**第十三章表单**

**表单**

acceptCharset：服务器能处理的字符集，elements:表单所以元素，enctype：请求编码类型，length：表单控件数量，

reset（），submit（）

Document.forms可以取得页面中所以表单。可以通过数值引用或name取得特定表单。

**提交表单**

Input type=’submit’ button type=’submit’区别是一个单标签一个双标签，input type=’image’。

提交时会触发submit事件，可以通过preventDefault取消表单操作。

Js可以通过对form调用submit（）方法提交表单，此时不需要提交按钮，也不会触发submit事件。

第一次提交之后，用户可能不耐烦而反复点击提交按钮，此时会导致多次提交数据，导致服务器处理重复请求降低性能或者下了好几份订单。第一次提交之后就禁用提交按钮或者利用onsubmit事件取消后续提交操作dome1

**重置表单**

Input type=’reset’ button type=’reset’，

与上述不同，再跳用reset（）函数会像单击一样触发reset事件。

**表单字段**

Document.forms, form.elements能取到表单中控件的引用，能通过序列或者name属性来访问，有多个元素name相同则取得类数组

1. 共有表单字段

Disabled，name，readOnly，tabIndex，type，value，from（只读，对所属表单的指针）

Input.focus()获得焦点

避免多次提交表单 [dome1](dome1.html)

From.on(‘submit’,function(event){

Var target = event,target;//from

a.disabled = true

})

除了fieldset之外，所以表单字段都有type，除了input，其他的type值为

Select type=”select-one” select multiple type=”select-multiple”

Button(无定义type的时候) type=”submit” button type=”button”

button type=”reset” button type=”submit”

input和button的type可读可写，而select只可读

1. 共有的表单字段方法

每个表单都有focus()和blur()方法。

Type为hidden，display和visibility的表单使用focus()都会报错。

先将元素tabIndex属性设置为-1，然后可以调用focus()属性，

1. 共有的表单字段事件

Blur、change、focus

**文本框脚本（input和textarea）**

Input的size表示能显示的字符数，maxlength表示可以接受的最大字符数。

Textarea的cols表示行数，rows表示表示列数（size）无maxlength属性[dome2](dome2.html)

推荐使用value属性操作他们的值而不用setAttr或者操作textarea的第一个子节点，可能不会再dom中显示出来。

1. 选择文本

上述文本框都支持select（）方法，这个方法用于选择文本框中所以文本，调用时会聚焦到文本框。一般在文本框**获得焦点**的时候**选中**其中所以文本。

选择事件select

与secelt方法对应，是一个事件，在选择了文本框中的文本时触发。一般浏览器在选中文本且释放鼠标时触发，IE在选中第一个字时触发。除了webkit内核的浏览器，都会在调用select方法时触发

取得选择文本

非IE提供两个属性：selectionStart,selectionEnd,保存2个index因此可以通过

Text.value.substring(text.selectionStart,text.selectionEnd)

而IE中，有一个document.selection对象。首先得创建一个范围，然后再讲文字提取出来。Document.selection.createRange().text。不过无法确定是哪一部分的文本。

选择部分文本

没有针对这一操作的标准。非IE中所以的文本框都支持使用setSelectionRange()，传2个参数，第一个字符所以和选择的最后一个索引。类似substring()方法。Dome3

1. 过滤输出

屏蔽字符

使用keypress事件，正则以及String.fromCharCode(Code)方法能过滤输入。

用String.fromCharCode方法时要先判断charCode是否存在。需要注意，要屏蔽上、下、左、右、等键（9以上），以及ctrl是否按下（复制黏贴等操作）[dome4](dome4.html)

操作剪贴板

Beforecopy：在发生复制操作前 copy：在发生复制操作时 beforecut：在发生剪切操作前 cut：在发生剪切操作时 beforepaste：在发生粘贴操作前 paste：在发生粘贴操纵时

由于没有操作标准，这些时间各浏览器不同。Safair、chrome、firefox中，before事件都发生在显示菜单的情况下???。而ie则发生在copy等事件前。[Dome5](deme5.html)