

第十课 jquery 动画效果

学习目录

- 淡入淡出效果
- 滑动效果
- 自定义动画效果

一 . 淡入淡出效果

1.fadeIn() 用于逐渐淡入被隐藏的元素，淡入成功之后还可触发一个回调函数。

```
$("#btn1").click(function(){  
  
    $("#div1").fadeIn();  
  
    $("#div2").fadeIn("slow");  
  
    $("#div3").fadeIn(1000);  
  
});  
  
$(".div1").fadeIn("fast","swing",function(){  
  
    console.log("淡入动画成功");  
  
});
```

2.fadeOut() 方法用于逐渐隐藏可见的元素，淡出成功之后也可以触发一个回调函数。

```
$("#btn1").click(function(){  
  
    $("#div1").fadeOut();  
  
});
```

3.fadeToggle() 方法可以在 fadeIn() 与 fadeOut() 方法之间进行切换，也就是说当元素隐藏时 fadeToggle 方法让元素慢慢显示出来，当元素显示时 fadeToggle 方法让元素慢慢隐藏起来。

```
$("#btn1").click(function(){  
  
    $("#div1").fadeToggle("slow","linear");  
  
});
```

4.fadeTo() 方法让元素渐变为指定的不透明度，这个不透明度的数值是 0-1。

```
$("#btn1").click(function(){  
  
    $("#div1").fadeTo("normal", 0.36);  
  
});
```

二 . 滑动效果

1.slideDown() 方法用来指定向下滑动从而显示出元素内容，在元素显示成功之后也可以触发一个回调函数。

```
$(".btn1").click(function(){  
  
    $("#div").slideDown();  
  
});  
  
$("#div1").slideDown("slow","swing",function(){  
  
    console.log("向下滑动动画成功");  
  
});
```

2.slideUp() 方法用来指定向上滑动从而隐藏元素，元素隐藏之后也可以触发一个回调函数。

```
$("#btn1").click(function(){  
  
    $("#div1").slideUp();  
  
});
```

3.slideToggle() 方法让元素在 slideDown() 与 slideUp() 方法之间进行切换，元素显示时让元素逐渐向上收起，元素隐藏时让元素逐渐向下显示。

```
$("#btn1").click(function(){  
  
    $("#div1").slideToggle();  
  
});
```

三 . 自定义动画效果

1.animate() 方法用来构建自定义动画。

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

params:定义形成一组动画的 CSS 属性及其属性值的集合。

speed:预定速度字符串("slow","normal", or "fast")或者以毫秒数值来表示的动画时长(如：2000 表示动画时间需要 2 秒)。

callback:元素动画完成之后要触发的函数。

```
$(".btn1").on('click',function){  
  
    $(".div1").animate({left:'100px'});  
  
});
```

```
$(".btn1").click(function(){  
  
    $(".div1").animate({  
  
        width: "100px",  
  
        height: "100px",  
  
        background: "orange",  
  
        fontSize:"30px",  
  
        fontWeight: "bold"  
  
    }, 3000);  
  
});
```

animate() 方法几乎可以来操作所有 CSS 属性,不过当使用 animate() 时,属性名必须使用 Camel 标记法来书写,比如,font-size 要写成 fontSize 等等。

jQuery 提供的 animate() 方法支持针对动画的队列功能,这些方法会形成一个调用的内部动画队列,这个队列用到了 Queue 队列的接口,这个队列接口是内部专门为动画服务的,属于比较基础的方法,跟排队买早饭的道理差不多,一个队列中的每个人都会逐一排队在早餐店买早餐,当动画形成队列之后会逐一运行这些 animate 调用。

```
$(".btn1").click(function(){  
  
    $(".div1").animate({background:'orange',opacity:'0.3'});  
  
    $(".div1").animate({background:blue,opacity:'0.8'});  
  
    $(".div1").animate({background:'orange',opacity:'1'});  
  
});
```

2.delay() 方法用来设置一个延时来推迟队列动画的执行。

```
$('.div1').fadeIn(300).delay(1000).animate({'background':'orange'});
```

3.stop() 方法用于停止动画或效果。

```
$(".btn1").click(function(){  
  
    $(".div1").animate({left: '+500px'}, 6000);  
  
    $(".div1").fadeOut(5000);  
  
});
```

```
$(".btn2").click(function(){  
  
    $(".div1").stop();  
  
});
```

stop([clearQueue],[jumpToEnd])

ps : stop() 方法只停止当前正在进行的动画，停止当前动画后，队列中的下一个动画会继续进行。可选的 clearQueue 参数规定是否需要清除动画队列。默认是 false，因此传递 true 的情况下默认会清除动画队列里的所有动画，第三个参数 jumpToEnd 表示是否立即完成当前动画，true 表示立即完成当前动画，默认是 false 不会立即完成当前动画。

谢谢观看！

我是星星课堂老师：周小周