

jQuery基础

学习目标：

1. 能够理解js框架的概念
2. 能够引入jQuery
3. 能够使用jQuery的基本选择器
4. 能够使用jQuery的层级选择器
5. 能够使用jQuery的属性选择器
6. 能够使用jQuery的基本过滤选择器
7. 能够使用jQuery的属性过滤选择器
8. 能够使用jQuery的DOM操作的方法
9. 能够完成隔行换色案例
10. 能够完成全选全不选案例

第1章 自定义js框架

1.1 js框架概述

框架（Framework）是整个或部分系统的可重用设计，表现为一组抽象构件及构件实例间交互的方法。框架是完成某种功能的半成品，框架是将我们日常的繁琐的代码进行抽取，并提供出更加简洁，功能更加强大的语法实现，我们可以对底层的API进行封装，对外提供更加简单的语法来实现最终的功能，从而提高开发效率。

1.2 代码实现

html代码：

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
<script type="text/javascript" src="jquery-my.js"></script>

</head>
<body>
  <div id="myDiv"></div>
  <script type="text/javascript">
    alert($("#myDiv"));
  </script>
</body>
</html>
```

jquery-my.js文件：

```
function jQuery(selector){  
    var id = selector.substring(1);  
    var element = document.getElementById(id);  
    return element;  
}  
  
$ = jQuery;
```

第2章 jQuery框架介绍

2.1 jQuery框架概述

2.1.1 jQuery的概述

jQuery是一个优秀的javascript的轻量级框架之一，兼容css3和各大浏览器，提供了dom、events、animate、ajax等简易的操作。并且jquery的插件非常丰富，大多数功能都有相应的插件解决方案。jquery的宗旨是write less do more。

2.1.2 jQuery的作用

jQuery最主要的作用是简化js的Dom树的操作

2.2 jQuery框架的下载与引入

2.2.1 jQuery下载位置

- jQuery的官方下载地址：<http://www.jquery.com>

2.2.2 jQuery的版本

- 1.x：兼容IE678，使用最为广泛的，官方只做BUG维护，功能不再新增。因此一般项目来说，使用1.x版本就可以了，最终版本：1.12.4 (2016年5月20日)
- 2.x：不兼容IE678，很少有人使用，官方只做BUG维护，功能不再新增。如果不考虑兼容低版本的浏览器可以使用2.x，最终版本：2.2.4 (2016年5月20日)
- 3.x：不兼容IE678，只支持最新的浏览器。除非特殊要求，一般不会使用3.x版本的，很多老的jQuery插件不支持这个版本。目前该版本是官方主要更新维护的版本
- 开发版本与生产版本，命名为jQuery-x.x.x.js为开发版本，命名为jQuery-x.x.x.min.js为生产版本，开发版本源码格式良好，有代码缩进和代码注释，方便开发人员查看源码，但体积稍大。而生产版本没有代码缩进和注释，且去掉了换行和空行，不方便发人员查看源码，但体积很小。

2.2.3 jQuery的引入

在使用jQuery的html中使用js的引入方式进行引入，如下：

```
<script type="text/javascript" src="jquery-x.x.x.js">
```

2.2.4 jQuery引入成功的测试

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
<!-- 引入jQuery -->
<script type="text/javascript" src="../../js/jquery-3.3.1.js"></script>
<!-- 测试jQuery -->
<script type="text/javascript">
    $(function(){
        alert("jQuery引入成功...");
    });
</script>
</head>
<body>

</body>
</html>
```

第3章 jQuery对象与js对象之间的转换

jQuery本质上虽然也是js，但如果使用jQuery的属性和方法那么必须保证对象是jQuery对象而不是js方式获得的DOM对象。使用js方式获取的对象是js的DOM对象，使用jQuery方式获取的对象是jQuery对象。两者的转换关系如下：

- js的DOM对象转换成jQuery对象，语法：\$(js对象)
- jQuery对象转换成js对象，语法：jquery对象[索引] 或 jquery对象.get(索引)

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
<!-- 引入jQuery -->
<script type="text/javascript" src="../../js/jquery-3.3.1.js"></script>
<!-- 测试jQuery -->
<script type="text/javascript">
    $(function(){

        //通过js方式打印div内部的内容
        var divEle = document.getElementById("myDiv");//js的DOM对象
        alert(divEle.innerHTML);//js对象的innerHTML属性

        //通过jQuery方式打印div内部的内容
        var $divEle = $("#myDiv");//jQuery对象

        alert($divEle.html());//jQuery对象的html方法
    });
</script>
</body>
</html>
```

```
//使用js的DOM对象调用jQuery对象的方法
alert(divEle.html());//错误写法，不能正常执行

//使用jQuery对象调用js的DOM对象的属性
alert($divEle.innerHTML);//错误写法，不能正常执行

//js转换成jQuery对象后在调用jQuery对象的html方法
alert($(divEle).html());//转换后正常执行

//jQuery转换成js的DOM对象后在调用js的innerHTML属性
alert($divEle[0].innerHTML);

});
</script>
</head>
<body>
    <div id="myDiv">通过不同方式获得文本内容</div>
</body>
</html>
```

第4章 jQuery选择器

4.1 基本选择器

4.1.1 语法

选择器名称	语法	解释
标签选择器（元素选择器）	\$("html标签名")	获得所有匹配标签名称的于元素
id选择器	\$("#id的属性值")	获得与指定id属性值匹配的元素
类选择器	\$(".class的属性值")	获得与指定的class属性值匹配的元素

4.1.2 代码

```
//<input type="button" value="改变 id 为 one 的元素的背景色为 红色" id="b1"/>
$("#b1").click(function(){
    $("#one").css("backgroundColor","red");
});

//<input type="button" value=" 改变元素名为 <div> 的所有元素的背景色为 红色" id="b2"/>
$("#b2").click(function(){
    $("div").css("backgroundColor","red");
});
```

```
//<input type="button" value=" 改变 class 为 mini 的所有元素的背景色为 红色" id="b3"/>
$("#b3").click(function(){
    $(".mini").css("backgroundColor","red");
});

//<input type="button" value=" 改变所有的<span>元素和 id 为 two 的元素的背景色为红色"
id="b4"/>
$("#b4").click(function(){
    $("span,#two").css("backgroundColor","red");
});
```

4.2 层级选择器

4.2.1 语法

选择器名称	语法	解释
后代选择器	\$("A B")	选择A元素内部的所有B元素
子选择器	\$("A > B")	选择A元素内部的所有B子元素

4.2.2 代码

```
//<input type="button" value=" 改变 <body> 内所有 <div> 的背景色为红色" id="b1"/>
$("#b1").click(function(){
    $("body div").css("background-color","red");
});

//<input type="button" value=" 改变 <body> 内 子 <div> 的背景色为 红色" id="b2"/>
$("#b2").click(function(){
    $("body>div").css("background-color","red");
});
```

4.3 属性选择器

4.3.1 语法

选择器名称	语法	解释
属性选择器	\$("A[属性名]")	包含指定属性的选择器
属性选择器	\$("A[属性名=值]")	包含指定属性等于指定值的选择器
复合属性选择器	\$("A[属性名=值]... ..")	包含多个属性条件的选择器

4.3.2 代码

```
//<input type="button" value=" 含有属性title 的div元素背景色为红色" id="b1"/>
$("#b1").click(function(){
    $("div[title]").css("background-color","red");
});

//<input type="button" value=" 属性title值等于test的div元素背景色为红色" id="b2"/>
$("#b2").click(function(){
    $("div[title='test']").css("background-color","red");
});

//<input type="button" value="选取有属性id的div元素，然后在结果中选取属性title等于“test”的 div
元素背景色为红色" id="b3"/>
$("#b3").click(function(){
    $("div[id][title='test']").css("background-color","red");
});
```

4.4 基本过滤选择器

4.4.1 语法

选择器名称	语法	解释
首元素选择器	:first	获得选择的元素中的第一个元素
尾元素选择器	:last	获得选择的元素中的最后一个元素
非元素选择器	:not(selector)	不包括指定内容的元素
偶数选择器	:even	偶数，从 0 开始计数
奇数选择器	:odd	奇数，从 0 开始计数
等于索引选择器	:eq(index)	指定索引元素
大于索引选择器	:gt(index)	大于指定索引元素
小于索引选择器	:lt(index)	小于指定索引元素
标题选择器	:header	获得标题元素，固定写法

4.4.2 代码

```
//<input type="button" value=" 改变第一个 div 元素的背景色为 红色" id="b1"/>
$("#b1").click(function(){
    $("div:first").css("background-color","red");
});
```



```
//<input type="button" value=" 改变最后一个 div 元素的背景色为 红色" id="b2"/>
$("#b2").click(function(){
    $("div:last").css("background-color","red");
});

//<input type="button" value=" 改变class不为 one 的所有 div 元素的背景色为 红色" id="b3"/>
$("#b3").click(function(){
    $("div:not(.one)").css("background-color","red");
});

//<input type="button" value=" 改变索引值为偶数的 div 元素的背景色为 红色" id="b4"/>
$("#b4").click(function(){
    $("div:even").css("background-color","red");
});

//<input type="button" value=" 改变索引值为奇数的 div 元素的背景色为 红色" id="b5"/>
$("#b5").click(function(){
    $("div:odd").css("background-color","red");
});

//<input type="button" value=" 改变索引值为大于 3 的 div 元素的背景色为 红色" id="b6"/>
$("#b6").click(function(){
    $("div:gt(3)").css("background-color","red");
});

//<input type="button" value=" 改变索引值为等于 3 的 div 元素的背景色为 红色" id="b7"/>
$("#b7").click(function(){
    $("div:eq(3)").css("background-color","red");
});

//<input type="button" value=" 改变索引值为小于 3 的 div 元素的背景色为 红色" id="b8"/>
$("#b8").click(function(){
    $("div:lt(3)").css("background-color","red");
});

//<input type="button" value=" 改变所有的标题元素的背景色为 红色" id="b9"/>
$("#b9").click(function(){
    $("*:header").css("background-color","red");
});
```

4.5 表单属性选择器

4.5.1 语法

选择器名称	语法	解释
可用元素选择器	:enabled	获得可用元素
不可用元素选择器	:disabled	获得不可用元素
选中选择器	:checked	获得单选/复选框选中的元素
选中选择器	:selected	获得下拉框选中的元素

4.5.2 代码

```
//<input type="button" value=" 利用 jQuery 对象的 val() 方法改变表单内可用 <input> 元素的值" id="b1"/>
$("#b1").click(function(){
    var $inputs = $("input[type='text']:enabled");//$inputs内部有两个input的js的dom对象
    for(var i=0;i<$inputs.length;i++){
        alert($inputs[i].value);
    }
});

// <input type="button" value=" 利用 jQuery 对象的 val() 方法改变表单内不可用 <input> 元素的值" id="b2"/>
$("#b2").click(function(){
    var $inputs = $("input[type='text']:disabled");//$inputs内部有两个input的js的dom对象
    for(var i=0;i<$inputs.length;i++){
        alert($inputs[i].value);
    }
});

//<input type="button" value=" 利用 jQuery 对象的 length 属性获取多选框选中的个数" id="b3"/>
$("#b3").click(function(){
    var $checkedInput = $("input[type='checkbox']:checked");
    alert($checkedInput.length);
});

// <input type="button" value=" 利用 jQuery 对象的 text() 方法获取下拉框选中的内容" id="b4"/>
$("#b4").click(function(){
    var $options = $("option:selected");
    for(var i=0;i<$options.length;i++){
        alert($($options[i]).text());
    }
});
```

第5章 jQuery的DOM操作

5.1 jQuery对DOM树中的文本和值进行操作

5.5.1 语法

API方法	解释
val([value])	获得/设置元素value属性相应的值
text([value])	获得/设置元素的文本内容
html([value])	获得/设置元素的标签体内容

5.5.2 代码

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
    <script src="../js/jquery-3.3.1.js" type="text/javascript" charset="utf-8">
    </script>
    <script type="text/javascript">
      //页面加载完毕事件
      $(function(){
        //获取张三
        alert($("#myinput").val());
        //获得mydiv的文本内容
        alert($("#mydiv").text());
        //获得mydiv的标签体内容
        alert($("#mydiv").html());
      });
    </script>
  </head>
  <body>
    <input id="myinput" type="text" name="username" value="张三" /><br />
    <div id="mydiv"><p><a href="#">标题标签</a></p></div>
  </body>
</html>
```

5.2 jQuery对DOM树中的属性进行操作

5.2.1 语法

API方法	解释
attr(name[,value])	获得/设置属性的值
prop(name[,value])	获得/设置属性的值(checked, selected)

5.2.1 代码

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>获取属性</title>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8">
    <script type="text/javascript" src="../js/jquery-3.3.1.js"></script>
  </head>
  <body>
    <ul>
      <li id="bj" name="beijing" xxx="yyy" >北京</li>
      <li id="tj" name="tianjin">天津</li>
    </ul>
    <input type="checkbox" id="hobby"/>
  </body>

  <script type="text/javascript">
    //获取北京节点的name属性值
    alert($("#bj").attr("name"));

    //设置北京节点的name属性的值为dabeijing
    $("#bj").attr("name", "dabeijing");

    //新增北京节点的discription属性 属性值是didu
    $("#bj").attr("discription", "wudu");

    //删除北京节点的name属性并检验name属性是否存在
    $("#bj").removeAttr("name");

    //获得hobby的选中状态
    alert($("#hobby").prop("checked"));

  </script>
</html>
```

5.2.2 attr与prop的注意事项

- attr与prop是以1.6为界限
- checked 和 selected 使用prop获取
- 其他使用attr获取 获取不到换成prop

5.3 jQuery对class进行操作

5.3.1 语法

API方法	解释
css(name[,value])	获取/设置指定的CSS样式
addClass(value)	addClass(类样式名) 给指定的对象添加新的类样式，指定类样式名字即可
removeClass(value)	removeClass(类样式名) 删除指定的类样式
toggleClass(value)	toggleClass(类样式名) 切换样式，如果没有类样式，则添加，如果有类样式，则删除

5.3.2 代码

```
//<input type="button" value="采用属性增加样式(改变id=one的样式)" id="b1"/>
$("#b1").click(function(){
    $("#one").attr("class","second");

});
//<input type="button" value=" addClass" id="b2"/>
$("#b2").click(function(){
    $("#one").addClass("second");
});

//<input type="button" value="removeClass" id="b3"/>
$("#b3").click(function(){
    $("#one").removeClass("second");
});

//<input type="button" value=" 切换样式" id="b5"/>
$("#b4").click(function(){
    $("#one").toggleClass("second");
});

//<input type="button" value=" 通过css()获得id为one背景颜色" id="b5"/>
$("#b5").click(function(){
    alert($("#one").css("background-color"));
});

// <input type="button" value=" 通过css()设置id为one背景颜色为绿色" id="b6"/>
$("#b6").click(function(){
    $("#one").css("background-color","green");
});
```

5.4 jQuery创建插入对象

5.4.1 语法

API方法	解释
\$("#")	创建A元素对象
append(element)	添加成最后一个子元素，两者之间是父子关系
prepend(element)	添加成第一个子元素，两者之间是父子关系
before(element)	添加到当前元素的前面，两者之间是兄弟关系
after(element)	添加到当前元素的后面，两者之间是兄弟关系

5.4.2 代码

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>内部插入脚本</title>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8">
    <script src="../js/jquery-3.3.1.js"></script>
  </head>
  <body>
    <ul id="city">
      <li id="bj" name="beijing">北京</li>
      <li id="tj" name="tianjin">天津</li>
      <li id="cq" name="chongqing">重庆</li>
    </ul>
    <ul id="love">
      <li id="fk" name="fankong">反恐</li>
      <li id="xj" name="xingji">星际</li>
    </ul>
    <div id="foo1">Hello1</div>
  </body>
  <script type="text/javascript">

    /**将反恐放置到city的后面*/
    $("#city").append($("#fk"));

    /**将反恐放置到city的最前面*/
    $("#city").prepend($("#fk"));

    //将反恐插入到天津后面
    $("#tj").after($("#fk"));

    //将反恐插入到天津前面
    $("#tj").before($("#fk"));

    //将反恐插入到天津后面
    $("#fk").insertAfter($("#tj"));

    //将反恐插入到天津前面
```

```
$("#fk").insertBefore($("#tj"));

</script>
</html>
```

5.5 jQuery删除对象

5.5.1 语法

API方法	解释
remove()	删除指定元素
empty()	清空指定元素的所有子元素

5.5.2 代码

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>删除节点</title>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8">
    <script type="text/javascript" src="../js/jquery-3.3.1.js"></script>
  </head>

  <body>
    <ul id="city">
      <li id="bj" name="beijing">北京</li>
      <li id="tj" name="tianjin">天津</li>
      <li id="cq" name="chongqing">重庆</li>
    </ul>
    <p class="hello">Hello</p> how are <p>you?</p>
  </body>

  <script type="text/javascript">

    //删除<li id="bj" name="beijing">北京</li>
    $("#bj").remove();

    //删除所有的子节点  清空元素中的所有后代节点(不包含属性节点)。
    $("#city").empty();

  </script>
</html>
```

第6章 综合案例

6.1 案例-表格隔行换色



6.1.1 效果

删除				
<input type="checkbox"/>	分类ID	分类名称	分类描述	操作
<input type="checkbox"/>	1	手机数码	手机数码类商品	修改 删除
<input type="checkbox"/>	2	电脑办公	电脑办公类商品	修改 删除
<input type="checkbox"/>	3	鞋靴箱包	鞋靴箱包类商品	修改 删除
<input type="checkbox"/>	4	家居饰品	家居饰品类商品	修改 删除

6.1.2 代码

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
    <script src="../../js/jquery-3.3.1.js" type="text/javascript" charset="utf-8">
    </script>
    <script type="text/javascript">
      $(function(){
        $("tr:gt(1):even").css("background-color","pink");
        $("tr:gt(1):odd").css("background-color","yellow");
      });
    </script>
  </head>
  <body>
    <table id="tab1" border="1" width="800" align="center" >
      <tr>
        <td colspan="5"><input type="button" value="删除"></td>
      </tr>
      <tr style="background-color: #999999;">
        <th><input type="checkbox"></th>
        <th>分类ID</th>
        <th>分类名称</th>
        <th>分类描述</th>
        <th>操作</th>
      </tr>
      <tr>
        <td><input type="checkbox"></td>
        <td>1</td>
        <td>手机数码</td>
        <td>手机数码类商品</td>
        <td><a href="#">修改</a>|<a href="#">删除</a></td>
      </tr>
      <tr>
        <td><input type="checkbox"></td>
        <td>2</td>
        <td>电脑办公</td>
        <td>电脑办公类商品</td>
        <td><a href="#">修改</a>|<a href="#">删除</a></td>
      </tr>
```

```

        </tr>
        <tr>
            <td><input type="checkbox"></td>
            <td>3</td>
            <td>鞋靴箱包</td>
            <td>鞋靴箱包类商品</td>
            <td><a href="">修改</a>|<a href="">删除</a></td>
        </tr>
        <tr>
            <td><input type="checkbox"></td>
            <td>4</td>
            <td>家居饰品</td>
            <td>家居饰品类商品</td>
            <td><a href="">修改</a>|<a href="">删除</a></td>
        </tr>
    </table>
</body>
</html>
    
```

6.2 案例-复选框全选全不选

6.2.1 效果

反选				
<input checked="" type="checkbox"/>	分类ID	分类名称	分类描述	操作
<input checked="" type="checkbox"/>	1	手机数码	手机数码类商品	修改 删除
<input checked="" type="checkbox"/>	2	电脑办公	电脑办公类商品	修改 删除
<input checked="" type="checkbox"/>	3	鞋靴箱包	鞋靴箱包类商品	修改 删除
<input checked="" type="checkbox"/>	4	家居饰品	家居饰品类商品	修改 删除

6.2.2 代码

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
        <script src="../../js/jquery-3.3.1.js" type="text/javascript" charset="utf-8">
        </script>
        <script type="text/javascript">
            function selectAll(obj){
                //this是js对象
                //将下面的所有的复选框的状态编程总选框一致
                $("input[class='itemSelect']").prop("checked",$(obj)
                    .prop("checked"));
            }
            function revSelect(){
                //将jquery(数组)对象内部的每个元素都执行一次操作
                $("input[class='itemSelect']").click();
            }
        </script>
    </head>
    <body>
        <table>
            <tr>
                <td>反选</td>
                <td></td>
                <td></td>
                <td></td>
                <td></td>
            </tr>
            <tr>
                <td><input checked="" type="checkbox"></td>
                <td>分类ID</td>
                <td>分类名称</td>
                <td>分类描述</td>
                <td>操作</td>
            </tr>
            <tr>
                <td><input checked="" type="checkbox"></td>
                <td>1</td>
                <td>手机数码</td>
                <td>手机数码类商品</td>
                <td><a href="#">修改</a><a href="#">删除</a></td>
            </tr>
            <tr>
                <td><input checked="" type="checkbox"></td>
                <td>2</td>
                <td>电脑办公</td>
                <td>电脑办公类商品</td>
                <td><a href="#">修改</a><a href="#">删除</a></td>
            </tr>
            <tr>
                <td><input checked="" type="checkbox"></td>
                <td>3</td>
                <td>鞋靴箱包</td>
                <td>鞋靴箱包类商品</td>
                <td><a href="#">修改</a><a href="#">删除</a></td>
            </tr>
            <tr>
                <td><input checked="" type="checkbox"></td>
                <td>4</td>
                <td>家居饰品</td>
                <td>家居饰品类商品</td>
                <td><a href="#">修改</a><a href="#">删除</a></td>
            </tr>
        </table>
    </body>
</html>
    
```



```
</script>

</head>
<body>
  <table id="tab1" border="1" width="800" align="center" >
    <tr>
      <td colspan="5">
        <input type="button" value="反选" onclick="revSelect()">
      </td>
    </tr>
    <tr>
      <th><input type="checkbox" onclick="selectAll(this)" ></th>
      <th>分类ID</th>
      <th>分类名称</th>
      <th>分类描述</th>
      <th>操作</th>
    </tr>
    <tr>
      <td><input type="checkbox" class="itemSelect"></td>
      <td>1</td>
      <td>手机数码</td>
      <td>手机数码类商品</td>
      <td><a href="">修改</a>|<a href="">删除</a></td>
    </tr>
    <tr>
      <td><input type="checkbox" class="itemSelect"></td>
      <td>2</td>
      <td>电脑办公</td>
      <td>电脑办公类商品</td>
      <td><a href="">修改</a>|<a href="">删除</a></td>
    </tr>
    <tr>
      <td><input type="checkbox" class="itemSelect"></td>
      <td>3</td>
      <td>鞋靴箱包</td>
      <td>鞋靴箱包类商品</td>
      <td><a href="">修改</a>|<a href="">删除</a></td>
    </tr>
    <tr>
      <td><input type="checkbox" class="itemSelect"></td>
      <td>4</td>
      <td>家居饰品</td>
      <td>家居饰品类商品</td>
      <td><a href="">修改</a>|<a href="">删除</a></td>
    </tr>
  </table>
</body>
</html>
```

6.3 案例 QQ表情案例

6.3.1 效果



请发言: 🤔😭😭😭😭😭

6.3.2 代码

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8" />
  <title>QQ表情选择</title>
  <style type="text/css">
    *{margin: 0;padding: 0;list-style: none;}
    .emoji{margin:50px;}
    ul{overflow: hidden;}
    li{float: left;width: 48px;height: 48px;cursor: pointer;}
    .emoji img{ cursor: pointer; }
  </style>
  <script src="../js/jquery-3.3.1.js"></script>
  <script>
    $(function () {
      $(".emoji li img").click(function () {
        //把当前的小表情克隆，再追加到p元素中
        $(this).clone().appendTo(".word");
      });
    });
  </script>
</head>
<body>
  <div class="emoji">
    <ul>
      <li></li>
      <li></li>
      <li></li>
      <li></li>
      <li></li>
      <li></li>
      <li></li>
      <li></li>
      <li></li>
      <li></li>
      <li></li>
      <li></li>
    </ul>
    <p class="word">
      <strong>请发言: </strong>
```



```

</p>
</div>
</body>
</html>
```