

操作表单

用JavaScript操作表单和操作DOM是类似的，因为表单本身也是DOM树。

不过表单的输入框、下拉框等可以接收用户输入，所以用JavaScript来操作表单，可以获得用户输入的内容，或者对一个输入框设置新的内容。

HTML表单的输入控件主要有以下几种：

- 文本框，对应的>，用于输入文本；
- 口令框，对应的>，用于输入口令；
- 单选框，对应的☐>，用于选择一项；
- 复选框，对应的☐>，用于选择多项；
- 下拉框，对应的<select>，用于选择一项；
- 隐藏文本，对应的>，用户不可见，但表单提交时会把隐藏文本发送到服务器。

获取值

如果我们获得了一个节点的引用，就可以直接调用value获得对应的用户输入值：

```
// <input type="text" id="email">
var input = document.getElementById('email');
input.value; // '用户输入的值'
```

这种方式可以应用于、、以及<select>。但是，对于单选框和复选框，value属性返回的永远是HTML预设的值，而我们需要获得的实际是用户是否“勾上了”选项，所以应该用checked判断：

```
// <label><input type="radio" name="weekday" id="monday" value="1"> Monday</label>
// <label><input type="radio" name="weekday" id="tuesday" value="2"> Tuesday</label>
var mon = document.getElementById('monday');
var tue = document.getElementById('tuesday');
mon.value; // '1'
tue.value; // '2'
mon.checked; // true或者false
tue.checked; // true或者false
```

设置值

设置值和获取值类似，对于`text`、`password`、`hidden`以及`select`，直接设置`value`就可以：

```
// <input type="text" id="email">
var input = document.getElementById('email');
input.value = 'test@example.com'; // 文本框的内容已更新
```

对于单选框和复选框，设置`checked`为`true`或`false`即可。

HTML5控件

HTML5新增了大量标准控件，常用的包括`date`、`datetime`、`datetime-local`、`color`等，它们都使用`<input>`标签：

```
<input type="date" value="2015-07-01">
```

```
<input type="datetime-local" value="2015-07-01T02:03:04">
```

```
<input type="color" value="#ff0000">
```

不支持HTML5的浏览器无法识别新的控件，会把它们当做`type="text"`来显示。支持HTML5的浏览器将获得格式化的字符串。例如，`type="date"`类型的`input`的`value`将保证是一个有效的`YYYY-MM-DD`格式的日期，或者空字符串。

提交表单

最后，JavaScript可以以两种方式来处理表单的提交（AJAX方式在后面章节介绍）。

方式一是通过`<form>`元素的`submit()`方法提交一个表单，例如，响应一个`<button>`的`click`事件，在JavaScript代码中提交表单：

```
<!-- HTML -->
<form id="test-form">
  <input type="text" name="test">
  <button type="button" onclick="doSubmitForm()">Submit</button>
</form>

<script>
function doSubmitForm() {
  var form = document.getElementById('test-form');
  // 可以在此修改form的input...
  // 提交form:
  form.submit();
}
```

```
</script>
```

这种方式的缺点是扰乱了浏览器对form的正常提交。浏览器默认点击`<button type="submit">`时提交表单，或者用户在最后一个输入框按回车键。因此，第二种方式是响应`<form>`本身的`onsubmit`事件，在提交form时作修改：

```
<!-- HTML -->
<form id="test-form" onsubmit="return checkForm()">
  <input type="text" name="test">
  <button type="submit">Submit</button>
</form>

<script>
function checkForm() {
  var form = document.getElementById('test-form');
  // 可以在此修改form的input...
  // 继续下一步:
  return true;
}
</script>
```

注意要`return true`来告诉浏览器继续提交，如果`return false`，浏览器将不会继续提交form，这种情况通常对应用户输入有误，提示用户错误信息后终止提交form。

在检查和修改`<input>`时，要充分利用`<input type="hidden">`来传递数据。

例如，很多登录表单希望用户输入用户名和口令，但是，安全考虑，提交表单时不传输明文口令，而是口令的MD5。普通JavaScript开发人员会直接修改`<input>`：

```
<!-- HTML -->
<form id="login-form" method="post" onsubmit="return checkForm()">
  <input type="text" id="username" name="username">
  <input type="password" id="password" name="password">
  <button type="submit">Submit</button>
</form>

<script>
function checkForm() {
  var pwd = document.getElementById('password');
  // 把用户输入的明文变为MD5:
  pwd.value = toMD5(pwd.value);
  // 继续下一步:
  return true;
}
```

```
</script>
```

这个做法看上去没啥问题，但用户输入了口令提交时，口令框的显示会突然从几个*变成32个*（因为MD5有32个字符）。

要想不改变用户的输入，可以利用实现：

```
<!-- HTML -->
<form id="login-form" method="post" onsubmit="return checkForm()">
  <input type="text" id="username" name="username">
  <input type="password" id="input-password">
  <input type="hidden" id="md5-password" name="password">
  <button type="submit">Submit</button>
</form>

<script>
function checkForm() {
  var input_pwd = document.getElementById('input-password');
  var md5_pwd = document.getElementById('md5-password');
  // 把用户输入的明文变为MD5:
  md5_pwd.value = toMD5(input_pwd.value);
  // 继续下一步:
  return true;
}
</script>
```

注意到id为md5-password的标记了name="password"，而用户输入的id为input-password的没有name属性。没有name属性的的数据不会被提交。

练习

利用JavaScript检查用户注册信息是否正确，在以下情况不满足时报错并阻止提交表单：

- 用户名必须是3-10位英文字母或数字；
- 口令必须是6-20位；
- 两次输入口令必须一致。

```
<!-- HTML结构 -->
<form id="test-register" action="#" target="_blank" onsubmit="return checkRegisterForm()">
  <p id="test-error" style="color:red"></p>
  <p>
    用户名: <input type="text" id="username" name="username">
  </p>
```