

一，js的事件

事件

事件源

响应行为

1，js的常用事件

onclick: 点击事件

onchange: 域内容被改变的事件

需求：实现二级联动：

```
<select id="city">
    <option value="bj">北京</option>
    <option value="sh">上海</option>
    <option value="tj">天津</option>
    <option value="cd">成都</option>
</select>
<select id="area">
    <option value="bj">海淀</option>
    <option value="bj">朝阳</option>
    <option value="bj">东城</option>
    <option value="bj">昌平</option>
</select>
</body>
<script type="text/javascript">
    var select = document.getElementById("city");
    select.onchange = function()
    {
        var optional = select.value;
        switch(optional)
        {
            case 'bj':
                var area = document.getElementById("area");
                area.innerHTML = "<option>海淀</option><option>
东城</option><option>昌平</option><option>朝阳</option><option>学院</option>";
                break;
            case 'tj':
```

```

        var area = document.getElementById("area");
        area.innerHTML = "<option>南开</option><option>
西青</option><option>河西</option>";
        break;
        case 'cd':
            var area =
document.getElementById("area");
            area.innerHTML = "<option>南开</option><option>
西青</option><option>河西</option>";
            break;
            case 'sh':
                var area =
document.getElementById("area");
                area.innerHTML = "<option>浦东</option><option>
清华大街</option><option>北城</option>";
                break;
                default : alert('wrong!');
            }

    };

</script>

```

onfocus: 获得焦点的事件

需求: 当输入框获得焦点的时候, 提示输入的内容格式

当输入框失去焦点的时候, 提示输入有误

```

<label for = "txt">name</label>
<input type = "text" id = "txt"/><span id = "action" style = ""></span>
</body>
<script type = "text/javascript">
var txt = document.getElementById("txt");
txt.onfocus = function ()
{
//显示
var span = document.getElementById("action");

```

```

span.innerHTML = "用户名格式最小8位";
span.style.color = "green";
};
txt.onblur = function()
{
//错误提示
    var span = document.getElementById("action");
    span.innerHTML = "对不起，输入错误！";
    span.style.color = "red";
};

```

onblur: 失去焦点的事件

onmouseover: 鼠标悬浮事件

需求: div元素, 鼠标移入变为绿色,

鼠标移开变为蓝色

```

<style type="text/css">
#d1 { background-color:red;width:200px;height :200px;}
</style>
</head>

<body>
    <div id = "d1">

    </div>
</body>
<script type = "text/javascript">
var div = document.getElementById("d1");
div.onmousemove = function ()
{
div.style.backgroundColor = "red";
//alert("xx");
};
div.onmouseout = function ()
{
div.style.backgroundColor = "yellow";

```

```
//alert("yyy");  
};
```

```
</script>
```

onmouseout: 鼠标离开事件

onload: 加载完毕的事件

需求: 等到页面加载完毕再去执行onload事件所指向的函数

若习惯将js代码写在body之前的话, 需要将js的代码放在window.onload = function() {
//js代码 };

2, 事件的绑定方式

①将事件和响应行为内嵌到HTML标签中

```
<input type = "button" value = "button" onclick = "alert('xxx')"/>
```

②将事件内嵌到HTML中而响应封装在js的函数中

```
<input type = "button" value = "button" onclick = "fun()"/>
```

```
<script type = "text/javascript">
```

```
function fun()
```

```
{
```

```
alert ("yyy");
```

```
}
```

```
</script>
```

③将事件的绑定和响应完全分开

```
<input id = "btn" type = "button" value = "button"/>
```

```
<script type = "text/javascript">
```

```
var btn = document.getElementById("id");
```

```
btn.onclick = function ()
```

```
{
```

```
alert("zzz");
```

```
};
```

```
</script>
```

this关键字: :

this经过事件的函数进行传递的是HTML标签对象

```
<input type = "button" value = "button" onclick = "fun(this)"/>
```

```
<script type = "text/javascript">
function fun(obj)
{
alert (obj.type); //打印出button
}
</script>
```

3, 阻止事件的默认行为

IE: window.event.returnValue = false;

W3C (火狐): 传递过来的事件对象.preventDefault ()

或者直接在跳转标签的onclick写为: onclick = "return false" 也可以阻止事件的默认行为

对于js中的方法, 如果在引用的时候使用括号的话, 表示调用这个方法, 如果不使用括号的话, 表示代表这个函数本身

4, 阻止事件的传播

ie: window.event.cancelBubble = false;

w3c: 传递过来的事件对象.stopPropagation();

二, js的bom

1, Windows对象

①

弹框的方法:

提示框: alert ("提示信息"); 没有返回值

确认框: confirm ("确认信息"); 有返回值: 如果点击确定返回true, 如果点击取消返回false

输入框: prompt ("提示信息"); 有返回值: 如果点击确定将输入框中的内容返回, 如果点击取消返

回null

②

open方法:

window.open ("URL地址");

③

定时器

```
setTimeout (函数值, 毫秒值);  
    setTimeout (  
        function ()  
        {  
            alert("xxxx");  
        }, 2000  
    );  
clearTimeout(定时器名称);表示清除定时器  
setInterval (函数, 毫秒值);  
clearInterval (定时器的名称);
```

需求：注册后5秒跳转首页

```
<script type="text/javascript">  
var time = 5;  
setInterval(function() {  
var second = document.getElementById("second");  
if(time>=1)  
{  
second.innerHTML = time;  
time = time - 1;  
}  
else  
{  
location.href = "demo3.jsp";  
}  
}, 1000);  
</script>  
</head>  
  
<body>  
    恭喜您注册成功! <span id = "second" style = "color:red;">5</span>秒后跳转到首  
页，若不跳转，请<a href = "demo3.jsp">点击这里</a>  
</body>  
2, location  
location.href = "url地址";
```

3, history

history.back(); 上一页

history.go(); 返回以及上下页

history.forward(); 下一页

Windows可以省略，但是history不能省略

三, js的dom

1, 理解一下文档对象模型

HTML文件加载到内存之后会形成一颗dom树，根据这些节点对象可以进行脚本代码的动态修改在dom树当中，一切皆为节点对象

2, dom方法和属性

optionnode.selected有返回值，若被选中，则返回true，否则返回false

①hasChildNodes() 该方法用来检查一个元素是否有子节点，返回值为true或者false

```
var booleanValue = element.hasChildNodes();
```

②var node = document.getElementById("元素的属性值");

```
var p = node.getAttributeNode("属性名");
```

alert(p.nodeName) 属性节点的节点名

alert(p.nodeType) 属性节点的节点类型

alert(p.nodeValue) 属性节点的节点值