目录

[新增属性 2](#_Toc430949826)

[多种input元素种类 3](#_Toc430949827)

[表单验证 3](#_Toc430949828)

[增强的页面元素 4](#_Toc430949829)

[文件API 5](#_Toc430949830)

[拖放API 7](#_Toc430949831)

## 新增属性

1.Form属性：属性值表单id。表单的从属元素从外指定到该表单

<form id="testform">

<input type="text">

</form>

<textarea form="testform"></textarea>

2.formaction和formmethod属性：所有的提交按钮增加不同的formaction和formmethod属性，可以将表单提交到不同的页面

<form action="s.asp">

<input type="submit" formaction="s1.asp" formmethod="post">

<input type="image" formaction="s2.asp" formmethod="get">

<button type="submit" formaction="s3.asp">

<input type="submit">

</form>

3.placeholder属性：当文本框处于未输入状态显示的输入提示

<form action="s.asp">

<input type="text" placeholder="在这里输入">

</form>

4.autofocus属性:给文本框、选择框、按钮控件加上属性时，打开页面自动获得光标焦点，一个页面只能有一个空间具有该属性谨慎使用

<form action="s.asp">

<input type="text" autofocus>

</form>

5.list属性：为单行文本框text增加list，属性值是datalist（类似select）元素的id。Datalist元素：用户想要的值不再列表内，允许自行输入，且不显示。当文本框获取焦点时已提示的方式显示通常用css(display：none)设为不显示

<form action="s.asp">

<input type="text" list="iiii" >

<datalist id="iiii" style="display:none;">

<option>fefeeee</option>

<option>fefeeee</option>

<option>fefeeee</option>

</datalist>

</form>

6.auotcomplete属性：自动完成功能，属性值：no，off，””。

## 多种input元素种类

1.<form action="s.asp">

<input type="url"><br>

<input type="email" required multiple>required:不允许输入为空，multiple：一串以逗号分割的email地址<br>

<input type="date" value="2010-6-5">

<input type="time">

<input type="datetime">

<input type="datetime-local">

<input type="month">

<input type="week">

<input type="number" min="10" max="100" step="5">专门要用来输入数字的文本框，在最小与最大的范围安装步幅增长

<input type="range" min="" max="" step="">输入一段范围内数字的文本框,进行拖动

<input type="search" style="-webkit-appearance:textfield;">用来搜索关键词的文本框

<input type="tel" pattern="">输入电话号码的文本框 pattern指定格式

<input type="color">颜色文本框

还可以使用样式对其编辑

</form>

2.output元素：必须从属表单，定义不同类型输出

<form action="s.asp">

<output onFormInput=""></output>

</form>

## 表单验证

1.自动验证：对元素使用属性的方法：required、pattern([0~9][A~Z]{3})正则表达式、min、max、step属性

2.显示验证：form元素与input元素都具有一个checkValidity方法，checkValidity方法以booleam的形式返回验证结果

<!DOCTYPE html>

<html lang="en" xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta charset="utf-8" />

<title></title>

<script language="javascript">

function check()

{

var email=document.getElementById("email");

if(email.value=="")

{

arler("addddd");

return false;

}

else if(!email.checkValidity())

alert("ddddd");

else

alert("ddddd");

}

</script>

</head>

<body>

<form id="testform" onSubmit="return check();" action="">

<label for="email">Email</label>

<input name="email" id="email" type="email"/><br/>

<input type="submit">

</form>

</body>

</html>

3.取消验证：form元素的novalidate属性。Input或submit元素的formnovalidate属性对单个元素失效，对submit也相当于对form元素使用novalidate

4.自定义错误信息：使用javascript调用各input元素的setCustomvalidaity方法来自定义错误信息：id.setCustomValidity(“ddfd”);

## 增强的页面元素

1.figure元素和figcaption元素：figure元素组合带有可选标题，代表独立内容，通常是图片、统计图、代码、音频插件、视频插件、统计表格。Figcaption表示figure的标题，从属figure元素必须在figure的内部

<body>

<figure>

<img>

<img>

<figcaption>标题</figcaption>

</figure>

</body>

2.details元素：将画面上布局区域进行展开或收放的方法

<details open>open：打开时将会打开

<summary>sss</summary>点击summary中的文字将打开其他部分

</details>

3.mark元素：页面中需要高亮突出显示的与原文作者无关。用于关键词和em、strong有区别

<mark>ddddddd</mark>

4.progress元素：完成进度

<progress value="" max=""></progress>

5.meter元素：规定范围内的数量值

<meter value="" min="" max="" low="" high="" optimum=""></meter>

6.menu与command元素：menu相当于菜单，command相当于菜单项

7.改良的ol列表：添加start开始编号和reversed方向排序属性。

<ol start="3" reversed></ol>

8.改良dl列表：定义术语

<dl>

<dt></dt>

<dd></dd>

<dt></dt>

<dd></dd>

</dl>

9.严格限制的cite元素：表示作品的标题

<cite>ddddd</cite>

10.重新定义的small元素：

## 文件API

1.File对象和filelist对象：filelist对象是file对象的集合。File包括name、lastModifiedDate、accept(类型控制)和multiple属性

<head>

<meta charset="utf-8" />

<title></title>

<script>

var file=getElementById("file").files[i];

file.name

</script>

</head>

<body>

<input id="file" type="file" multiple size="80" accept=”image/\*”/>

2.Blob对象：属性包括size、type(多为图像文件)。File也继承类Blob对象

File.size和file.type

<script>

var file;

file=document.getElementById("file").files[0];

file.innerHTML=file.size;

file.innerHTML=file.type;

</script>

3.filereader接口：

①readAsBinaryString方法：接受一个file参数以二进制读取

②readAsText方法：2个参数：file、encoding(默认是utf-8)

③readAsDataURL方法：file参数，以URL格式直接读入页面

④abort方法：无参数，中断读取操作

⑤onabort事件：数据读取中断时触发

⑥onerror事件：数据读取错误是触发

⑦onloadstart事件：开始读取数据

⑧onprogress事件：读取中

⑨onload事件：读取成功完成

⑩onloadend事件：读取完成，无论失败还是成功

<script language="javascript">

var result=document.getElementById("result");

var file=document.getElementById("file");

var reader=new FileReader();

if(typeof FileReader=='undefined')

{

result.innerHTML="<p>你的浏览器不支持filereader接口</p>";

file.setAttribute('disabled','disabled');

}

function readAsDataURL()

{

if(!/image\/\w+/.test(file.files[0]).type)

{

alert("请确保文件是图像文件");

return false;

}

reader.readAsDataURL(file);

reader.onload=function(e)

{

result.innerHTML='<ima src="'+this.result+'" alt="" />';

}

}

function readAsBinaryString()

{

reader.readAsBinaryString(file);

reader.onload=function(f)

{

result.innerHTML=this.result;

}

}

function readAsText()

{

reader.readAsText(file);

reader.onload=function(f)

{

result.innerHTML=this.result;

}

}

</script>

<p>

<label>请选择一个文件</label>

<input type="file" id="file" multiple /><br>

<input type="button" value="读取图像" onClick="readAsDataURL()" /><br>

<input type="button" value="读取二进制" onClick="readAsBinaryString()" /><br>

<input type="button" value="读取文本" onClick="readAsText()" /><br>

</p>

<div name="result" id="result">

</div>

## 拖放API

1.被拖放的对象元素的draggable属性设为true，img和a（必须指定href）默认允许拖

Draggable=”true”。

为了所有支持拖放API，需要指定“-webkit-user-drat:element”css属性。

2.拖放事件

①dragstart事件：拖动的数据存入DataTransfer对象，可以被设置为拖动事件对象的dataTransfer属性。setData()方法的第一个参数为携带数据的数据种类的字符串（只能填写类似：“text/plain”或”text/html”），第二个为携带的数据。

②drag事件：拖放过程

③dragenter事件：拖放过程中鼠标经过的元素，开始进入本元素范围内

④dragover事件：拖放过程中鼠标经过的元素，正在本元素范围内

⑤dragleave事件：拖放过程中鼠标经过的元素，离开本元素范围内

⑥drop事件：被拖放的目标元素，有其他元素被拖放到了本元素中

⑦dragend事件:拖放结束

<head>

<meta charset="utf-8" />

<title></title>

<script type="text/ecmascript">

function init()

{

var source=document.getElementById("dragme");

var dest=document.getElementById("text");

source.addEventListener("dragstart",function(ev)

{

var dt=ev.dataTransfer;

dt.effectAllowed='all';

dt.setData("text/plain","你好");

},false);

dest.addEventListener("dragend",function(ev)

{

ev.preventDefault();

},false);

dest.addEventListener("drop",function(ev)

{

var dt=ev.dataTransfer;

var text=dt.getData("text/plain");

dest.textContent+=text;

ev.preventDefault();

ev.stopPropagation();

},false);

}

document.ondragover=function(e){e.preventDefault();};

document.ondrop=function(e){e.preventDefault();};

</script>

</head>

<body onLoad="init()">

<div id="dragme" draggable="true" style="width:200px; border:1px solid gray;">

请拖放

</div>

<div id="text" style="width:200px; height:200px; border:1px solid gray;"></div>

</body>

3.DataTransfer对象的属性与方法：

①dropEffect属性：拖放操作效果，允许指定值为：none、copy、ink、move。一般在ondragover事件中设定

②effectAllowed属性：被拖放的视觉效果，指定值：none、copy、copyLink、copyMove、link、linkMove、move、all、unintialize。一般在ondragstart事件中设定。dropEffect与effectAllowed效果值必须完全相等

③types属性:存入数据种类，字符串的伪数组

④void clearData(DOMString format) 方法：清除DaetaTransfer对象中的数据，如果省略format参数，则全部清除

⑤void setData() 方法：存入数据

⑥DOMString getData(DOMString,format)：读数据

⑦void setDragImage(Elment image,long x,long y)：用img元素设置拖放图标。X为拖放图标离鼠标指针的x轴方向位移量