前端开发技术之

JavaScript事件处理程序

主要内容

- 事件与事件处理概述
- DOM事件模型
- 键盘鼠标事件
- 表单事件

事件

- **事件(Event)**:就是用户与Web页面交互时产生的操作。比如按下鼠标、移动窗口、选择菜单等。
- 事件处理:是指程序对事件作出的响应。
- 事件处理中涉及的程序称为事件处理程序。事件处理程序通常定义为函数。
- 在Web页面中产生事件的对象，称为事件目标。在不同事件目标上可以产生不同类型的事件。
- 应用程序通过指明事件类型和事件目标，在Web浏览器中注册它们的事件处理程序。当在特定的目标上发生特定类型的事件时，浏览器会调用对应的处理程序。

JavaScript 常用事件

参见教材P250-P252

事件		描述
鼠标事件	click	用户单击鼠标时触发此事件
	dblclick	用户双击鼠标时触发此事件
	mousedown	用户按下鼠标时触发此事件
	mouseup	用户按下鼠标后松开鼠标时触发此事件
	mouseover	当用户将鼠标移动到某对象范围的上方时触发此事件
	mousemove	用户移动时鼠标触发此事件
	mouseout	当用户鼠标离开某对象范围时触发此事件
键盘事件	keypress	当用户键盘上的某个字符键被按下并且释放时触发此事件
	keydown	当用户键盘上某个按键被按下时触发此事件
	keyup	当用户键盘上某个按键被按下后松开时触发此事件

JavaScript 常用事件

事　　件		描　　述
窗口事件	abort	当图形尚未完全加载前，用户就单击了一个超链接，或单击停止按钮时触发此事件
	error	加载文件或图像发生错误时触发此事件
	load	页面内容加载完成时触发此事件
	resize	当浏览器的窗口大小被改变时触发此事件
	unload	当前页面关闭或退出时触发此事件
表单事件	blur	当前表单元素失去焦点时触发此事件
	click	用户单击复选框、单选按钮或button、submit和reset按钮时触发此事件
	change	表单元素的内容发生改变并且元素失去焦点时触发此事件
	focus	当表单元素获得焦点时触发此事件
	reset	用户单击表单上的reset按钮时会触发此事件
	select	用户选择了一个input或textarea表单域中的文本时触发此事件
	submit	用户单击submit按钮提交表单时触发此事件

DOM事件模型

- 事件流
  - 描述的是从页面中接收事件的顺序。

--------事件冒泡流

--------事件捕获流

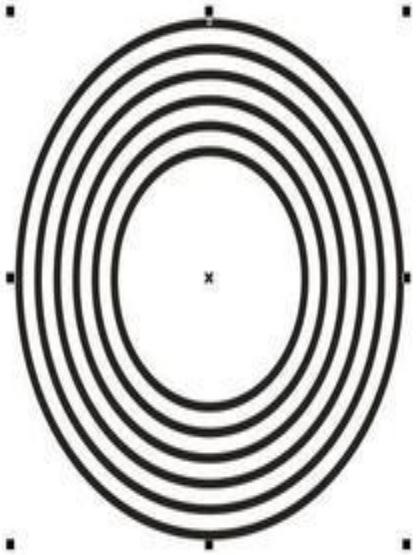
DOM事件模型

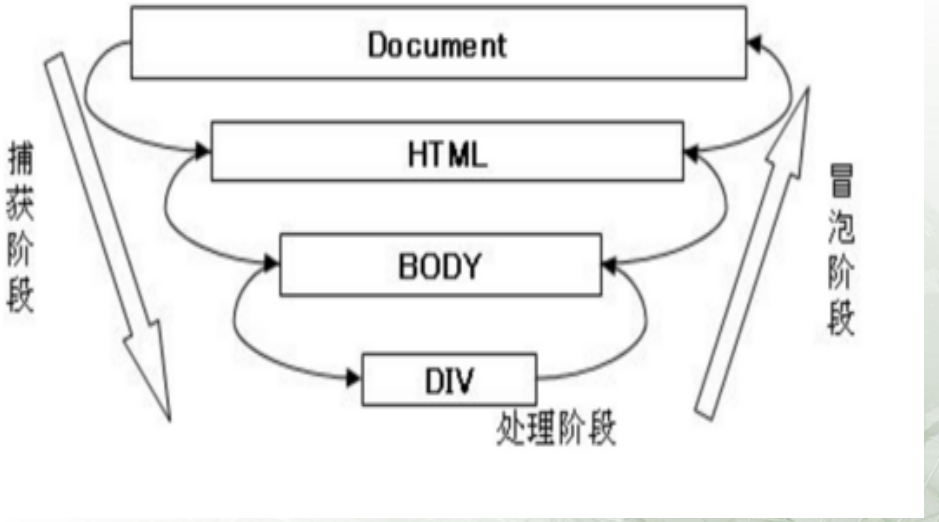
事件流

事件冒泡: 即事件最开始由最具体的元素（文档中嵌套层次最深的那个节点）接收，然后逐级向上传播至最不具体的那个节点（文档）。

事件捕获: 不太具体的节点应该更早接收到事件，而最具体的节点最后接收到事件。

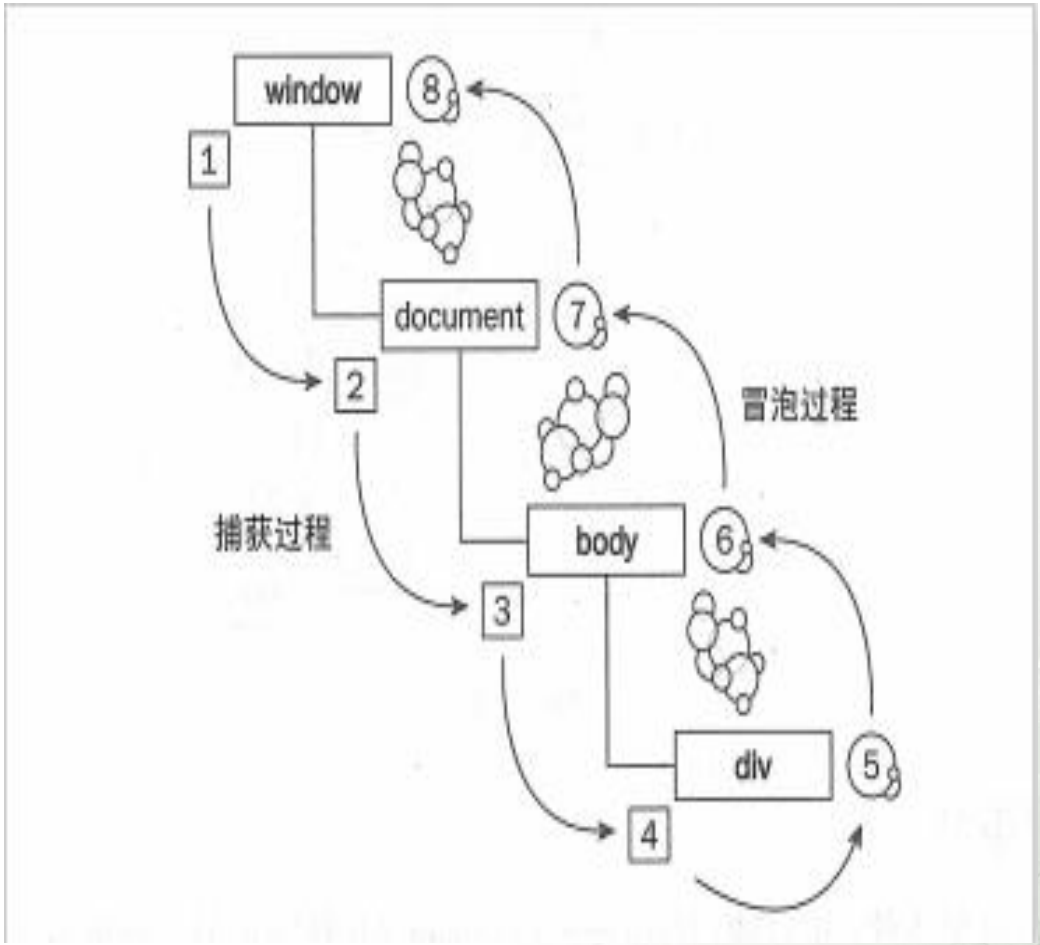
事件流





事件流

事件捕获与事件冒泡



代码说明

代码说明2

事件处理程序

- 为了使浏览器在事件发生时，能自动调用相应的事件处理程序处理事件，需要对事件目标注册事件处理程序（也称事件绑定）。

事件处理程序

- JavaScript 事件处理程序就是一组语句，在事件（如点击鼠标或移动鼠标等）发生时执行 。

HTML事件处理程序

事件处理程序的基本语法可以是：

事件名=“ JavaScript 代码或调用函数”

例如：

<input type=”button” … onclick=“alert(“单击我！ ”);”>

<input type=”button” … onmousedown=“check( )”>

表示鼠标按下时，将调用执行函数check( ) 。

事件处理程序

- JavaScript 事件处理程序就是一组语句，在事件（如点击鼠标或移动鼠标等）发生时执行 。

事件处理程序调用2：
对象.事件名=函数名　　JavaScript事件处理程序

onmousedown=check;
function check(){alert(“单击我！”)}
或者：
onmousedown=function(){alert(“单击我！”);};

例如：
<input id=“btn1” type=”button” >
表示鼠标按下时，将调用执行函数check( ) 。

事件处理程序

- 注册事件处理程序有以下三种方式：
  - 设置HTML标签的事件属性为事件处理程序。
  - 设置事件目标的事件属性为事件处理函数。
  - 使用事件目标调用addEventListener()方法。

- **注：第一和第二种注册方式中的事件属性名的组成形式是：“on”+事件名，例如onclick、onfocus等等。**

事件处理程序

方式1.设置HTML标签的事件属性为事件处理程序

- 通过设置HTML标签的事件属性为事件处理程序的方式来注册事件处理程序时，事件属性中的脚本代码不能包含函数声明，但可以是函数调用或一系列使用分号分隔的脚本代码。

示例

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>HTML标签的事件属性为函数调用</title>
<script>
function printName(){
    var name="张三";
    alert(name);
}
</script>
</head>
<body>
  <form action="">
    <input type="button" onClick="printName()" value="测试"/>
  </form>
</body>
</html>
```

事件处理程序

方式2.设置事件目标的事件属性为事件处理函数

- 使HTML和JavaScript分离的最简单的注册事件处理程序的方式就是通过设置事件目标的事件属性为所需事件处理程序函数。这也是最常用的事件处理程序注册方式。

示例

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>设置事件目标的事件属性为事件处理函数</title>
<script type="text/javascript">
window.onload=function(){
    var oTxt=document.getElementById("username");
    oTxt.onfocus=function(){
        alert("请输入用户名");
    };
    oTxt.onblur=function(){
        alert("您输入的用户名为: "+this.value);
    };
};
</script>
</head>
<body>
  <form action="">
    <input type="text" name="username" id="username"/>
  </form>
</body>
</html>
```

事件处理程序

方式3.使用事件目标调用addEventListener()方法：

addEventListener()

removeEventListner()

接收3个参数

要处理的事件名

作为事件处理程序的函数

布尔值

事件处理程序

方式3.使用事件目标调用addEventListener()方法：

- **在标准事件模型中，任何能成为事件目标的对象都定义了addEventListener()方法，使用这个方法可以为事件目标注册事件处理程序。**
- **语法:**

addEventListner(事件类型名,事件处理函数,false);

- 说明：

第一个参数为事件类型名，例如：load、click等事件名；第二个参数必须为函数，其中，可以是函数的声明代码，也可以是函数调用语句；第三个参数为布尔值，通常为false。

示例

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>使用addEventListener()注册事件处理程序</title>
<script>
var p;
function mouseOverFn(){
    p.style.color="blue";
    p.style.fontSize="30px";
    p.style.fontStyle="italic";
    p.style.textDecoration="underline";
}
function mouseOutFn(){
    p.style.color="red";
    p.style.fontStyle="normal";
    p.style.textDecoration="none";
}
window.onload=function(){
    var p=document.getElementById("p");
     //click事件处理程序直接使用函数的声明
    p.addEventListener("click",function(){alert("单击事件");},false);
    p.addEventListener("mouseover",mouseOverFn,false);//事件处理程序使用函数调用语句
    p.addEventListener("mouseout",mouseOutFn,false);
    p.addEventListener("click",function(){alert("使用addEventListener()注册事件处理程序");},false);
}
</script>
</head>
<body>
<p id="p">Hello World!</p>
</body>
</html>
```

使用addEventListener()可以为同一个对象注册不同类型的事件处理程序，也可以为同一事件类型的多个处理程序函数。当对象上发生事件时，所有该事件类型的注册处理程序就会按照注册的顺序调用。

三种事件处理程序举例

```
<div id="box">
    <input type="button" value="按钮" id="btn" onclick="showMes()">
    <input type="button" value="按钮2" id="btn2">
    <input type="button" value="按钮3" id="btn3">
</div>
<script>
    function showMes(){
        alert('hello world!');
    }
    var btn2=document.getElementById('btn2');
    var btn3=document.getElementById('btn3');
    btn2.onclick=function(){
        alert('这是通过DOM0级添加的事件！ ');
    }
    btn2.onclick=null;
    // DOM2级事件
    btn3.addEventListener('click',showMes,false);
    btn3.addEventListener('click',function(){
        alert(this);
    },false);
    // 删除事件
    //btn3.removeEventListener('click',showMes,false);
```

源代码

鼠标键盘事件

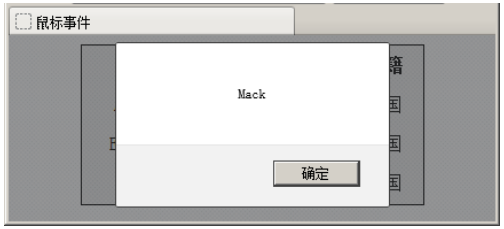
- 鼠标事件
  - 鼠标事件是指通过鼠标动作触发的事件，鼠标事件有很多，下面列举几个常用的鼠标事件，如下表所示。

类别	事件	事件说明
鼠标事件	onclick	鼠标单击时触发此事件
	ondblclick	鼠标双击时触发此事件
	onmousedown	鼠标按下时触发此事件
	onmouseup	鼠标弹起时触发的事件
	onmouseover	鼠标移动到某个设置了此事件的元素上时触发此事件
	onmousemove	鼠标移动时触发此事件
	onmouseout	鼠标从某个设置了此事件的元素上离开时触发此事件

鼠标事件举例

onmouseover事件被触发

onmouseout事件被触发



onclick事件被触发

源代码：
event_mouse

鼠标键盘事件

- 键盘事件
  - 键盘事件是指通过键盘动作触发的事件，常用于检查用户向页面输入的内容。下面列举几个常用的键盘事件，如下表所示。

类别	事件	事件说明
键盘事件	onkeydown	当键盘上的某个按键被按下时触发此事件
	onkeyup	当键盘上的某个按键被按下后弹起时触发此事件
	onkeypress	当输入有效的字符按键时触发此事件

事件与事件名称

事件（event）

事件：就是Web浏览器通知应用程序发生了什么事情。

事件类型：是一个用来说明发生什么类型事件的字符串。
“mousemove”、“keydown”、“load”

事件目标：是发生的事件与之相关的对象。

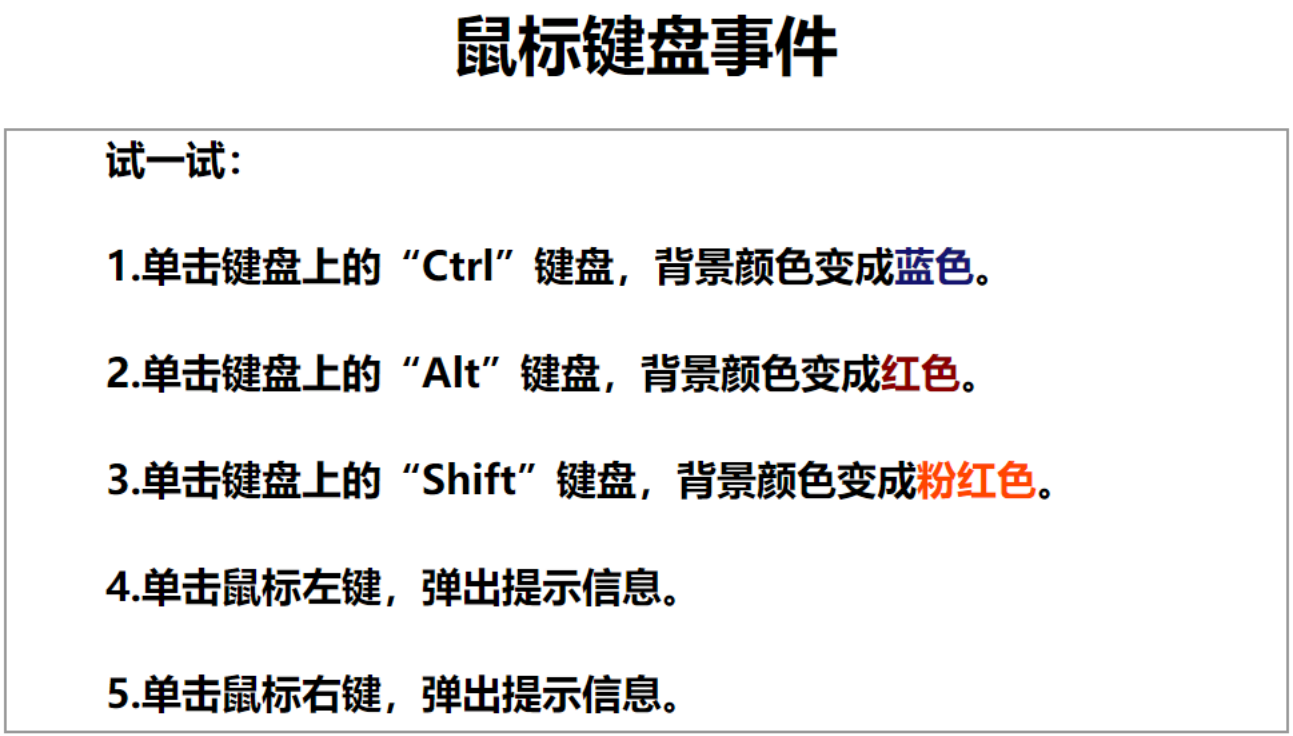
事件对象Event

事件对象（Event对象）代表事件的状态，例如触发event对象的元素、鼠标的位置及状态、按下的键等等。下表中列举了以下常用的按键和相应的键码值，具体如下：

按键	键码	按键	键码	按键	键码	按键	键码
A	65	J	74	S	83	1	49
B	66	K	75	T	84	2	50
C	67	L	76	U	85	3	51
D	68	M	77	V	86	4	52
E	69	N	78	W	87	5	53
F	70	O	79	X	88	6	54
G	71	P	80	Y	89	7	55
H	72	Q	81	Z	90	8	56
I	73	R	82	0	48	9	57
Backspace	8	Esc	27	right	39	-_	189
Tab	9	Spacebar	32	down	40	.>	190
Clear	12	Page up	33	Insert	45	/?	191
Enter	13	Page down	34	Delete	46	`~	192
Shift	16	End	35	Num Lock	144	[{	219
Ctrl	17	Home	36	:;	186	|	220
Alt	18	left	37	=+	187	]}	221
Caps Lock	20	up	38	,<	188	‘”	222

键盘事件举例

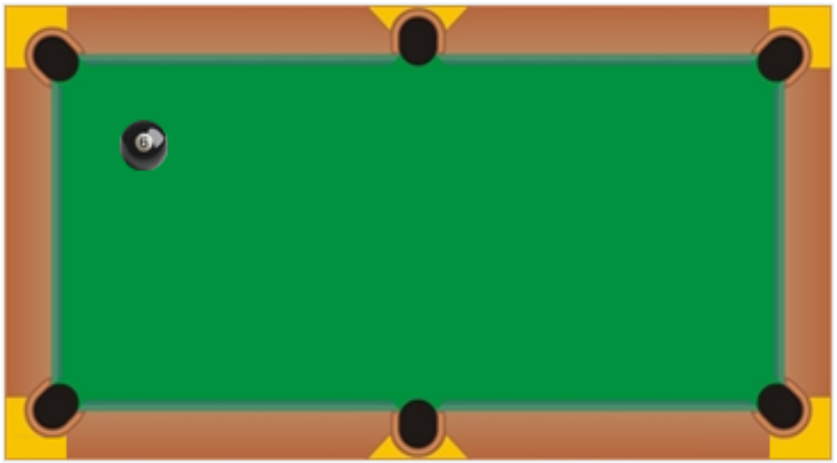
例：测试鼠标和键盘事件，效果图如下



查看源文件

实验

- 台球移动小游戏



源码

页面事件

- 页面事件是指通过页面触发的事件，下面列举几个常用的页面事件，如下表所示。

类别	事件	事件说明
页面事件	onload	当页面加载完成时触发此事件
	onunload	当页面卸载时触发此事件

表单事件

表单事件	**blur**	**当前表单元素失去焦点时触发此事件**
	click	**用户单击复选框、单选按钮或**button、submit**和**reset**按钮时触发此事件**
	change	**表单元素的内容发生改变并且元素失去焦点时触发此事件**
	focus	**当表单元素获得焦点时触发此事件**
	reset	**用户单击表单上的**reset**按钮时会触发此事件**
	select	**用户选择了一个**input**或**textarea**表单域中的文本时触发此事件**
	submit	**用户单击**submit**按钮提交表单时触发此事件**

文本框对象

- 文本框元素用于在表单中输入字、词或一系列数字
- 可以通过将 HTML 的 INPUT 标签中的 type 设置为 “text”，以创建文本框元素

```
<INPUT type="text" id="t1">
```

文本框对象 – 事件处理程序

文本框	事件	**onBlur**	文本框失去焦点
		onChange	文本框的值被修改
		onFocus	光标进入文本框中
	方法	**focus( )**	获得焦点，即获得鼠标光标
		select( )	选中文本内容，突出显示输入区域
	属性	**readonly**	只读，文本框中的内容不能修改

```
<SCRIPT Text= text/javascript >
function  clearText( )
{
  var a=document.getElementById("card");
   if (a.value=="输入您的会员帐号")
      a.value="" ;
}
function check( )
{
  var a=document.getElementById("card");
  if (a. value.substr(0,2)!="10" || isNaN(a. value))
  {
    alert("格式错误，请重新输入") ;
    a.focus( );
    a.select( );
  }
}
function compute( )
{
   var price= document.getElementById("price").value;
   var number= document.getElementById("number").value ;
   var tot =document.getElementById("total");
   tot.value= price*number;
```

onfocus事件调用的函数**clearText**（）清空帐号文本框中的内容

onblur事件调用的函数check（）检查输入的帐号是否是“10”打头，并且是数字

onchange事件调用的函数compute( )用来计算总价

文本框对象

```
<BODY>
<FORM name="myform">
    .......
<TD>帐号：
<INPUT id="card" onfocus="clearText( )“   onblur="check( )"
       type = text  value="输入您的会员帐号“></TD>
    .......
<TD>单价：
<INPUT id="price" type = text value="25.00” readonly >￥</TD>
    .......
<TD>数量：
 <INPUT  id="number" onchange="compute( )“ type = text >个</TD>
<TD>总价：
 <INPUT  id="total" type = text value="0.00" >￥</TD>
    .......
</BODY>
```

帐号文本框添加onfocus和onblur焦点事件

价格只读属性

数量文本框添加onchange事件

命令按钮对象

- 命令按钮对象是网页中最常用的元素之一

```
<INPUT type="submit" name="button1"  value="提交">
<INPUT type="reset" name="button2"  value="重置">
<INPUT type="button" name="button3"  value="计算">
```

“按钮 – 事件处理程序

表单元素	事件处理程序	说明
命令按钮	**onsubmit**	表单提交事件，单击“提交”按钮时产生，此事件属于<FORM>元素，不属于提交按钮
	onclick	按钮单击事件

onSubmit事件处理代码：

<FORM onsubmit=”return 调用函数名” >…</FORM>

如果函数返回true，则向远程服务器提交表单；

如果函数返回false，则取消提交。

```
function check( )
{
 var userName=document.getElementById(“userName”);
 var pass1= document.getElementById(“pass1”);
 var pass2= document.getElementById(“pass2”);
 if (pass1.value==pass2.value)
 {
    if (pass1.value.length!=0)
   {
  else
  {
    alert("密码不能为空！\n请输入密码");
    return false;
   }
 }
 else
 {
    alert("确认码必须和输入的密码相同！");
    return false;
 }
}
```

onSubmit事件调用的函数：输入数据检查

```
<BODY>
<FORM  id="myform" onsubmit="return check( )"  >
```

如果输入格式正确，返回true，提交表单信息；
如果格式错误，返回false，取消提交，提醒用户重填

复选框对象

- 当用户需要在选项列表中选择多项时，可以使用复选框对象
- 要创建复选框对象，请使用 <INPUT> 标签

```
请选择您的爱好
<INPUT type="checkbox" value="电影"> 电影
<INPUT type="checkbox" value="电影"> 电影
```

复选框 – 事件处理程序

复选框 .	事件	**onBlur**	复选框失去焦点
		onFocus	复选框获得焦点
		onClick	复选框被选定或取消选定
	属性	**checked**	复选框是否被选中，选中为true，未选中为false。您可以使用此属性查看复选框的状态或设置复选框是否被选中
		value	设置或获取复选框的值

单选按钮对象

- 当用户只需要从选项列表中选择一个选项时，可以使用单选按钮对象
- 要创建单选按钮对象，请使用 <INPUT> 标签

```
<INPUT type=“radio” value=“M”>男
<INPUT type=“radio” value=“F”>女
```

单选按钮 - 事件和属性

单选按钮.	事件	**onBlur**	单选按钮失去焦点
		onFocus	单选按钮获得焦点
		onClick	单选按钮被选定或取消选定
	属性	**checked**	单选按钮是否被选中，选中为true，未选中为false。您可以使用此属性查看单选按钮的状态或设置单选按钮是否被选中
		value	设置或获取单选按钮的值

表单验证 2-1

- JavaScript 最常见的用法之一就是验证表单
- 对于检查用户输入是否存在错误和是否疏漏了必选项，JavaScript 是一种十分便捷的方法

正则表达式

- 字符串操作
  - Search
  - Substring
  - charAt
  - split

正则表达式

- 什么是正则表达式？
  - 什么叫“正则”
    - 规则、模式
  - 强大的字符串匹配工具
  - 是一种正常人类很难读懂的文字
  - RegExp对象
    - JS风格——new RegExp(“a”, “i”)
    - perl风格——/a/i

正则表达式

- search
  - 字符串搜索
    - 返回出现的位置
    - 忽略大小写：i——ignore
    - 判断浏览器类型
- match
  - 获取匹配的项目
    - 量词：+
    - 量词变化：\d、\d\d和\d+
    - 全局匹配：g——global
    - 例子：找出所有数字

正则表达式

- replace
  - 替换所有匹配
    - 返回替换后的字符串
    - 例子：敏感词过滤

正则表达式

字符串：

- 任意字符
  - [abc]
    - 例子：o[usb]t——obt、ost、out
- 范围
  - [a-z]、[0-9]
    - 例子：id[0-9]——id0、id5
- 排除
  - [^a]
    - 例子：o[^0-9]t——oat、o?t、o t

正则表达式

- 组合
  - [a-z0-9A-Z]
- 实例：过滤HTML标签
  - 自定义innerText方法
- 转义字符
  - .（点）——任意字符
  - \d（数字[0-9]）、\w(英文数字下划线)、\s （空白字符）
  - \D、\W、\S

正则表达式

- 什么是量词
  - 出现的次数
  - {n,m}，至少出现n次，最多m次
- 常用量词
  - {n,}　　至少n次
  - *　　任意次 {0,}
  - ?　　零次或一次　　{0,1}
  - +　　一次或任意次{1,}
  - {n}　　正好n次

利用正则表达式进行模式匹配

预定义字符类

\d	[0-9]	a digit
\D	[^0-9]	not a digit
\w	[A-Za-z_0-9]	a word character
\W	[^A-Za-z_0-9]	not a word character
\s	[\r\t\n\f]	a whitespace character
\S	[^ \r\t\n\f]	not a whitespace character

利用正则表达式进行模式匹配

- 例：

  [abcd]　匹配“a”、“b”、“c”、”d”

  [a-z]　a-z中所有小写字母

  \d*　0或者多个数字

  \d+　1或者多个数字

  \d?　0或者1个数字

例：

/^Lee/

/Lee Ann$/　可匹配Mary Lee Ann

“Mary Lee Ann is nice”?

正则表达式	说明
^[0-9]*$	数字
^\d{n}$	n位的数字
^\d{n,}$	至少n位的数字
^\d{m,n}$	m-n位的数字
^(0|[1-9][0-9]*)$	零和非零开头的数字
^([1-9][0-9]*)+(.[0-9]{1,2})?$	非零开头的最多带两位小数的数字
^(\-|\+)?\d+(\.\d+)?$	正数、负数、和小数
^\d+$ 或 ^[1-9]\d*|0$	非负整数
^-[1-9]\d*|0$ 或 ^((-\d+)|(0+))$	非正整数
^[\u4e00-\u9fa5]{0,}$	汉字
^[A-Za-z0-9]+$ 或 ^[A-Za-z0-9]{4,40}$	英文和数字
^[A-Za-z]+$	由26个英文字母组成的字符串
^[A-Za-z0-9]+$	由数字和26个英文字母组成的字符串
^\w+$ 或 ^\w{3,20}$	由数字、26个英文字母或者下划线组成的字符串
^[\u4E00-\u9FA5A-Za-z0-9_]+$	中文、英文、数字包括下划线
^\w+([-+.]\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*$	Email地址
[a-zA-z]+://[^\s]* 或 ^http://([\w-]+\.)+[\w-]+(/[\w-./?%&=]*)?$	URL地址
^\d{15}|\d{18}$	身份证号(15位、18位数字)
^([0-9]){7,18}(x|X)?$ 或 ^\d{8,18}|[0-9x]{8,18}|[0-9X]{8,18}?$	以数字、字母x结尾的短身份证号码
^[a-zA-Z][a-zA-Z0-9_]{4,15}$	帐号是否合法(字母开头，允许5-16字节，允许字母数字下划线)
^[a-zA-Z]\w{5,17}$	密码(以字母开头，长度在6~18之间，只能包含字母、数字和下划线)

HTML5的input元素

pattern属性

pattern属性用于验证input类型输入框中，用户输入的内容是否与所定义的正则表达式相匹配。pattern属性适用于的类型是：text、search、url、tel、email和password的<input/>标记。

常用的正则表达式如下表所示。

练习

- H5中input的pattern属性完成以下的表单验证。

•pattern属性的应用

总结

- 掌握两种事件流模型
  - 冒泡型
  - 捕获型
- 注册事件处理程序的三种表达方式
  - Html方式
  - Javascript方式
  - addEventListener (事件名，函数名，false)方法
- 键盘鼠标事件
- Event事件
- 表单事件：

总结

- 表单事件：
  - onblur、onchange 和 onfocus 是一些与表单对象相关的事件处理程序
  - 在浏览器窗口中，如果文本框获得焦点，则会调用 onfocus 事件处理程序
  - 当对象失去焦点或光标退出对象时，将执行 onblur 事件处理程序
  - 当修改文本框内容或改写下拉列表框的选项时，则会调用 onchange 事件处理程序
- 正则表达式
  - 表单验证可结合正则表达式