

# 1. 表单基础

## 1. 取得表单: document.forms

```
var form1=document.forms[0];  
var form2=document.forms["form's name"];
```

## 2. 提交表单:

### 三种提交按钮

```
(1) <input type="submit" value="Submit">  
(2) <button type="submit">Submit</button>  
(3) <input type="image" src="xx.gif">
```

只要表单中存在上述任何一种按钮, 在相应表单控件拥有焦点的情况下, 按回车就可以提交表单。

### submit事件

以上述按钮方式提交表单时, 浏览器会在请求发送给服务器之前触发submit事件。这样就有机会验证表单是否提交。

以下代码可阻止表单提交

```
var form=document.forms[0];  
EventUtil.addHandler(form,"submit",function(event){  
    event=EventUtil.getEvent(event);  
    Event.preventDefault(event);  
})
```

### submit()方法提交表单

- 该方式提交的表单无需表单包含提交按钮。
- 该方式提交表单不会触发submit事件。

```
var form=document.forms[0]; form.submit();
```

### 提交表单的问题

用户可能会重复提交表单, 带来问题。

解决办法是: 第一次提交后就禁用提交按钮。

## 3. 重置表单

### 两种重置按钮

```
<input type="reset" value="Reset">  
<button type="reset">Reset</button>
```

初始值为空的字段就恢复为空; 带有默认值的字段就恢复为默认值。

### reset事件

单击重置按钮会触发reset事件。可利用这个机会取消重置操作。

取消重置操作按钮:

```
var form=document.forms[0];  
EventUtil.addHandler(form,"reset",function(event){  
    event=EventUtil.getEvent(event);
```

```
    EventUtil.preventDefault(event);
  })
```

## reset()方法

- 也可通过JS的reset方法重置表单。
- 和submit()方法不会触发submit事件不同，该方法也会触发reset事件。

```
form.reset()
```

## 4.表单字段

### (1)取得表单字段

form.elements集合包含着表单中的所有字段，如input、textarea、button、fieldset。

```
var field1=form.elements[0];//取得表单中第一个字段
var field2=form.elements["field1"]//取得名为field1的字段
var count=form.elements.length;//取得表单中包含的字段数量
```

### (2)共有的表单字段属性

- disabled:布尔值，表示当前字段是否被禁用
- form:指向当前字段所属表单，只读
- name:当前字段的名称
- type:当前字段类型，如“checkbox”、“radio”等
- value:当前字段提交给服务器的值

除form外，以上属性都可修改。

### (3)动态修改字段属性和方法

```
field1.value="Another value";//修改value属性
console.log(field1.form==form);//检查字段form属性
field1.focus();//把焦点设置到当前字段
field1.disabled=true;//禁用当前字段
```

动态操作表单有用实例：避免重复提交表单

```
EventUtil.addHandler(form,"submit",function(event){
    event=EventUtil.getEvent(event);
    var target=EventUtil.getTarget(event);

    var btn=target.elements["submit-btn"];//取得提交按钮
    btn.disabled=true;//禁用它
});
```

注意：只能用侦听submit事件来禁用提交按钮，不能用click事件，因为click事件和提交行为之间有时差。对于一先触发click事件再提交的浏览器，如果在提交之前禁用按钮，则永远都不会提交表单。

### (4) 共有的表单字段方法：focus()和blur()

eg:页面加载后自动聚焦于第一个字段

```
EventUtil.addHandler(window,"load",function(event){
    if (element.autofocus!==true) { //如果表单字段的html中已经有
        document.forms[0].elements[0].focus();
    }
})
```

)) 以上代码在这些情况下会出错： - 第一个字段为input且其type="hidden" - 第一个字段的CSS的display:none - 第一个字段的CSS的visibility:hidden

## H5新增属性：autofocus

支持autofocus的有Firefox4+、Safari5+、Chrome和Opera9.6

### (5)共有的表单字段事件：blur、change、focus

- blur:当字段失去焦点时触发
- change:注意！！对于input和textarea元素，在它们失去焦点且value值改变时触发；对于select元素，在其选项改变时触发。
- focus:当前字段获得焦点时触发

利用blur和focus事件可改变用户界面,利用change事件可以验证用户在字段中输入的数据。eg:

```
var textbox=document.forms[0].elements[0];

EventUtil.addHandler(textbox,"focus",function(event){
    event=EventUtil.getEvent(event);
    var target=EventUtil.getTarget(event);

    if (target.style.backgroundColor!="red") {
        target.style.backgroundColor="yellow";
    }
});

EventUtil.addHandler(textbox,"blur",function(event){
    event=EventUtil.getEvent(event);
    var target=EventUtil.getTarget(event);

    if (/^[^d]/.test(target.value)) { //如果含有非数值字符，则该字段在失去焦点时背景颜色变为红色
        target.style.setProperty("background-color","red");
    }
    else{
        target.style.removeProperty("background-color");
    }
});

EventUtil.addHandler(textbox,"change",function(event){
    event=EventUtil.getEvent(event);
    var target=EventUtil.getTarget(event);

    if (/^[^d]/.test(target.value)) { //如果输入了含有非数值字符，则该字段景颜色变为红色
        target.style.setProperty("background-color","red");
    }
    else{
        target.style.removeProperty("background-color");
    }
});
```