过调用元素实例对象上的方法。

### 插件调用

使用小部件的插件调用方法，把方法名称以字符串形式进行传递。例如，[点击这里查看](#)，如何调用 [dialog \( 对话框 \) 小部件的 close\(\) 方法](#)。

```
$( ".selector" ).dialog( "close" );
```

如果方法要求参数，请作为额外的参数传递给插件。[点击这里查看](#)，如何调用 [dialog \( 对话框 \) 的 option\(\) 方法](#)。

```
$( ".selector" ).dialog( "option", "height" );
```

这会返回 [dialog \( 对话框 \) 的 height 选项](#) 的值。

### 实例调用

每个小部件的每个实例都是使用 [jQuery.data\(\)](#) 存储在元素上。为了检索实例对象，请使用小部件的全称作为键名调用 `jQuery.data()`。具体如下面实例所示。

```
var dialog = $( ".selector" ).data( "ui-dialog" );
```

在您引用实例对象之后，可以直接在上面调用方法。

```
var dialog = $( ".selector" ).data( "ui-dialog" );
dialog.close();
```

在 jQuery UI 1.11 中，新的 `instance()` 方法会使得这个过程变得更简单。

```
$( ".selector" ).dialog( "instance" ).close();
```

### 返回类型

大多数通过小部件的插件调用的方法将返回一个 jQuery 对象，所以方法调用可以通过额外的 jQuery 方法链接。当在实例上进行调用时，则会返回 `undefined`。具体如下面实例所示。

```
var dialog = $( ".selector" ).dialog();
```

```
// Instance invocation - returns undefined
dialog.data( "ui-dialog" ).close();

// Plugin invocation - returns a jQuery object
dialog.dialog( "close" );

// Therefore, plugin method invocation makes it possible to
// chain method calls with other jQuery functions
dialog.dialog( "close" )
    .css( "color", "red" );
```

例外的是，返回小部件相关信息的那些方法。例如 [dialog \(对话框\) 的 isOpen\(\) 方法](#)。

```
$( ".selector" )
    .dialog( "isOpen" )
    // This will throw a TypeError
    .css( "color", "red" );
```

这会产生一个 `TypeError` 错误，因为 `isOpen()` 返回的是一个布尔值，而不是一个 jQuery 对象。

← jQuery UI 通过部件库扩展小部件

jQuery UI 为什么使用部件库 →

 点我分享笔记