

## jQuery hide() 和 show()

通过 jQuery，您可以使用 hide() 和 show() 方法来**隐藏和显示** HTML 元素：

```
1      <button id="hide">隐藏</button>
2      <button id="show">显示</button>
3      <div class="a">
4          123
5      </div>
6      <script>
7          $("#hide").click(function(){
8              //隐藏
9              $(".a").hide();
10         });
11
12         $("#show").click(function(){
13             //显示
14             $(".a").show();
15         });
16     </script>
```

### 语法

```
1  $(selector).hide(speed,callback);
2
3  $(selector).show(speed,callback);
```

可选的 `speed` 参数规定隐藏/显示的速度，可以取以下值："slow"、"fast" 或毫秒。

可选的 `callback` 参数是隐藏或显示完成后所执行的函数名称。

下面的例子演示了带有 `speed` 参数的 `hide()` 方法：

```
1      <button id="hide">隐藏</button>
2      <button id="show">显示</button>
3      <div class="a">
4          123
5      </div>
6      <script>
7          $("#hide").click(function(){
8              $(".a").hide(1000);
9          });
10
11         $("#show").click(function(){
12             $(".a").show(1000);
13         });
14     </script>
```

下面的例子演示了带有 `speed` 参数的 `hide()` 方法，并使用回调函数

```
1      <button id="hide">隐藏</button>
2      <button id="show">显示</button>
3      <div class="a">
```

```

4      123
5    </div>
6    <script>
7      $("#hide").click(function(){
8        $(".a").hide(1000,function(){
9          alert('隐藏已经完成了');
10        });
11      });
12
13      $("#show").click(function(){
14        $(".a").show(1000);
15      });
16    </script>

```

## jQuery toggle()

可以使用 `toggle()` 方法来切换 `hide()` 和 `show()` 方法。

显示被隐藏的元素，并隐藏已显示的元素

语法

```
1 $(selector).toggle(speed,callback);
```

例子

```

1    <button id="btn">集成按钮，可以显示也可以隐藏</button>
2    <div class="a">
3      123
4    </div>
5    <script>
6      $('#btn').click(function(){
7        $('.a').toggle(1000,function(){
8          console.log('隐藏或者显示');
9        });
10      });
11    </script>

```

## jQuery 效果 - 淡入淡出

### jQuery fadeIn() 方法 淡入已隐藏的元素

语法

```
1 $(selector).fadeIn(speed,callback);
```

可选的 `speed` 参数规定效果的时长。它可以取以下值："slow"、"fast" 或 毫秒。

可选的 `callback` 参数是 `fading` 完成后所执行的函数名称。

下面的例子演示了带有不同参数的 `fadeIn()` 方法：

```
1 <button id="btn">按钮</button>
2 <div id="div1" style="display:none;">div1</div>
3 <div id="div2" style="display:none;">div2</div>
4 <div id="div3" style="display:none;">div3</div>
5 <script>
6     $('#btn').click(function(){
7         $('#div1').fadeIn();
8         $('#div2').fadeIn("slow");
9         $('#div3').fadeIn(3000);
10    })
11 </script>
```

## jQuery fadeOut() 方法 淡出可见元素

语法

```
1 $(selector).fadeOut(speed,callback);
```

可选的 `speed` 参数规定效果的时长。它可以取以下值："slow"、"fast" 或 毫秒。

可选的 `callback` 参数是 `fading` 完成后所执行的函数名称。

下面的例子演示了带有不同参数的 `fadeOut()` 方法：

```
1 <button id="btn">按钮</button>
2 <div id="div1">div1</div>
3 <div id="div2">div2</div>
4 <div id="div3">div3</div>
5 <script>
6     $('#btn').click(function(){
7         $('#div1').fadeOut();
8         $('#div2').fadeOut("slow");
9         $('#div3').fadeOut(3000);
10    })
11 </script>
```

## jQuery fadeToggle() 方法

jQuery `fadeToggle()` 方法可以在 `fadeIn()` 与 `fadeOut()` 方法之间进行切换。

如果元素已淡出，则 `fadeToggle()` 会向元素添加淡入效果。

如果元素已淡入，则 `fadeToggle()` 会向元素添加淡出效果。

语法

```
1 $(selector).fadeToggle(speed,callback);
```

可选的 `speed` 参数规定效果的时长。它可以取以下值："slow"、"fast" 或 毫秒。

可选的 `callback` 参数是 `fading` 完成后所执行的函数名称。

下面的例子演示了带有不同参数的 `fadeToggle()` 方法：

```
1 <button id="btn">按钮</button>
2 <div id="div1">div1</div>
3 <div id="div2">div2</div>
4 <div id="div3">div3</div>
5 <script>
6     $('#btn').click(function(){
7         $("#div1").fadeToggle();
8         $("#div2").fadeToggle("slow");
9         $("#div3").fadeToggle(3000);
10    })
11 </script>
```

## jQuery fadeTo() 方法

jQuery `fadeTo()` 方法允许渐变为给定的不透明度（值介于 0 与 1 之间）。

语法

```
1 $(selector).fadeTo(speed,opacity,callback);
```

必需的 `speed` 参数规定效果的时长。它可以取以下值："slow"、"fast" 或 毫秒。

`fadeTo()` 方法中必需的 `opacity` 参数将淡入淡出效果设置为给定的不透明度（值介于 0 与 1 之间）。

可选的 `callback` 参数是该函数完成后所执行的函数名称。

下面的例子演示了带有不同参数的 `fadeTo()` 方法：

```
1 <button id="btn">按钮</button>
2 <div id="div1">div1</div>
3 <div id="div2">div2</div>
4 <div id="div3">div3</div>
5 <script>
6     $('#btn').click(function(){
7         $("#div1").fadeTo("slow",0.15);
8         $("#div2").fadeTo("slow",0.4);
9         $("#div3").fadeTo("slow",0.7);
10    })
11 </script>
```

## jQuery 效果 - 滑动

### slideDown() 方法 向下滑动元素

语法

```
1 $(selector).slideDown(speed,callback);
```

可选的 `speed` 参数规定效果的时长。它可以取以下值: `"slow"`、`"fast"` 或 `毫秒`。

可选的 `callback` 参数是滑动完成后所执行的函数名称。

```
1 <button id="btn">集成按钮，可以显示也可以隐藏</button>
2 <div class="a" style="display:none;">
3     123
4 </div>
5 <script>
6     $('#btn').click(function(){
7         $('.a').slideDown();
8     })
9 </script>
```

## slideUp() 方法 用于向上滑动元素

语法

```
1 $(selector).slideUp(speed,callback);
```

可选的 `speed` 参数规定效果的时长。它可以取以下值: `"slow"`、`"fast"` 或 `毫秒`。

可选的 `callback` 参数是滑动完成后所执行的函数名称。

下面的例子演示了 `slideUp()` 方法:

```
1 <button id="btn">集成按钮，可以显示也可以隐藏</button>
2 <div class="a">
3     123
4 </div>
5 <script>
6     $('#btn').click(function(){
7         $('.a').slideUp();
8     })
9 </script>
```

## slideToggle() 方法

可以在 `slideDown()` 与 `slideUp()` 方法之间进行切换。

如果元素向下滑动, 则 `slideToggle()` 可向上滑动它们。

如果元素向上滑动, 则 `slideToggle()` 可向下滑动它们。

语法

```
1 $(selector).slideToggle(speed,callback);
```

可选的 `speed` 参数规定效果的时长。它可以取以下值: `"slow"`、`"fast"` 或 `毫秒`。

可选的 `callback` 参数是滑动完成后所执行的函数名称。

下面的例子演示了 `slideToggle()` 方法：

```
1 <button id="btn">集成按钮，可以显示也可以隐藏</button>
2 <div class="a">
3     123
4 </div>
5 <script>
6     $('#btn').click(function(){
7         $('.a').slideToggle();
8     })
9 </script>
```

## jQuery 效果- 动画

### animate() 方法 创建自定义动画

语法

```
1 $(selector).animate({params}, speed, callback);
```

必需的 `params` 参数定义形成动画的 `css` 属性。

可选的 `speed` 参数规定效果的时长。它可以取以下值："`slow`"、"`fast`" 或毫秒。

可选的 `callback` 参数是动画完成后所执行的函数名称。

下面的例子演示 `animate()` 方法的简单应用。它把 `<div>` 元素往右边移动了 250 像素：

```
1 <button id="btn">集成按钮，可以显示也可以隐藏</button>
2 <div class="a" style="width:200px;position: relative;">
3     123
4 </div>
5 <script>
6     $('#btn').click(function(){
7         $('.a').animate({left:'200px'})
8     })
9 </script>
```

备注：默认情况下，所有的 HTML 元素有一个静态的位置，且是不可移动的。如果需要改变为，我们需要将元素的 `position` 属性设置为 `relative`，`fixed`，或 `absolute`

### animate() - 操作多个属性

生成动画的过程中可同时使用多个属性

```
1 <button id="btn">集成按钮，可以显示也可以隐藏</button>
2 <div class="a" style="width:200px;position: relative;">
3     123
4 </div>
5 <script>
6     $('#btn').click(function(){
7         $('.a').animate({
```

```

8         left: '250px',
9         opacity: '0.5',
10        height: '150px',
11        width: '150px',
12        paddingLeft: '10px'
13    })
14    })
15    </script>

```

## animate() - 使用相对值

可以定义相对值（该值相对于元素的当前值）。需要在值的前面加上 += 或 -=

```

1    <button id="btn">集成按钮，可以显示也可以隐藏</button>
2    <div class="a" style="width:200px;position: relative;">
3        123
4    </div>
5    <script>
6        $('#btn').click(function(){
7            $('.a').animate({
8                left: '250px',
9                opacity: '0.5',
10               height: '+=150px',
11               width: '+=150px',
12               paddingLeft: '10px'
13            })
14        })
15    </script>

```

## animate() - 使用预定义的值

可以把属性的动画值设置为 "show"、"hide" 或 "toggle"

```

1    <button id="btn">集成按钮，可以显示也可以隐藏</button>
2    <div class="a" style="width:200px;position: relative;">
3        123
4    </div>
5    <script>
6        $('#btn').click(function(){
7            $('.a').animate({
8                height: 'toggle'
9            })
10        })
11    </script>

```

## animate() - 使用队列功能

jQuery 提供针对动画的队列功能，这意味着如果您在彼此之后编写多个 `animate()` 调用，jQuery 会创建包含这些方法调用的"内部"队列。然后逐一运行这些 `animate` 调用。

```
1      <button id="btn">集成按钮，可以显示也可以隐藏</button>
2      <div class="a" style="width:200px;position: relative;">
3          123
4      </div>
5      <script>
6          $('#btn').click(function(){
7              var div = $('div');
8              div.animate({height: '300px',opacity: '0.4'}, 'slow');
9              div.animate({width: '300px',opacity: '0.8'}, 'slow');
10             div.animate({height: '100px',opacity: '0.4'}, "slow");
11             div.animate({width: '100px',opacity: '0.8'}, "slow");
12         })
13     </script>
```

## jQuery 停止动画

### stop() 方法 用于停止动画或效果，在它们完成之前

jQuery `stop()` 方法用于停止动画或效果，在它们完成之前。

`stop()` 方法适用于所有 jQuery 效果函数，包括滑动、淡入淡出和自定义动画。

语法

```
1 | $(selector).stop(stopAll,goToEnd);
```