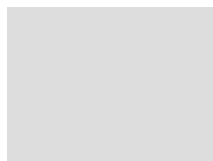
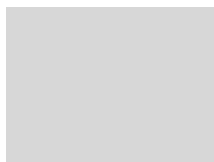
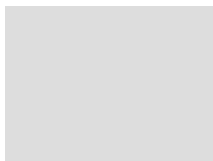


任务3 制作邮件删除效果



任务描述



- 在邮件列表前都有一个复选框，单击可以选中或取消选中。当单击“全选/全不选”按钮，所有复选框都随之选中或取消选中。单击“删除邮件”按钮可删除被选中的邮件

收件箱

状态	发件人	主题
<input type="checkbox"/>	里德	周末出去爬山，去吗？
<input type="checkbox"/>	德培	大爷，你的工作报表还没交
<input type="checkbox"/>	Moscow	Are you free tomorrow night? I'm wondering if you're free for dinner.
<input type="checkbox"/>	淮南	一不小心闹饥荒了，求赞助啊大爷！！
<input type="checkbox"/>	MO	临时有事，今天就不去找你啦，乖乖上班干活哦
<input type="checkbox"/>	xxxxxx公司	恭喜你，你获得我们公司的一等奖：价值1000w的宝马一辆！请汇款5000到.....

任务分析



1. 设计页面HTML结构，应用CSS样式。
2. 找到“全选/全不选”按钮，绑定onclick事件。
3. 查找所有的复选框，把每个复选框的checked属性值变为true或false，实现全选或全不选。
4. 找到“删除邮件”按钮，绑定onclick事件。
5. 查找所有checked属性值为true的复选框，删除其所在的行，即删除了该邮件。

相关知识点



- 删除节点
- 复制节点
- 替换节点
- 遍历节点

删除节点



方法	功能	示例
remove()	删除指定的元素，后代节点也会删除。返回值是一个指向已被删除的节点的引用	<code>\$("#p").remove()</code> 删除p元素
detach()	删除指定元素，但其绑定的事件、附加的数据都会保留下来	<code>\$("#p").detach()</code>
empty()	清空指定元素	<code>\$("#p").empty()</code> 清空p元素

remove方法示例



```
<p title="你最喜欢的运动">你最喜欢的运动是什么?</p>
```

```
<ul>
```

```
<li title="篮球">篮球</li>
```

```
<li title="足球">足球</li>
```

```
<li title="羽毛球">羽毛球</li>
```

```
</ul>
```

你最喜欢的运动是什么?

- 篮球
- 足球
- 羽毛球

删除“篮球”节点，jQuery代码如下：

```
$("ul li:eq(0)").remove();
```

你最喜欢的运动是什么?

- 足球
- 羽毛球

使用带参数的remove()方法，jQuery代码如下：

```
$("ul li").remove("li[title=篮球]");
```

复制节点



- **clone()** 方法

`$("p").clone().appendTo("body");`

可以实现复制节点，但该方法所生成的节点不会触发相应的事件。

- **clone(true)** 方法

实现复制节点，使用新复制的节点具有原节点相同的事件。

clone方法示例



```
$("#ul li").click(function() {  
    $(this).clone().appendTo("ul");  
})
```

当单击“篮球”后，列表下方出现新节点“篮球”。

你最喜欢的运动是什么？

- 篮球
- 足球
- 羽毛球
- 篮球

```
$(this).clone(true).appendTo("ul");
```


替换节点



- **replaceWith()**

`$("p").replaceWith("hello world!");`

将页面中的所有p元素替换为hello world!, 注意是每一个匹配的p都会被替换。

- **replaceAll()**

`$("Hello World!").replaceAll("p")`

将页面中的所有p元素用Hello World!替换。同样也是每一个匹配项都会被替换。

注意



- jquery中很多方法都是同一个函数实现获取(getter)和设置(setter)的。

例如：

attr()

html()

text()

val()

css()

height()

width()

.....

设置和获取HTML, 文本, 值



- html () 方法

`var tit=$("#p").html();` //获得p元素的内容

`$("#p").html("Hello World!");`//设置p的内容

- text () 方法

如果`<p>Hello World</p>`

`alert($("#p").text());` //得到Hello World!

`$("#p").text("hello world!");` 则p变成

`<p>hello world!</p>`

- val () 方法

用于获取和设置元素的值。包括：文本框 、 下拉列表、 还有单选按钮、复选框。请自己编写代码验证。

设置和获取HTML, 文本, 值



- val () 方法示例:

```
$(function() {  
    $("#username").focus(function(){  
        var txt= $(this).val();  
        if(txt=="请输入用户名")  
            $(this).val("");  
    })  
})
```

除了focus () 方法之外还有元素的blur () 方法, 这些方法仅限于表单中的相关元素。

设置和获取HTML, 文本, 值



- val () 方法补充:

可以通过val () 方法来改变下拉列表的选中项.

```
<select id="sin">
```

```
    <option>A1</option>
```

```
    <option>A2</option>
```

```
    <option>A3</option>
```

```
    <option>A4</option>
```

```
</select>
```

`$("#sin").val("A3");` //可以让A3项选中。

`$("#sin").val(["A3","A4"]);` //选中A3和A4选项。

也可以使用**`attr()`**来实现选中效果。

遍历节点



- children() 方法

用于获取指定元素下的子元素集合。（只考虑子元素，不考虑后代元素）

```
var $p=$("ul").children();
```

```
alert($p.length);
```

```
$p[2].html();           //获取第3个子元素的内容
```

- next() 方法

用于获取**匹配元素后面紧邻**的同辈元素。

如:<p></p>

则:

```
$("p").next(); //获得ul元素
```

遍历节点



- `prev()`

取得指定元素前面紧邻的同辈元素。

- `siblings()`

取得指定元素前后所有的同辈元素

- `closest()`

取得指定元素最近的匹配元素。首先检查当前元素是否匹配，如果匹配则直接返回元素本身。如果不匹配，则向上查找父元素，逐级向上，直到匹配到。如果什么也没找到，则返回一个空的jQuery对象。



jQuery中可以遍历前辈元素，方法如下：

- **parent ()：** 获取元素的父级元素
- **parents ()：** 元素元素的祖先元素

任务实现



收件箱

状态	发件人	主题
<input checked="" type="checkbox"/>	里德	周末出去爬山，去吗？
<input checked="" type="checkbox"/>	德培	大爷，你的工作报表还没交呐，再不交这个月末你就等shi啦~~~~
<input checked="" type="checkbox"/>	Moscow	Are you free tomorrow night? I'm wondering if you're free for dinner.
<input checked="" type="checkbox"/>	贱男春	一不小心闹饥荒了，求赞助啊大爷！！！！
<input checked="" type="checkbox"/>	MO	临时有事，今天就不去找你啦，乖乖上班干活哦，不然回来收拾你o(∩_∩)o
<input checked="" type="checkbox"/>	xxxxx公司	恭喜你，你获得我们公司的一等奖：价值1000w的宝马一辆！请汇款5000到.....

全选 / 全不选

删除邮件

总结



- 创建、插入节点
- 设置和获取属性
- 删除、复制、替换节点
- 遍历节点