## 安装

为了跟踪部件的状态，我们首先介绍一下小部件的全生命周期。当小部件安装时，生命周期开始。我们只需要在一个或多个元素上调用插件，即安装了小部件。

$( "#elem" ).progressbar();

这将会初始化 jQuery 对象中的每个元素，在本例中，元素 id 为 "elem"。因为我们调用无参数的 .progressbar() 方法，小部件则会按照它的默认选项进行初始化。我们可以在安装时传递一组选项，这样既可重写默认选项。

$( "#elem" ).progressbar({ value: 20 });

## 方法

既然小部件已经初始化，我们就可以查询它的状态，或者在小部件上执行动作。所有初始化后的动作都以方法调用的形式进行。为了在小部件上调用一个方法，我们可以向 jQuery 插件传递方法的名称。例如，为了在进度条（progressbar）小部件上调用 value 方法，我们应该使用：

$( "#elem" ).progressbar( "value" );

如果方法接受参数，我们可以在方法名后传递参数。例如，为了传递参数 40 给 value 方法，我们可以使用：

$( "#elem" ).progressbar( "value", 40 );

就像 jQuery 中的其他方法一样，大部分的小部件方法为链接返回 jQuery 对象。

$( "#elem" )

.progressbar( "value", 90 )

.addClass( "almost-done" );

### **公共的方法**

每个小部件都有它自己的一套基于小部件所提供功能的方法。然而，有一些方法是所有小部件都共同具有的。

#### option

正如我们前面所提到的，我们可以在初始化之后通过 option 方法改变选项。例如，我们可以通过调用 option 方法改变 progressbar（进度条）的 value 为 30。

$( "#elem" ).progressbar( "option", "value", 30 );

请注意，这与之前我们调用 value 方法的实例有所不同。在本实例中，我们调用 option 方法，改变 value 选项为 30。

我们也可以为某个选项获取当前的值。

$( "#elem" ).progressbar( "option", "value" );

另外，我们可以通过给 option 方法传递一个对象，一次更新多个选项。

$( "#elem" ).progressbar( "option", {

value: 100,

disabled: true});

#### disable

disable 方法禁用小部件。在进度条（progressbar）实例中，这会改变样式让进度条显示为禁用状态。

$( "#elem" ).progressbar( "disable" );

#### enable

enable 方法是 disable 方法的对立面。

$( "#elem" ).progressbar( "enable" );

#### destroy

如果您不再需要小部件，那么可以销毁它，返回到最初的标记。这意味着小部件生命周期的终止。

$( "#elem" ).progressbar( "destroy" );

#### widget

一些小部件生成包装器元素，或与原始元素断开连接的元素。在下面的实例中，widget 将返回生成的元素。在进度条（progressbar）实例中，没有生成的包装器，widget 方法返回原始的元素。

$( "#elem" ).progressbar( "widget" );

## 事件

所有的小部件都有跟他们各种行为相关的事件，用于在状态改变时通知您。

例如，我们可以绑定进度条（）的 change 事件，一旦值发生变化时就触发。

$( "#elem" ).bind( "progressbarchange", function() {

alert( "The value has changed!" );});

每个事件都有一个相对应的回调，作为选项进行呈现。我们可以使用进度条（progressbar）的 change 回调，这等同于绑定 progressbarchange 事件。

$( "#elem" ).progressbar({

change: function() {

alert( "The value has changed!" );

}});