

## 一、显示和隐藏

hide():隐藏 在HTML文档中，是把一个元素的css属性display设置为none

show():显示。把隐藏的元素显示出来。把css的display属性设置为block或inline或者去了none之外的其他值。到底是什么状况，要看隐藏之前的状态。隐藏前是什么状态就显示为什么状态

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
    <style type="text/css">
      * {
        margin: 0px;
        padding: 0px;
      }
      .box {
        width: 300px;
        height: 300px;
        background-color: pink;
        border: 1px dashed blue;
      }
    </style>
    <script src="libs/jquery-3.1.1.js" type="text/javascript" charset="utf-8"></script>
    <script type="text/javascript">
      $(function() {
        var $box = $(".box");
        $("button:eq(0)").click(function() {
          //显示元素。
          $box.show();
        })
        $("button:eq(1)").click(function() {
          //隐藏元素
          $box.hide();
        })
      })
    </script>
  </head>
  <body>
    <div>
      <button>显示</button>
      <button>隐藏</button>
    </div>
    <div class="box"></div>
  </body>
</html>

```

show和hide方法也可以传入表示时间的参数。意思是说，从显示到隐藏，或隐藏到显示经过的事件。在这个过程中，元素的透明度会变大或变小，尺寸会变小或变大。

show(时间)、hide(时间)

jQuery还提供了3个值：slow(600ms), nomal(400ms), fast(200ms)

```

$box.show("slow");
$box.show(2000); // 1000ms
$box.hide("slow");
$box.hide(2000);

```

## 二、淡入和淡出

---

`fadeIn()`和`fadeOut()`淡入和淡出。只改变透明度，不改变尺寸。

参数和`show`、`hide`一致

```
$box.fadeOut(2000); //淡入  
$box.fadeIn(2000); //淡出
```

## 三、展开和收起

---

`slideUp()`和`slideDown()`。收起和展开。通过改变元素的尺寸

```
$box.slideUp(1000); //收起  
$box.slideDown(1000); //展开
```

## 四、动画其他方法和属性

---

```

<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
    <style type="text/css">
      * {
        margin: 0px;
        padding: 0px;
      }
      .box {
        width: 300px;
        height: 300px;
        background-color: pink;
        border: 1px dashed blue;
      }
    </style>
    <script src="libs/jquery-3.1.1.js" type="text/javascript" charset="utf-8"></script>
    <script type="text/javascript">
      $(function() {
        var $box = $(".box");
        $("button:eq(0)").click(function() {

          //toggle()专门用于切换元素的状态，如果是隐藏的就显示，如果是显示的就隐藏。可以替
          换hide和show

          $box.toggle(2000);
        })
        $("button:eq(1)").click(function() {
          //用于替换fadeIn和fadeOut
          $box.fadeToggle(2000);
        })
        $("button:eq(2)").click(function() {
          //用于替换slideUp和slideDown
          $box.slideToggle(2000);
        })
        $("button:eq(3)").click(function() {
          //改变透明度到指定透明度值。值改变透明度
          $box.fadeTo(2000, 0.3)
        })
      })
    </script>
  </head>
  <body>
    <div>
      <button>显示和隐藏</button>
      <button>淡入或淡出</button>
      <button>展开和收缩</button>
      <button>改变透明度到指定的值</button>
    </div>
    <div class="box"></div>
  </body>
</html>

```

## 五、自定义动画

---

仅仅上面的动画并不能满足我们的需求，如个想要更复杂的动画效果，就需要自定义动画。自定义动画只需要用到一个函数：`animate(params, speed, callback)`

参数1：一个包含样式的属性和值的映射对。可以有多个映射。 必须

参数2：可选。速度

参数3：回调函数。可选。表示动画执行完毕之后然后再执行的函数。

```

<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
    <style type="text/css">
      * {
        margin: 0px;
        padding: 0px;
      }
      .box {

        width: 300px;
        height: 300px;
        background-color: pink;
        border: 1px dashed blue;
        position: absolute;

      }
    </style>
    <script src="libs/jquery-3.1.1.js" type="text/javascript" charset="utf-8"></script>
    <script type="text/javascript">
      $(function() {
        var $box = $(".box");
        $(".button:eq(0)").click(function () {
          //自定义动画
          $box.animate({left:"800px"}, 2000);

        });
        $(".button:eq(1)").click(function () {
          //自定义动画
          $box.animate({top:"500px"}, 2000);
        });
        $(".button:eq(2)").click(function () {
          //构成了动画流，按照动画添加的顺序执行。
          $box
            .animate({left:"800px"}, 500)
            .animate({top:"500px"}, 2000)
            .fadeOut();
        });
        $(".button:eq(3)").click(function () {
          //同时执行多个属性的动画
          $box
            .animate({left:"800px", top:"500px"}, 5000)
        });
      })
    </script>
  </head>
  <body>
    <div>
      <button>向右</button>
      <button>向下</button>
      <button>先向右，再向下</button>
      <button>同时向右和向下</button>

    </div>
  </body>
</html>

```

```
<div class="box"></div>  
</body>  
</html>
```

## 六、判断动画是否在执行和停止动画

---

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
    <style type="text/css">
      * {
        margin: 0px;
        padding: 0px;
      }
      .box {
        width: 300px;
        height: 300px;
        background-color: pink;
        border: 1px dashed blue;
        position: absolute;
      }
    </style>
    <script src="libs/jquery-3.1.1.js" type="text/javascript" charset="utf-8"></script>
    <script type="text/javascript">
      $(function() {
        var $box = $(".box");
        $(".button:eq(0)").click(function () {
          $box.animate({left:"800px"}, 5000)
            .animate({top:"600px"}, 6000);
        });
        $(".button:eq(1)").click(function () {
          //判断动画是否在执行
          if($box.is(":animated")){
            alert("动画正在执行");
          }else{
            alert("动画没有在执行");
          }
        });
        $(".button:eq(2)").click(function () {
          //不传入任何参数：表示结束正在执行的动画，开始执行动画队列中的下一个动画。
          $box.stop();
          /*stop有两个可选参数：
            参数1: true/false true表示立即清除所有动画队列中的所有动画，并结束当前动画。
            已经执行到了什么
            地方就停在什么地方。 默认是false，表示停止当前动画，并开始动画队列
            中的下一个。
            参数2: true/false: 表示结束动画的时候，是否直接到达当前动画的末尾状态。true
            表示直接去当前动画的末尾状态，f
            false,表示维持当前状态

          */
          $box.stop(true, true);
        });
      });
    </script>
  </head>

```



```
<body>
  <div>
    <button>开始执行动画</button>
    <button>判断动画是否在执行</button>
    <button>停止动画</button>
  </div>
  <div class="box"></div>
</body>
</html>
```

## 七、AJAX

---

原生JavaScript使用AJAX相对比较繁琐，jQuery对AJAX做了封装，使用起来更简单。

对AJAX一共封装了6个方法，分成三层。

最底层：\$.ajax()

第二层：\$.load()、\$.get()、\$.post()

第三层：\$.getScript()、\$.getJSON()

第二层的方法使用最多最广泛

### 7.1 load方法

---

load(url, [data], [callback])

url:待装入 HTML 网页网址。

data:发送至服务器的 key/value 数据。在jQuery 1.3中也可以接受一个字符串了。可选

callback:载入成功时回调函数。可选

注意：默认使用 GET 方式 - 传递附加参数时自动转换为 POST 方式

#### 7.1.1 加载静态HTML页面

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
    <script src="libs/jquery-3.1.1.js" type="text/javascript" charset="utf-8"></script>
    <script type="text/javascript">
      $(function () {
        $("div button").click(function () {
          //加载指定的url资源，并直接把加载的内容插入到 .content元素中。如果.content中原来
          有内容，则
          //会覆盖原来的内容。 默认情况get请求。如果添加了第二个参数，则会自动使用post请求
          $("#content").load("html_data.html");
        })
      })
    </script>
  </head>
  <body>
    <div>
      <button>使用ajax加载静态页面到下面的div中</button>
    </div>
    <div id="content"></div>
  </body>
</html>

```

### 7.1.2 筛选添加的内容

加载的数据，我们也可以筛选出我们需要的数据，而不用全部填充到元素中。

只需要在url的后面使用空格隔开，添加我们合适的选择器

```

//内容加载完成后，只会把h1元素和id是#p的添加到#content中，其他的则不会
$("#content").load("html_data.html h1,#p");

```

### 7.1.3 回调函数

load方法的最后一个参数是回调函数，当请求完成后总会回调这个函数，不管请求是否成功。

```

$("#content").load("html_data.html h1,#p", function function_name (response,status,xhr) {
  //response - 包含来自请求的结果数据
  //status - 包含请求的状态 ("success", "notmodified", "error", "timeout" 或 "parsererror")
  //xhr - 包含 XMLHttpRequest 对象
  alert("加载完毕");
  console.log(response + "\n" + status + "\n" + xhr);
});

```

## 7.2 get方法和post方法

load一般用来读取静态html页面(动态也可以)。如果向方便传递数据到服务器，更多的使用get或post。

load方法是jQuery对象的方法，而get和post方法是jQuery中的全局函数，所以可以直接使用\$.get()和\$.post()来调用

### 7.3 getScript方法

---

### 7.4 getJson方法

---

### 7.5 ajax方法

---