课程目标

- 理解 jQuery 是什么，为什么要使用它。
- 区分 DOM 对象和 jQuery 对象。
- 掌握使用 jQuery 找页面元素。
- 掌握使用 jQuery 给元素绑定事件。
- 掌握使用 jQuery 操作页面元素。

一、jQuery 简介

1、jQuery 介绍

- jQuery 是一个优秀的 Javascript 框架。
- jQuery 是轻量级的 JS 库 ，它兼容 CSS3，还兼容各种浏览器（IE 6.0+，FF 1.5+，Safari 2.0+，Opera 9.0+）。
- jQuery 是免费、开源的。
- jQuery 是一个兼容多浏览器的 Javascript 库，核心理念是 write less，do more（写得更少，做得更多）。

2、jQuery 版本介绍

- jQuery1.x：经典版本，兼容 IE6，7，8。
- jQuery2.0：改进版本，及后续版本将不再支持 IE6，7，8 浏览器。

3、jQuery 能干什么

jQuery 使用户能更方便地处理 HTML（标准通用标记语言下的一个应用）、events、实现动画效果，并且方便地为网站提供 AJAX 交互。
jQuery 的语法设计可以使开发者更加便捷，例如操作文档对象、选择 DOM 元素、制作动画效果、事件处理、使用 AJAX 以及其他功能。

4、jQuery 文件介绍

jQuery1.x.js：jQuery 源文件；学习 jQuery 或者 debug 的时候使用。
jQuery1.x.min.js：jQuery 压缩之后的文件；正常项目中使用。

二、jQuery 引入与体验

1、拷贝 jQuery 文件到项目中

把资料中的 jQuery-1.11 文件夹拷贝到项目的 webapp 下的 static 目录下。

2、引入 jQuery

新建 webapp/jq_01/01.jQuery_hello.html，在文件中引入 jQuery，如下：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>jQuery 入门</title>
    <script src="/static/jQuery-1.11/jquery-1.11.3.min.js"></script>
</head>
<body>
</body>
</html>
```

3、验证引入 jQuery

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>jQuery 入门</title>
    <script src="/static/jQuery-1.11/jquery-1.11.3.min.js"></script>
    <script>
        alert($);
    </script>
</head>
<body>
</body>
</html>
```

若控制台出现 ReferenceError:$ is not defined 这样的错误，证明引入 jQuery 失败。

4、需求

点击按钮，显示文本和隐藏文本。

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>jQuery 入门</title>
    <script src="/static/jQuery-1.11/jquery-1.11.3.min.js"></script>
    <script>
        // 使用原生JS
        /*
        window.onload = function () {
            var btn = document.getElementById("toggle");
            var temp;
            btn.onclick = function () {
                var div = document.getElementById("content");
                var content = div.innerHTML;
                if(content){ // 当显示，改成隐藏
                    temp = content; // 存起内容
                    div.innerHTML = ''; // 清除内容
                }else { // 当隐藏，改成显示
                    div.innerHTML = temp;
                }
            }
        }*/

        // 使用 jQuery
        $(function () {
            $('#toggle').click(function () {
                $('#content').toggle(3000);
            });
        });
    </script>
</head>
<body>
<div id="content">
    【乘客#为少付1元车费致两公交相撞#：辱骂并拉拽驾驶员使公交车失控】5月7日16时18分，浙江宁波一男性乘客因不愿按规定缴纳车费，辱骂并强行拉拽驾驶员胳膊，致使正常行使中的公交车失控，穿过中间绿化带与对向行驶的公交车相撞。事故造成2名驾驶员和该肇事乘客受伤，无人员死亡。
</div>
<button id="toggle">切换</button>
</body>
</html>
```

三、jQuery 对象

1、问题引入

通过 document.getElementById() 找到的元素和通过 $() 找不到元素不一样：

- 通过 jQuery 方法获取的页面元素，都是 jQuery 对象。
- jQuery 对象其实就是对 DOM 对象进行了包装，增强相关了方法，让开发者使用起来更加便利。
- **虽然 jQuery 对象包装了 DOM 对象但是两者不能混用**，各位可以理解为 jQuery 对象与 DOM 对象是两个不类型的对象，但是我们调用 jQuery 对象的方法，事实上底层代码还是操作的是 DOM 对象。

2、jQuery 对象与 DOM 对象之间转换

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>jQuery_Object_vs_DOM_Object</title>
    <script src="/static/jQuery-1.11/jquery-1.11.3.min.js"></script>
    <script>
        console.log($ === jQuery); // 都是一个函数

        window.onload = function () {
            var btn = document.getElementById("btn");
            console.log(btn); // btn 变量存的就是 DOM 对象

            var $btn = $('#btn');  // 找页面元素 jQuery
            console.log($btn);

            // DOM 对象转成 jQuery 对象
            console.log($(btn));

            // jQuery 对象转成 DOM 对象，这个很少用
            console.log($btn.get(0));

            // 以后尽量使用 jQuery 方法操作页面元素，绑定事件等等.
        }
    </script>
</head>
<body>
<button id="btn">按钮</button>
</body>
</html>
```

3、$ 与 jQuery

新建 webapp/jq_01/02.jQuery_Object_vs_DOM_Object.html，代码如下：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>jQuery_Object_vs_DOM_Object</title>
    <script src="/static/jQuery-1.11/jquery-1.11.3.min.js"></script>
    <script>
        console.log($ === jQuery); // 都是一个函数
    </script>
</head>
<body>
</body>
</html>
```

四、jQuery 对象常用方法

1、准备页面

新建 webapp/jq_01/03.normal_method.html，拷贝代码如下：

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>jQuery 常用方法</title>
<script type="text/javascript" src="/static/jQuery-1.11/jquery-1.11.3.min.js">
</script>
</head>
<body>
    <h1 id="h1">做人<i>要低调</i></h1>
    <input type="text" id="username" value="wolfcode"/><br/><br/>
    <div>
        jQuery 常用方法:<br/>
        jQuery对象.size();    // 获取 jQuery 中包含元素的个数<br/>
        jQuery对象.val(); // 操作元素的 value 属性<br/>
        jQuery对象.html();    // 操作元素内的 HTML 代码<br/>
        jQuery对象.text();    // 操作元素内的文本，忽略 HTML 标签<br/>
        jQuery对象.css();     // 操作元素的 style 属性
    </div>
    <hr/>
    <div>
        <p>
            问题 1：获取 jQuery 中包含 DOM 的个数，比如获取页面上 p 元素的个数
        </p>
        <p>
            问题 2：获取 id 为 username 元素的 value 属性值
        </p>
        <p>
            问题 3：设置 id 为 username 元素的 value 属性值为"叩丁狼教育"
        </p>
        <p>
            问题 4：对比 h1 元素的内容和纯文本的区别
        </p>
        <p>
            问题 5：把 h1 元素内容的颜色改为黄色
        </p>
    </div>
</body>
</html>
```

2、练习

注意在上面的页面引入 jQuery 之后加入下面的代码：

```
<script>
    $(function () {
        // 问题 1：获取 jQuery 中包含 DOM 的个数，比如获取页面上 p 元素的个数
        console.log($('p').size());
        // 问题 2：获取 id 为 username 元素的 value 属性值
        console.log($('#username').val());
        // 问题 3：设置 id 为 username 元素的 value 属性值为"叩丁狼教育"
        var $ret = $('#username').val('叩丁狼教育');
```

```
        console.log($ret); // 返回的是被修改 value 属性值的 jQuery 对象
        // 问题 4：对比 h1 元素的内容和纯文本的区别
        console.log($('#h1').text());
        console.log($('#h1').html());
        // 若我想修改这 h1 元素中内容
        $('#h1').html('人生要有一点绿');
        // 问题 5：把 h1 元素内容的颜色改为黄色
        $('#h1') .css('color', 'green');
    });
</script>
```

五、jQuery 选择器

1、作用

jQuery 选择器是 jQuery 类库最重要功能之一，**jQuery 提供获取页面元素一种语法。**这些选择器的用法和 CSS 的语法非常相似，结合 jQuery 类库的方法你可以很方便快速地定位页面中任何元素，并为其添加响应的行为。

2、选择器的组成

选择器一般由“特殊符号”+“字符串”组成。比如:“#”代表 id，“mydiv”是一个字符串，那么整体 #mydiv 代表的是 id 为 mydiv 的元素对象。

3、如何使用选择器获取元素

语法：$(“选择器”)，如 $ (“#mydiv”)。
注意：如果通过 jQuery 方法获取页面中元素，没有查找到，返回值不是 null，返回值为一个空数组 []，所以判断是否获取到元素，通过 jQuery.size() != 0 来判断。

4、学习技巧

记住一些常用的即可。

六、基本选择器

1、基本选择器

jQuery 最常用，最简单选择器，通过元素的 id、class 或标签等查找元素。在网页中，每个 id 名称只能使用一次[0, 1]，class 允许重复使用 [0, n]。

1.1、id 选择器

#id，用法：$(“#myDiv”) 返回值单个元素的组成的集合。
说明：这个就是直接选择 html 中的 id=“myDiv”。

1.2、元素选择器

Element，用法：$(“div”) 返回值元素集合。
说明：element 的英文翻译过来是“元素”，所以 element 其实就是 HTML 已经定义的标签元素，例如 div，input，a 等等。

1.3、类选择器

class，用法：$(".myClass") 返回值元素集合。
说明：这个标签是直接选择 HTML 代码中 class="myClass" 的元素或元素组（因为在同一 HTML 页面中 class 是可以存在多个同样值的元素）。

2、练习

2.1、准备页面

新建 webapp/jq_01/04.normal_selector.html，拷贝代码如下：

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>jQuery 常用选择器</title>
<script type="text/javascript" src="/static/jQuery-1.11/jquery-1.11.3.min.js">
</script>
<style type="text/css"> .selected{background-color: gray;}</style>
</head>
<body>
    <div id="msg">使用 ID 选择器获取当前 DIV元素</div>
    <ul>
        <li>item1</li>
        <li>item2</li>
        <li>item3</li>
        <li>item4</li>
    </ul>
    <ul id="myul">
        <li>item1</li>
        <li class="selected">item2</li>
        <li>item3</li>
        <li class="selected">item4</li>
    </ul>
    <hr/>
    <div>
        <p>
            问题 1：获取 id 为 msg 的元素的内容
        </p>
        <p>
            问题 2：获取所有的 li 元素并打印数量
        </p>
        <p>
            问题 3：获取所有 class 为 selected 的元素，字体颜色改为 red
        </p>
    </div>
</body>
</html>
```

2.2、做练习

注意在上面的页面引入 jQuery 之后加入下面的代码：

```
<script>
    $(function () {
```

```
        // 问题 1：获取 id 为 msg 的元素的内容
        console.log($('#msg').html());
        // 问题 2：获取所有的 li 元素并打印数量
        console.log($('li').size());
        // 问题 3：获取所有 class 为 selected 的元素，字体颜色改为 red
        $('.selected').css('color', 'red'); // 找到多少改多少

        console.log($('*').size());
        // , 相当于或
        console.log($('#msg,.selected').size());
    });
</script>
```

七、层次选择器

1、需求

若想通过 DOM 元素之间的层次关系来获取特定元素，例如后代元素，子元素， 相邻元素，兄弟元素等，则需要使用层次选择器。

2、层次选择器

2.1、ancestor descendant

用法：$(“form input”)
说明：在给定的祖先元素下匹配所有后代元素。

2.2、parent > child

用法：$(“form > input”)
说明：在给定的父元素下匹配所有子元素。

2.3、prev + next

用法：$(”label + input“)
说明：匹配所有紧接在 prev 元素后的 next 元素。

2.4、prev ~ siblings

用法：$(“form ~ input”)
说明：匹配 prev 元素之后的所有 siblings 元素。
注意：是匹配之后的元素，不包含该元素在内，并且 siblings 匹配的是和 prev 同辈的元素，其后辈元素不被匹配。

3、练习

3.1、准备页面

新建 webapp/jq_01/05.hierarchy_selector.html，拷贝代码如下：

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

```
<title>jQuery 层次选择器</title>
<script type="text/javascript" src="/static/jQuery-1.11/jquery-1.11.3.min.js">
</script>
<style type="text/css"> .selected{background-color: gray;}</style>
</head>
<body>
    <ul id="myul">
        <li>item1</li>
        <li>item2</li>
        <li>item3</li>
        <li>
            <ul>
                <li>item1</li>
                <li>item2</li>
                <li>item3</li>
                <li>item4</li>
            </ul>
        </li>
    </ul>

    <label>LABEL1</label>
    <input type="text" value="text1"/>
    <input type="text" value="text2"/>
    <br/>
    <label>LABEL2</label>
    <input type="text" value="text3"/>
    <input type="text" value="text4"/>
    <br/>
    <label>
        LABEL3
        <input type="text" value="text5"/>
        <input type="text" value="text6"/>
    </label>
    <hr/>
    <div>
        <p>
            问题 1：获取所有 ul 下的所有 li 元素，并打印分析结果
        </p>
        <p>
            问题 2：获取 id 为 myul 下的所有子 li 元素，并打印分析结果
        </p>
        <p>
            问题 3：获取所有 label 元素后的 input 元素，并打印分析结果
        </p>
        <p>
            问题 4：获取紧跟着 label 元素后的 input 元素，并打印分析结果
        </p>
    </div>
</body>
</html>
```

3.2、做练习

注意在上面的页面引入 jQuery 之后加入下面的代码：

```
<script>
    $(function () {
        // 问题 1：获取所有 ul 下的所有 li 元素，并打印分析结果
        console.log($('ul li'));
        // 问题 2：获取 id 为 myul 下的所有子 li 元素，并打印分析结果
        console.log($('#myul > li'));
        // 问题 3：获取所有 label 元素后的 input 元素，并打印分析结果
        console.log($('label ~ input'));
        // 问题 4：获取紧跟着 label 元素后的 input 元素，并打印分析结果
        console.log($('label + input'));
    });
</script>
```

八、过滤选择器

1、定义

过滤选择器：通过特定的过滤规则来筛选所需要的 DOM 元素，过滤规则与 CSS 中的伪类选择器语法相同。
该选择器一般以一个冒号（:）开头，按照不同的过滤规则，可分为基本过滤，内容过滤，可见性过滤，属性过滤，子元素过滤，表单对象属性过滤选择器。

2、练习

2.1、准备页面

新建 webapp/jq_01/06.filter_selector.html，拷贝代码如下：

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>jQuery 过滤选择器</title>
<script type="text/javascript" src="/static/jQuery-1.11/jquery-1.11.3.min.js">
</script>
<style type="text/css"> .selected{background-color: gray;}</style>
</head>
<body>
    <input type="hidden" name="id" value="1">
    <select>
        <option value="1">Flowers</option>
        <option value="2" selected>Gardens</option>
        <option value="3">Trees</option>
    </select>
    <hr/>
    <div>
        <p>
            问题 1：获取隐藏 input 的 value 属性值，不能使用根据元素名，也不能通过给元素加 id 属性，再通过 id 选择器找
        </p>
        <p>
            问题 2：获取选中的 option
        </p>
    </div>
</body>
```

```
</html>
```

2.2、做练习

注意在上面的页面引入 jQuery 之后加入下面的代码：

```
<script>
    $(function () {
        // 问题 1：获取隐藏 input 的 value 属性值，不能使用根据元素名，也不能通过给元素加 id 属性，再通过 id 选择器找
        console.log($('input[name=id]').val());
        // 问题 2：获取选中的 option
        console.log($('option:selected').val());
    });
</script>
```

九、jQuery事件绑定

1、传统的事件绑定

1.1、标签中使用on事件属性

```
<button onclick="clickT()"></button>
```

1.2、通过JS给标签设置 on 事件属性

```
btn.onCliick = function(){}
```

1.3、通过JS调用方法的方式

- W3C：btn.addEventListner(事件名, 响应函数);
- IE：btn.attchEvent(事件名, 响应函数);

2、jQuery 事件绑定

查看 jQuery 文档：jQuery对象.click(fn)，页面加载事件处理，对比之前使用原生的区别，之前只能绑定一个处理函数。

3、练习

新建 webapp/jq_01/07.jQuery_event_bind.html，代码如下：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>jQuery 事件绑定</title>
    <script src="/static/jQuery-1.11/jquery-1.11.3.min.js"></script>
    <script>
        $(function () {
```

```
            $('#btn1').click(function () { // 匿名函数什么时候执行，当用户在页面点击这个按钮就会执行这个函数
                console.log(this); // this 就是事件源 DOM 对象
                console.log($(this)); // 事件源 DOM 对象 => jQuery 对象
            });

            // 找到多少个绑定多少个
            $('.myBtn').click(function () {
                console.log('hello');
            });
        });

        window.onload = function () {
            console.log(1);
        };
        window.onload = function () {
            console.log(2);
        };

        $(function () {
           console.log(3);

        });
        $(function () {
            console.log(4);
        });
    </script>
</head>
<body>
<button id="btn1">btn1</button>
<button id="btn2" class="myBtn">btn2</button>
<button id="btn3" class="myBtn">btn3</button>
</body>
</html>
```

十、jQuery 常用 DOM 操作的方法

1、append 方法

给元素添加子元素，且是最小的子元素。

2、after 方法

给元素添加弟弟元素。

3、empty、remove 和 detach 方法

empty 断子绝孙，remove 和 detach 自杀。

4、练习

4.1、准备页面

新建 webapp/jq_01/08.jQuery_DOM.html，拷贝代码如下：

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>jQuery DOM 操作</title>
<script type="text/javascript" src="/static/jQuery-1.11/jquery-1.11.3.min.js">
</script>
</head>
<body>
    <span style="background-color : blue;color: red;" id="span">SPAN</span>
    <div id="div1" style="background-color: gray;">DIV1</div>
    <div id="div2" style="background-color: green;">DIV2</div>

    <ul id="ul">
        <li>item1</li>
        <li>item2</li>
        <li>item3</li>
        <li>item4</li>
        <li>item5</li>
    </ul>

    <input id="btn" type="button" value="删除我"/>

    <form>
        <fieldset>
            <legend>内部插入节点(插入子节点)</legend>
            <input type="button" value="append" id="append"/>
        </fieldset>
    </form>

    <form>
        <fieldset>
            <legend>外部插入节点(插入兄弟节点)</legend>
            <input type="button" value="after" id="after"/>
        </fieldset>
    </form>

    <form>
        <fieldset>
            <legend>删除节点</legend>
            <input type="button" value="删除所有子节点" id="empty"/>
            <input type="button" value="删除节点" id="remove"/>
            <input type="button" value="恢复节点" id="resume"/>
        </fieldset>
    </form>
</body>
</html>
```

4.2、做练习

注意在上面的页面引入 jQuery 之后加入下面的代码：

```
<script>
    $(function () {
        $('#append').click(function () {
            // 父亲加小儿子
            $('#div1').append($('#span'));
```

```
        });

        $('#after').click(function () {
            // 哥哥加弟弟
            $('#div2').after($('#div1'));
        });

        $('#empty').click(function () {
            $('#ul').empty();
        });

        $('#btn').click(function () {
            console.log(1);
        });

        var $btn;
        $('#remove').click(function () {
            $btn = $('#btn').remove(); // 这个方法返回被删除元素
        });
        $('#resume').click(function () {
            $('#ul').after($btn);
        });
    });
</script>
```

十一、jQuery 元素属性操作的方法

1、属性操作的方法

- css
- val
- addClass，removeClass
- prop
- data
- attr

2、练习

2.1、准备页面

新建 webapp/jq_01/09.jQuery_attr.html，拷贝代码如下：

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>jQuery 属性操作</title>
<!-- 引入jQuery -->
<script type="text/javascript" src="/static/jQuery-1.11/jquery-1.11.3.min.js">
</script>
<style type="text/css">
    .other {
        background-color: orange;
        font-size: 20px;
    }
```

```
    .myBtn{
        background-color: red;
    }
</style>
</head>
<body>
    <input id="btn" type="button" value="点我"/>
    <form>
        <fieldset>
            <legend>属性操作</legend>
            <input type="button" value="获取属性值" id="getAttr"/>
            <input type="button" value="设置属性值" id="setAttr"/>
        </fieldset>
    </form>

    <form>
        <fieldset>
            <legend>CSS 操作</legend>
            <input type="button" value="添加CSS" id="addClass"/>
            <input type="button" value="删除CSS" id="removeClass"/>
            <input type="button" value="轮换CSS" id="toggleClass"/>
            <input type="button" value="判断CSS" id="hasClass"/>
        </fieldset>
    </form>

    <input type="checkbox" value="1" checked name="gender" id="gender" data-
option='{"name" : "蒋干"}'
        style="color: red; background: aqua" class="myBtn other" > 男
</body>
</html>
```

2.2、做练习

注意在上面的页面引入 jQuery 之后加入下面的代码：

```
<script>
    $(function () {
        $('#getAttr').click(function () {
            console.log($('#btn').attr('type'));
        });
        $('#setAttr').click(function () {
            $('#btn').attr('my', 'haha');
        });

        $('#addClass').click(function () {
            $('#btn').addClass('other');
        });
        $('#removeClass').click(function () {
            $('#btn').removeClass('other');
        });
        $('#toggleClass').click(function () {
            $('#btn').toggleClass('other');
        });
        $('#hasClass').click(function () {
            console.log($('#btn').hasClass('other'));
        });
    });
```

```
</script>
```

3、操作属性方法总结

注意在上面的页面引入 jQuery 之后加入下面的代码：

```
<script>
    $(function () {
        // css 方法针对 style 属性
        console.log($('#gender').css('color'));
        // val 方法针对 value 属性
        console.log($('#gender').val());
        // addClass 方法、removeClass 方法，针对 class 属性
        $('#gender').removeClass('other');
        // prop 方法针对 checked 属性 selected 属性
        console.log($('#gender').prop('checked'));
        // $('#gender').prop('checked', false)
        // data 方法，针对 data- 开头的属性的获取，若值是 JSON 格式，自动转成 JS 对象
        console.log($('#gender').data('option'));
        // 其他情况使用 attr 方法
    });
</script>
```

十二、练习-下拉框回显

1、准备页面

新建 webapp/jq_01/10.echo.html，拷贝代码如下：

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>下拉框回显</title>
<script src="/static/jQuery-1.11/jquery-1.11.3.min.js"></script>
</head>
<body>
    <select id="s1">
        <option value="1">选项1</option>
        <option value="2">选项2</option>
        <option value="3">选项3</option>
        <option value="4">选项4</option>
        <option value="5">选项5</option>
    </select>
    <br/>
    <input type="button" value="回显第3个选项" onclick="echo();"/><br/>
</body>
</html>
```

2、代码实现

```
<script>
    function echo() {
        // $('#s1').val("3");
        $('#s1 > option:eq(2)').prop('selected', true);
    }
</script>
```

十三、练习-列表移动

1、准备页面

新建 webapp/jq_01/11.select.html，拷贝代码如下：

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>列表移动</title>
<script type="text/javascript"  src="/static/jQuery-1.11/jquery-1.11.3.min.js">
</script>
</head>
<body>
    <table border="1">
        <tr>
            <td>
                <select id="select1" style="width:100px;height:200px" size="10"
multiple="multiple">
                    <option value="选项1">选项1</option>
                    <option value="选项2">选项2</option>
                    <option value="选项3">选项3</option>
                    <option value="选项4">选项4</option>
                    <option value="选项5">选项5</option>
                    <option value="选项6">选项6</option>
                    <option value="选项7">选项7</option>
                    <option value="选项8">选项8</option>
                    <option value="选项9">选项9</option>
                </select>
            </td>
            <td align="center">
                <input type="button" onclick="moveSelected('select1','select2')"
value="-->"/><br/>
                <input type="button" onclick="moveAll('select1','select2')"
value="==>"/><br/>
                <input type="button" onclick="moveSelected('select2','select1')"
value="<--"/><br/>
                <input type="button" onclick="moveAll('select2','select1')"
value="<=="/>
            </td>
            <td>
                <select id="select2" style="width:100px;height:200px" size="10"
multiple="multiple"></select>
            </td>
        </tr>
    </table>
</body>
</html>
```

2、代码实现

```
<script>
    function moveAll(srcId, targetId) {
        $('#' + targetId).append($("#" + srcId + " > option"));
    }

    function moveSelected(srcId, targetId) {
        $('#' + targetId).append($('#' + srcId + ' > option:selected'));
    }
</script>
```

十四、练习-下拉框去重

1、准备页面

新建 webapp/jq_01/12.distinct.html，拷贝代码如下：

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>下拉框去重</title>
<script type="text/javascript" src="/static/jQuery-1.11/jquery-1.11.3.min.js">
</script>
</head>
<body>
    <table border="1">
        <tr>
            <td>
                <select id="s1" style="width:100px;" size="10"
multiple="multiple" >
                    <option value="1">选项1</option>
                    <option value="2">选项2</option>
                    <option value="3">选项3</option>
                    <option value="4">选项4</option>
                    <option value="5">选项5</option>
                </select>
            </td>
            <td align="center">
                <input type="button" value="去除重复" onclick="distinct();"/>
<br/>
            </td>
            <td>
                <select id="s2" style="width:100px;" size="10"
multiple="multiple" >
                    <option value="1">选项1</option>
                    <option value="3">选项3</option>
                    <option value="5">选项5</option>
                </select>
            </td>
        </tr>
    </table>
</body>
```

```
</html>
```

2、代码实现

```
<script>
    function distinct() {
        // 先获取右边 select 中 option 获取 value 属性值，存到一个数组中
        var arr = [];
        $('#s2 > option').each(function(i, domEle){
            var val = $(domEle).val();
            arr.push(val); // 往数组最后添加一个元素
        });
        console.log(arr);

        // 遍历左边 select 中 option 获取其 value 值，跟上面数组中到对比，若存在，则删除
对应 option
        $('#s1 > option').each(function(i, domEle){
            var $option = $(domEle);
            var val = $option.val();
            if($.inArray(val, arr) >= 0){ // 此 val 在数组存在
                $option.remove(); // 删除对应 option 元素
            }
        });
    }
</script>
```

十五、练习-全选

1、准备页面

新建 webapp/jq_01/13.checkbox.html，拷贝代码如下：

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>全选</title>
<script type="text/javascript" src="/static/jQuery-1.11/jquery-1.11.3.min.js">
</script>
</head>
<body>
    请选择你的爱好:<br/>
    <input type="checkbox" onchange="checkChange(this)" id="checkAll"/>全选/全不选
<br/>
    <div>
        <input type="checkbox" name="hobby"/>JAVA 
        <input type="checkbox" name="hobby"/>打篮球 
        <input type="checkbox" name="hobby"/>上网 
        <input type="checkbox" name="hobby"/>撩妹 
    </div>

    <div>
        <input type="button" id="btn_checkAll" onclick="checkAll(true)"
value="全选"/>
        <input type="button" onclick="checkAll(false)" value="全不选"/>
```

```
        <input type="button" onclick="checkUnAll()" value="反选"/>
    </div>
</body>
</html>
```

2、代码实现

```
<script>
    function checkChange(src) {
        // 根据被点击的复选框，来决定下面这些爱好的复选框的选中状态
        var checked = $(src).prop('checked');
        checkAll(checked);
    }

    function checkAll(flag) {
        $('[name=hobby]').prop('checked', flag);
        // 点击全选按钮的时候，最上面复选框要选中
        // 点击全不选按钮的时候，最上面复选框不要选中
        $('#checkAll').prop('checked', flag);
    }

    // 反选
    function checkUnAll() {
        $('[name=hobby]').each(function (i, domEle) {
            // 获取其选中状态
            var oldValue = $(domEle).prop('checked');
            // 取反
            var newValue = !oldValue;
            // 再设置回去
            $(domEle).prop('checked', newValue);

            // $(domEle).prop('checked', !$(domEle).prop('checked'));
        });

        // 点击反选，若爱好全选中，最上面复选框要选中，反之不要选中
        check();
    }

    $(function () {
        // 点击所有爱好复选框，若爱好全选中，最上面复选框要选中，反之不要选中
        $('[name=hobby]').click(function () {
            check();
        });
    });


    function check() {
        var total = true; // 统计
        $('[name=hobby]').each(function (i, domEle) {
            var checked = $(domEle).prop('checked'); // 取每个爱好选中状态值
            total = total && checked; // 跟变量 total 与并设置存到变量 total 中
        });

        // 当遍历结束的时候，total 值仍是 true，那么代表所有爱好复选框是选中的
        $('#checkAll').prop('checked', total);
    }
```

```
</script>
```

练习

- 完成使用分别使用代码打印 DOM 对象与 jQuery 对象。
- 完成使用 jQuery 基本选择器、层次选择器和过滤选择器查找页面元素，并控制台打印这些元素，所要使用选择器如下：
  - `id 选择器`
  - `元素选择器`
  - `class 选择器`
  - `ancestor descendant 选择器`
  - `parent > child 选择器`
  - `prev + next 选择器`
  - `prev ~ siblings 选择器`
  - `[attribute=value] 选择器`
  - `:selected 选择器`
- 完成使用 jQuery 给页面元素绑定事件：
  - 点击一个按钮打印`今天天气很好`。
  - 给输入框绑定失去焦点事件，当触发事件时打印输入框元素的 value 属性值。
  - 给下拉框绑定值改变事件，当出发事件时打印用户选择 option 的 value 属性值。
- 完成使用 jQuery 操作页面元素，
  - 给页面某元素加子元素。
  - 给页面某元素加弟弟元素。
  - 删除页面某元素的子孙元素。
  - 删除页面某元素。
  - 获取页面某元素的 style 属性值。
  - 追加，移除页面某元素的 class 属性值。
  - 获取和设置页面某元素的 value 属性值。
  - 获取和设置页面某元素的 checked 属性值。
  - 获取页面某元素的 data- 开头的属性值。