



jQuery效果

管理科学与工程学科
耿方方



主要内容

- 显示与隐藏
- 滑动
- 淡入淡出
- 自定义动画
- 动画效果综述

- jQuery中众多的动画与特效方法为提高页面的用户体验带来了极大的方便，通过少量几行代码，就可以实现元素的飞动、淡入淡出等动画效果，还可以自定义各种动画效果。

显示与隐藏

- 在页面中，元素的显示与隐藏是使用最频繁的操作，在传统的JavaScript中，一般通过改变元素显示的方式实现：
- `document.getElementById("p1").style.display=" block" ;`
- 隐藏：
- `document.getElementById("p1").style.display=" none" ;`
- 在jQuery中实现显示与隐藏的方式更为简单：
- `show ()` 与 `hide ()`
- 上面的代码就等同于：
- `$("p1").css("display" , " block");`
- `$("p1").css("display" , " none");`

显示与隐藏

- `show()` 与 `hide()` 方法，不仅可以实现“无动画”版的显示与隐藏，还可以完成动画效果的显示与隐藏，语法格式如下：
- `show(speed, [callback]);` //动画效果的显示功能
- `hide(speed, [callback]);` //动画效果的隐藏功能
- 其中 `speed` 表示执行动画时的速度，`callback` 表示动画完成时执行的回调函数。

显示与隐藏

- 案例1:

- `<script>`
- `$(function() {`
- `var $link=$(".artList a");`
- `var $hide=$(".artList :eq(2)");`
- `$link.click(function() {`
- `if($(this).html()=="显示"){`
- `$(this).html("隐藏");`
- `$hide.show();`
- `}`
- `else{`
- `$(this).html("显示");`
- `$hide.hide();`
- `}`
- `})`
- `</script>`

显示与隐藏

- 案例1-1:

- `<script>`

- `$(function() {`
- `$("a").click(function() {`
- `$("a").html("隐藏");`
- `$("img").show(3000, function() {`
- `$(this).css("border", "solid 1px #ccc");`
- `});`
- `});`
- `$("img").click(function() {`
- `$(this).hide(3000);`
- `$("a").html("显示");`
- `});`
- `});`
- `</script>`

显示与隐藏

- 切换元素的可见状态
- 使用toggle () 方法可以切换元素的可见状态，即如果元素是可见的，切换为隐藏；如果元素是隐藏的，切换为可见。toggle () 方法的语法格式如下：
 - toggle ()
 - 动画格式调用：
 - Toggle(speed, [callback])

显示与隐藏

■ 案例2:

- `<script>`
- `$(function() {`
- `$("#button").click(function() {`
- `$("#p").toggle();`
- `});`
- `});`
- `</script>`
- `</head>`
- `<body>`
- `<p>This is a paragraph.</p>`
- `<button class="btn1">Toggle</button>`
- `</body>`

滑动

- 在jQuery中，还有一种滑动的动画效果改变元素的高度，要实现元素的滑动效果，需要调用jQuery中的两个方法，一个是slideDown()，另一个是slideUp()，其语法结构如下所示：
- `slideDown(speed, [callback])`
- 其功能是以动画的效果将元素的高度向下增大，使其呈现一种“滑动”效果，而元素的其他属性并不发生变化；参数speed为动画显示的速度，可选项callback为动画显示完成后，执行的回调函数。
- `slideUp(speed, [callback])`
- 其功能是以动画的效果将所选择元素的高度向上减小，其他与slideDown () 方法一样。

滑动

■ 案例3:

- `<div class="divFrame">`
- `<div class="divTitle">隐藏图片</div>`
- `<div class="divContent">`
- ``
- `<div id="divTip"></div>`
- `</div>`
- `</div>`

```
<script>
    $(function(){
        var blnShow=false;
        var $Title=$(".divTitle");
        var $Tip=$("#divTip");
        $Title.click(function(){
            if(!blnShow){
                $("img").slideUp(3000,function(){
                    $Tip.html("关闭成功!");
                });
                $(this).html("显示图片");
                blnShow=true;
            }
            else{
                $Tip.html("");
                $("img").slideDown(3000);
                $(this).html("隐藏图片!");
                blnShow=false;
            }
        })
    })
</script>
```

滑动

- 上面的例子，为了判断当前的状态首先定义了一个**blnShow**变量，根据变量的值来判断执行哪个方法。在jQuery中无需定义变量，可以通过**slideToggle ()**方法，该方法可以根据当前元素的显示状态自动进行切换。语法：
- `slideToggle (speed, [callback])`

滑动

- 案例4:

- `<script>`
- `$(function() {`
- `$(".divFrame").click(function() {`
- `$("img").slideToggle(3000, function() {`
- `$("img").css("border", "1px solid #ccc");`
- `});`
- `});`
- `});`
- `</script>`
- `</head>`
- `<body>`
- `<div class="divFrame">`
- ``
- `</div>`

滑动

■ 案例5：伸缩菜单

```
■ <script>
■     $(document).ready(function() {
■         $("dd").hide(); //隐藏全部子菜单
■         $("dt[class!=' title']").toggle(function() {
■             // slideDown:通过高度变化（向下增长）来动态地显示所有匹配的元素
■             $(this).css("backgroundImage", "url(Images/13-title_hide.gif)"); //改变主菜单的背景
■             $(this).next().slideDown("slow");
■         },
■         function() {
■             // slideUp:通过高度变化（向上缩小）来动态地隐藏所有匹配的元素
■             $(this).css("backgroundImage", "url(images/title_show.gif)"); //改变主菜单的背景
■             $(this).next().slideUp("slow");
■         }
■     });
■ </script>
```

淡入淡出

- 在jQuery中我们可以通过实现元素渐渐变换背景色的动画效果来显示或隐藏元素，也就是所谓的淡入淡出效果。
- `show()`、`hide()`方法与`fadeIn()`、`fadeOut()`方法相比较，相同之处是都切换元素的显示状态，不同之处在于，前者的动画效果使元素的宽高属性都发生了变化，而后者仅仅是改变了元素的透明度，并不修改其它的属性。`fadeIn()`与`fadeOut()`方法语法结构如下所示：
- `fadeIn(speed, [callback])`
- `fadeOut(speed, [callback])`

淡入淡出

■ 案例6:

■ `<script>`

■ `$(function() {`

■ `$img=$("#img");`

■ `$tip=$("#divTip");`

■ `$("#input:eq(0)").click(function() {`

■ `$tip.html("");`

■ `$img.fadeIn(3000,function() {`

■ `$tip.html("淡入成功!");`

■ `});`

■ `});`

■ `</script>`

```
$("#input:eq(1)").click(function(){
    $tip.html("");
    $img.fadeOut(3000,function(){
        $tip.html("淡出成功!");
    });
});
});
```


淡入淡出

- fadeTo ()
- 在jQuery中，fadeIn () 和fadeOut () 方法通过动画效果，改变元素的透明度切换元素显示状态，其透明度从0.0到1.0淡出或从1.0到0.0淡入，从而实现淡入淡出的动画效果；如果透明度要指定到某一个值，则需要调用fadeTo () 方法，其调用语法格式为：
- fadeTo (speed, opacity, [callback])

淡入淡出

■ 案例7:

- `<script>`
- `$(function() {`
- `var $img=$("#img");`
- `var $sel=$("#select");`
- `$sel.change(function() {`
- `var fltValue=$sel.val();`
- `$img.fadeTo(3000,fltValue);`
- `})`
- `})`
- `</script> </script>`

自定义动画

- 前面介绍的动画效果都是元素局部属性发生变化，如高度、宽度、可见性等。在jQuery中，也允许用户自定义动画效果，通过使用`animate()`方法，可以制作出更复杂、更好的动画效果。
- `animate()`方法给Web前端开发者自定义各种复杂、高级的动画提供了极大的方便和空间，其语法格式如下所示：
- `animate(params, [duration], [easing], [callback])`
- `params`表示用于制作动画效果的属性样式和值的集合。可选项`duration`表示三种默认的速度字符“slow”、“normal”、“fast”或自定义的数字。`easing`表示控制动画的表现效果，通常有“linear”、“swing”字符值。

自定义动画

■ 案例8：简单动画

```
■ <script>
■     $(function() {
■         $(".divFrame").click(function() {
■             $(this).animate({
■                 width:"150px",
■                 height:"70px"
■             },3000,function() {
■                 $(this).css("border","3px solid #666")
■                     .html("变大了!");
■             }
■         );
■     });
■ });
■ </script>
■ </head>
■ <body>
■     <div class="divFrame">点击变大</div>
■ </body>
```

自定义动画

■ 案例9：移动动画

- `<script>`
- `$(function() {`
- `$("input:eq(0)").click(function() {`
- `$(".divList").animate({left:"-=52px"}, 3000);`
- `});`
- `$("input:eq(1)").click(function() {`
- `$(".divList").animate({left:"+=52px"}, 3000);`
- `})`
- `})`
- `</script>`

自定义动画

- 队列动画
- 所谓“队列”动画，是指在元素中执行一个以上的多个动画效果，即有多个animate () 方法在元素中执行，因此，根据这些animate () 方法执行的先后顺序，形成了动画“队列”，产生队列后，动画的效果便按“队列”的顺序进行展示。

自定义动画

■ 案例10：队列动画

- `<script>`
- `$(function() {`
- `$("div").click(function() {`
- `$(this).animate({height:100},"slow")`
- `.animate({width:100},"slow")`
- `.animate({height:50},"slow")`
- `.animate({width:50},"slow");`
- `});`
- `});`
- `</script>`
- `</head>`
- `<body>`
- `<div>队列</div>`
- `</body>`

自定义动画

- 动画停止和延时
- 在jQuery中，通过animate () 可以实现元素的动画显示，但在显示的过程中，必然要考虑各种实现因素和限制性条件的存在，因此，在执行过程中，可通过stop () 方法停止或delay () 方法延时某个动作的执行。stop () 与delay () 方法的语法调用格式介绍如下：
 - stop ([clearQueue], [gotoEnd])
 - clearQueue是一个布尔值，表示是否停止正在执行的动画，另外一个参数gotoEnd也是布尔值，表示是否立即完成正在执行的动画。
 - delay (duration, [queueName])
 - 表示设置一个延时值来推迟后续队列中动画的执行，其中参数duration为延时的时间值，单位是毫秒。queueName表示队列名词，即动画队列。

自定义动画

■ 案例11：动画的停止和延时

- `<script>`
- `$(function() {`
- `$("a:eq(0)").click(function() {`
- `$("img").slideToggle(3000);`
- `});`
- `$("a:eq(1)").click(function() {`
- `$("img").stop();`
- `});`
- `$("a:eq(2)").click(function() {`
- `$("img").delay(2000).slideToggle(3000);`
- `});`
- `});`
- `</script>`

动画效果综述

- 在jQuery中，虽然有很多动画方法，实现各种各样的动画效果，但综合起来，绝大部分的动画方法仅仅是改变元素的属性或样式，如高度、宽度、透明度和CSS样式属性。
- show () 和hide () 方法，元素以动画效果实现显示与隐藏，可以同时改变元素的多个属性，如高度、宽度、透明度。
- fadeIn () 和fadeOut () 方法，元素以动画的效果淡入与淡出，仅改变元素的透明度。
- slideup () 和slidedown () 方法，元素以“卷窗帘”的动画效果显示与隐藏，仅改变元素的高度，其他属性不发生变化。

动画效果综述

- `fadeTo ()` 方法，元素按指定的透明度进行渐进式调整，从而达到一种动画效果，该方法改变的是元素的透明度，高度和宽度都不发生变化。
- `toggle ()` 方法，可以代替`show ()`与`hide ()`两个方法，因此，其改变元素的属性与`show ()`和`hide ()`方法一样。
- `slideToggle ()` 方法，可以代替`slideUp ()`与`slideDown ()`两个方法，改变的元素也与他们一样。
- `animate ()` 方法，自定义元素的动画效果，可以实现上述6种方法中全部属性改变的功能。

动画效果综述

- `animate()` 方法不仅可以使元素实现各种各样的动画效果，还可以代替其他的动画方法，详细代码如下：
 - (1) `animate()` 方法代替 `hide()` 方法，代码如下：
 - `$(“页面元素”).animate({height:” hide”,width:” hide”,opacity:” hide” },600);` 等价于 `$(“页面元素”).hide(600);`
 - (2) `animate()` 方法代替 `fadeOut()` 方法，代码如下：
 - `$(“页面元素”).animate({opacity:” hide” },600);` 等价于 `$(“页面元素”).fadeOut(600);`
 - (3) `animate()` 方法代替 `slideUp()` 方法，代码如下：
 - `$(“页面元素”).animate({height:” hide” },600);` 等价于 `$(“页面元素”).slideUp(600);`

动画效果综述

- `animate()` 方法不仅可以使元素实现各种各样的动画效果，还可以代替其他的动画方法，详细代码如下：
- (4) `animate()` 方法代替 `fadeTo()` 方法，代码如下：
- `$(“页面元素”).animate({opacity:” 0.8” }, 600);` 等价于 `$(“页面元素”).fadeTo(600, ” 0.8”);`

想一想

- 1、