

*nodeTech.*节点科技

前端岗前培训

第四课 JS基础

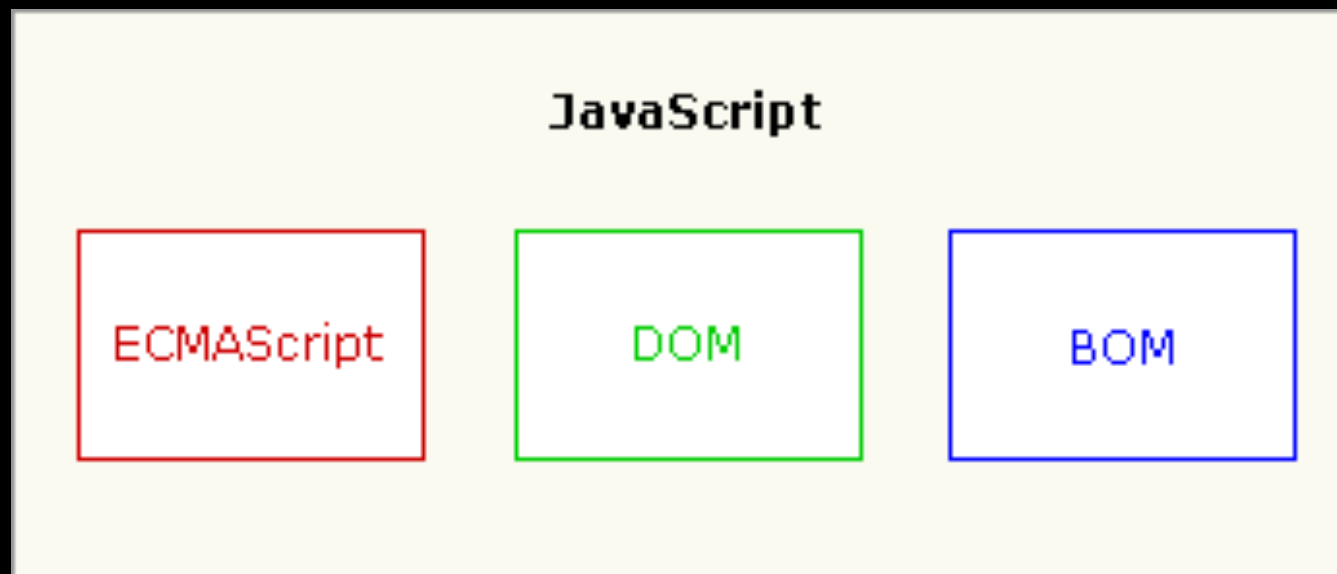
作业点评

- 设备的尺寸? 要根据手机尺寸来设计 <https://wingsren.github.io/3ke/rentingt/03/%E5%93%8D%E5%BA%94%E5%BC%8F%E5%B8%83%E5%B1%80/second.html>
- 虽然做了响应式 却没有调节好展示效果 <https://guicx.github.io/Gui/guichenxing/03/index.html>
- 设备尺寸默认尺寸可以不用强调, 后面的根据尺寸 依次向下定义 <https://hbuilderx.github.io/HellowGitHub/myHomeWork03/response.html> 否则可能出现意想不到的现象
- flex布局如何做响应式? <https://wingsren.github.io/3ke/rentingt/03/flex%E5%B8%83%E5%B1%80.html>
- 滚动页面注重留白
- 其实响应式和滚动效果可以做在一起
- https://yahoo-back.github.io/NodeTech/zhaoshasha/3/3-1/media_3/index.html 手机上看 宽度还是有点问题
- 优秀的 <https://github.com/smileCris/NodeTech/tree/master/0503work>



JS

- http://www.w3school.com.cn/js/index_pro.asp
- <https://www.hackerrank.com/domains/tutorials/10-days-of-javascript>



Window 对象
Navigator 对象
Screen 对象
History 对象
Location 对象



函数的返回类型

- 返回数字
- 返回布尔型 true / false
- 返回字符串
- 没有返回值
- 函数体内按顺序执行代码， 执行return代码后， 函数体结束， 将不会执行函数体内其他的代码



函数返回类型

返回数字
返回布尔型 true / false
返回字符串
没有返回值

```
function square(number1)
{
    var c = number1*number1;
    return c;
}

var x = 4;
var des = square(x);

document.write(" 4的平方是 : "+des);
```

返回数字



函数返回类型

返回数字

返回布尔型 true / false

返回字符串

没有返回值

```
function areYouStudent(me)
{
    if(me.work=="student"){
        return true;
    }else{
        return false;
    }
}

var lilu={name:"lilu",work:"developer"};
var shelly={name:"shelly",work:"student"};

if(areYouStudent(lilu)){ 可以用if直接判断
    document.write(lilu.name + "是学生");
}else{
    document.write(lilu.name + "不是学生");
}

document.write("<br/>");

if(areYouStudent(shelly)){ 可以用if直接判断
    document.write(shelly.name + "是学生");
}else{
    document.write(shelly.name + "不是学生");
}
```



函数返回类型

返回数字
返回布尔型 true / false
返回字符串
没有返回值

```
function whoAreYou(me)    返回一个字符串
{
    return "I'm "+me.name+". I'm a "+me.work+".";
}
```

```
var lilu={name:"lilu",work:"developer"};
var shelly={name:"shelly",work:"student"};
document.write(whoAreYou(lilu));    打印字符串
document.write("<br/>");
document.write(whoAreYou(shelly));
```



函数返回类型

返回数字
返回布尔型 true / false
返回字符串
没有返回值

```
function justDoIt()  
{  
  document.write("Just Do It!");  
}  
justDoIt();
```

不返回任何值，只在函数中做事



函数返回类型

返回数字
返回布尔型 true / false
返回字符串
没有返回值

```
function justDoIt()  
{  
  document.write("Just Do It!");  
}  
justDoIt();
```

不返回任何值，只在函数中做事



理解递归函数

$$n! = \begin{cases} 1 & n \leq 1 \\ n \cdot (n-1)! & \text{Otherwise} \end{cases}$$

```
function factorial(n) {  
  if (n <= 1) {  
    return 1;  
  }  
  
  return n * factorial(n - 1);  
}
```



了解ES6

Let 和 var的区别

<https://www.hackerrank.com/challenges/js10-let-and-const/topics/javascript-let-const-var>

Class

<https://www.hackerrank.com/challenges/js10-class/topics/javascript-classes>

Arrow Functions

<https://www.hackerrank.com/challenges/js10-arrows/topics/javascript-arrow-functions>



Button按钮的事件绑定



事件绑定的方法

I: 绑定元素属性

这些属性称为“事件处理器”

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5
6     <script>
7       function clickme(){
8         alert("我被点击了一下");
9       }
10    </script>
11  </head>
12  <body>
13    <div id="login">
14      <button onClick="clickme()" id="clickbtn" >这是一个按钮</button>
15    </div>
16  </body>
17 </html>
```

事件处理器



事件绑定的方法

II: 绑定对象属性onclick

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8" />
5      </head>
6      <body>
7          <div id="login">
8              <form>
9                  <button name="clickbtn" value="点我" >点我</button>
10             </form>
11         </div>
12     </body>
13
14     <script>
15         function clickme(){
16             alert("我被点击了一下");
17         }
18         document.forms[0].clickbtn.onclick=clickme;
19     </script>
20 </html>
```



事件绑定的方法

III: 绑定 IE4+<SCRIPT FOR> 标识

只在IE中有效 避免使用

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8" />
5      </head>
6      <body>
7          <div id="login">
8              <form>
9                  <button name="clickbtn" id="btn1" value="点我" >点我</button>
10             </form>
11         </div>
12     </body>
13
14     <script for="btn1" event="onclick">
15         alert("我被点击了一下");
16     </script>
17 </html>
```



事件绑定的方法

V: 使用 W3C DOM 的 `addEventListener()` 方法

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8" />
5      </head>
6      <body>
7          <div id="login">
8              <form>
9                  <button name="clickbtn" id="btn1" value="点我" >点我</button>
10             </form>
11         </div>
12     </body>
13
14     <script>
15         function clickme(){
16             alert("我被点击了一下");
17         }
18
19         document.getElementById("btn1").addEventListener('click',clickme);
20     </script>
21 </html>
```



无处不在的表单

注册
登录
报名页
在线申请
网上开店

登录

帐号

帐号

文本框

请输入帐号

文本框数据验证结果显示

密码

密码

文本框

请输入密码

文本框数据验证结果显示

登录

提交按钮

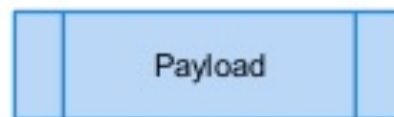
找回密码

还没有注册帐号？[立即注册](#)



表单数据发送的方式 GET和POST

GET vs. POST



Form Data, JSON Strings, Query Parameters, View States, etc.



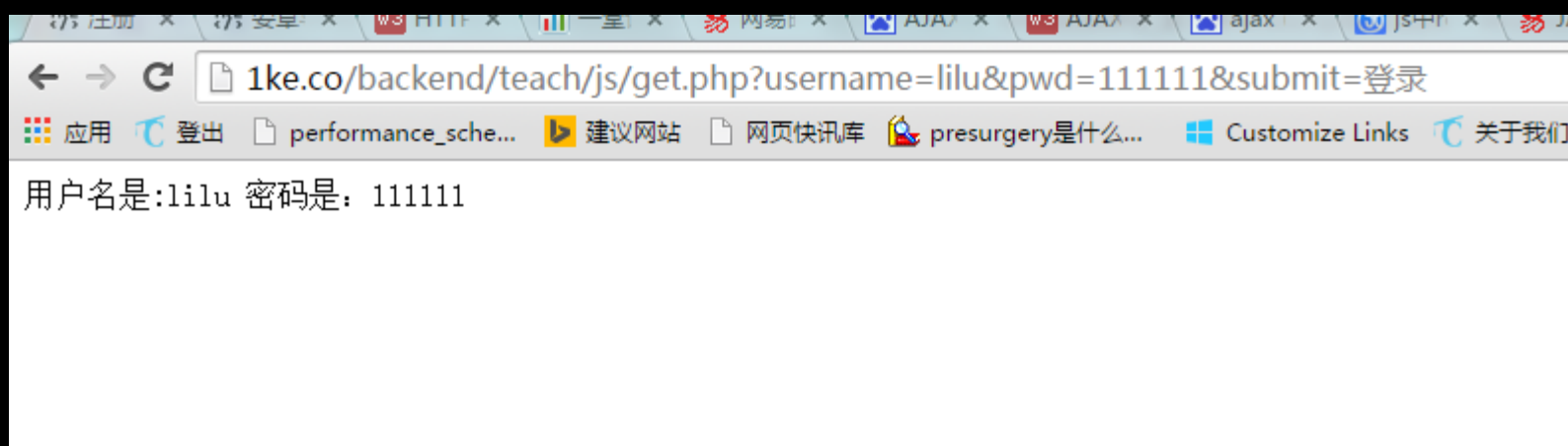
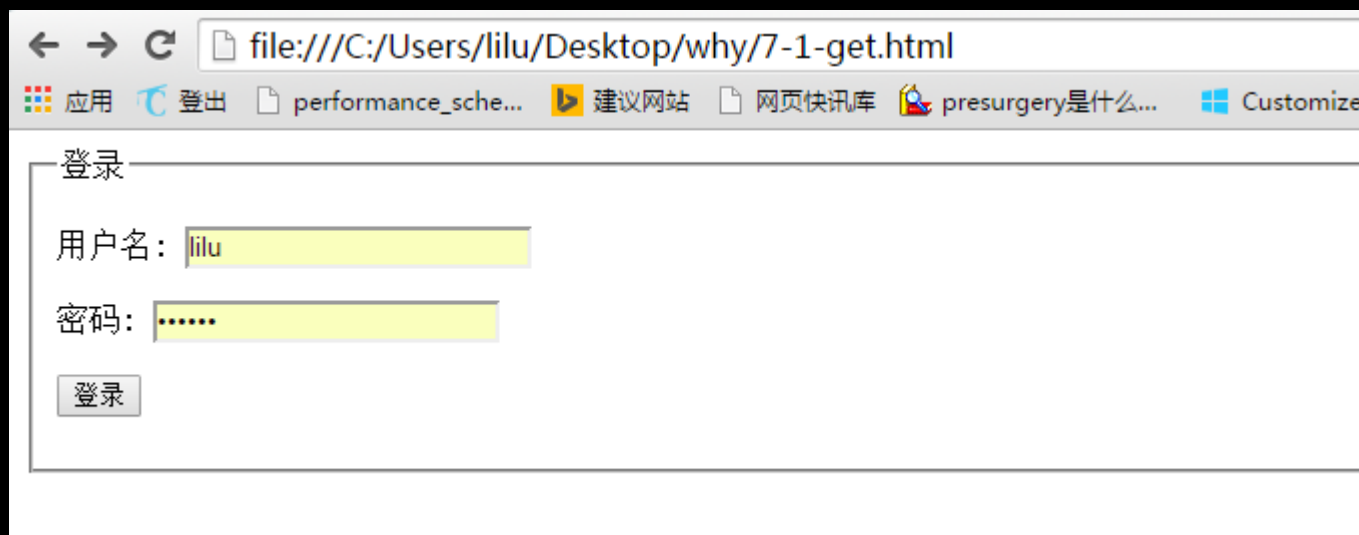
GET方式发送表单

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5   </head>
6   <body>
7     <form action="http://1ke.co/backend/teach/js/get.php" method="get">
8       <fieldset>
9         <legend>登录</legend>
10        <p>
11          用户名:
12          <input type="text" name="username" />
13        </p>
14        <p>
15          密码:
16          <input type="password" name="pwd"/>
17        </p>
18        <p>
19          <input type="submit" value="登录" />
20        </p>
21      </fieldset>
22    </form>
23  </div>
24 </body>
25 </html>
```

思考安全问题



GET方式发送表单

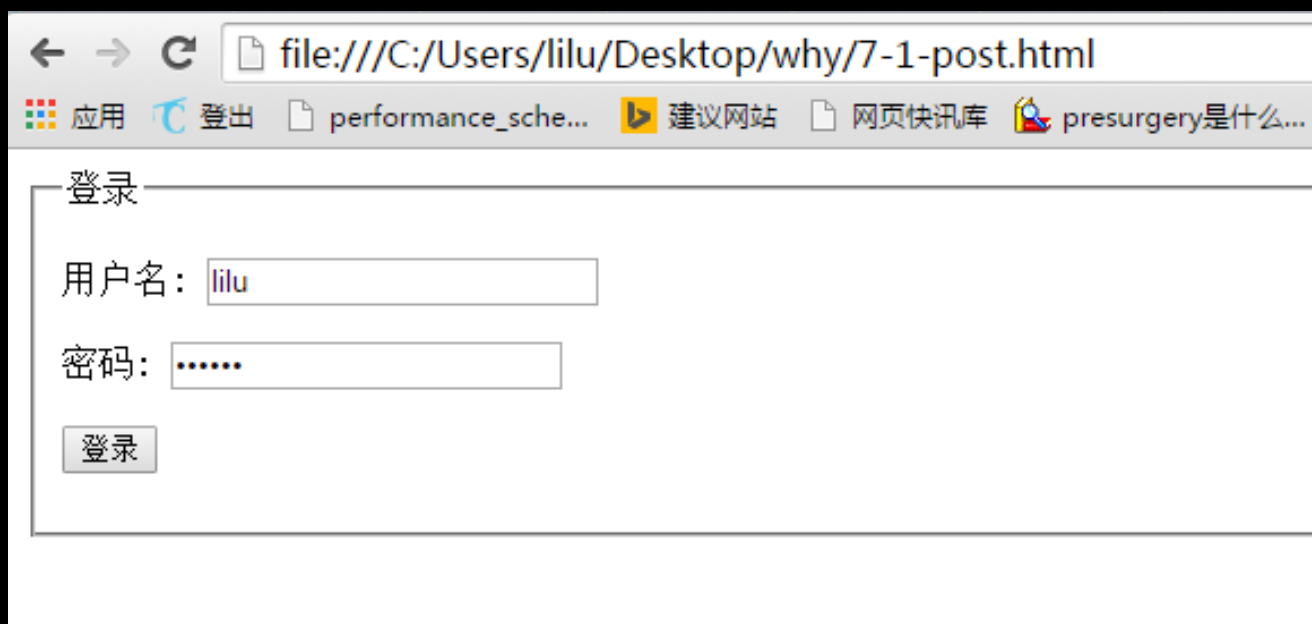


POST方式发送表单

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5   </head>
6   <body>
7     <form action="http://1ke.co/backend/teach/js/post.php" method="post">
8       <fieldset>
9         <legend>登录</legend>
10        <p>
11          用户名:
12          <input type="text" name="username" />
13        </p>
14        <p>
15          密码:
16          <input type="password" name="pwd"/>
17        </p>
18        <p>
19          <input type="submit" value="登录" name="submit" id="submit"/>
20        </p>
21      </fieldset>
22    </form>
23  </div>
24 </body>
25 </html>
```



POST方式发送表单



file:///C:/Users/lilu/Desktop/why/7-1-post.html

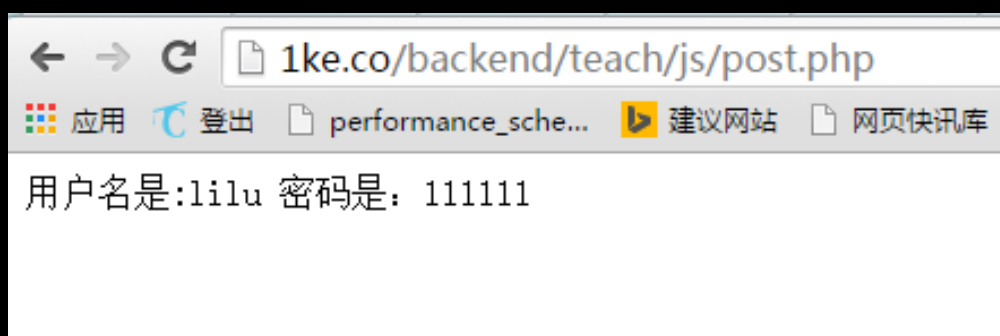
应用 登出 performance_sche... 建议网站 网页快讯库 presurgery是什么...

登录

用户名: lilu

密码: 111111

登录



1ke.co/backend/teach/js/post.php

应用 登出 performance_sche... 建议网站 网页快讯库

用户名是:lilu 密码是: 111111



PHP对于GET提交和POST提交的处理

节点科技

```
1 <?php
2 header("Content-type: text/html; charset=UTF-8");
3 $username= $_GET["username"];
4 $pwd=$_GET["pwd"];
5 echo "用户名是:".$username." 密码是: ".$pwd;
6
7 ?>
```

```
1 <?php
2 header("Content-type: text/html; charset=UTF-8");
3 $username= $_POST["username"];
4 $pwd=$_POST["pwd"];
5 echo "用户名是:".$username." 密码是: ".$pwd;
6
7 ?>
```

傻傻的展示只是为了向同学们展示后端PHP语言如何与前端提交的数据交互的，真正的项目里面没人会这么做



UNICRONCLASS

GET和POST的区别

表面上的区别：

GET使用URL或Cookie传参。而POST将数据放在BODY中。
GET的URL会有长度上的限制，则POST的数据则可以非常大。
POST比GET安全，因为数据在地址栏上不可见。

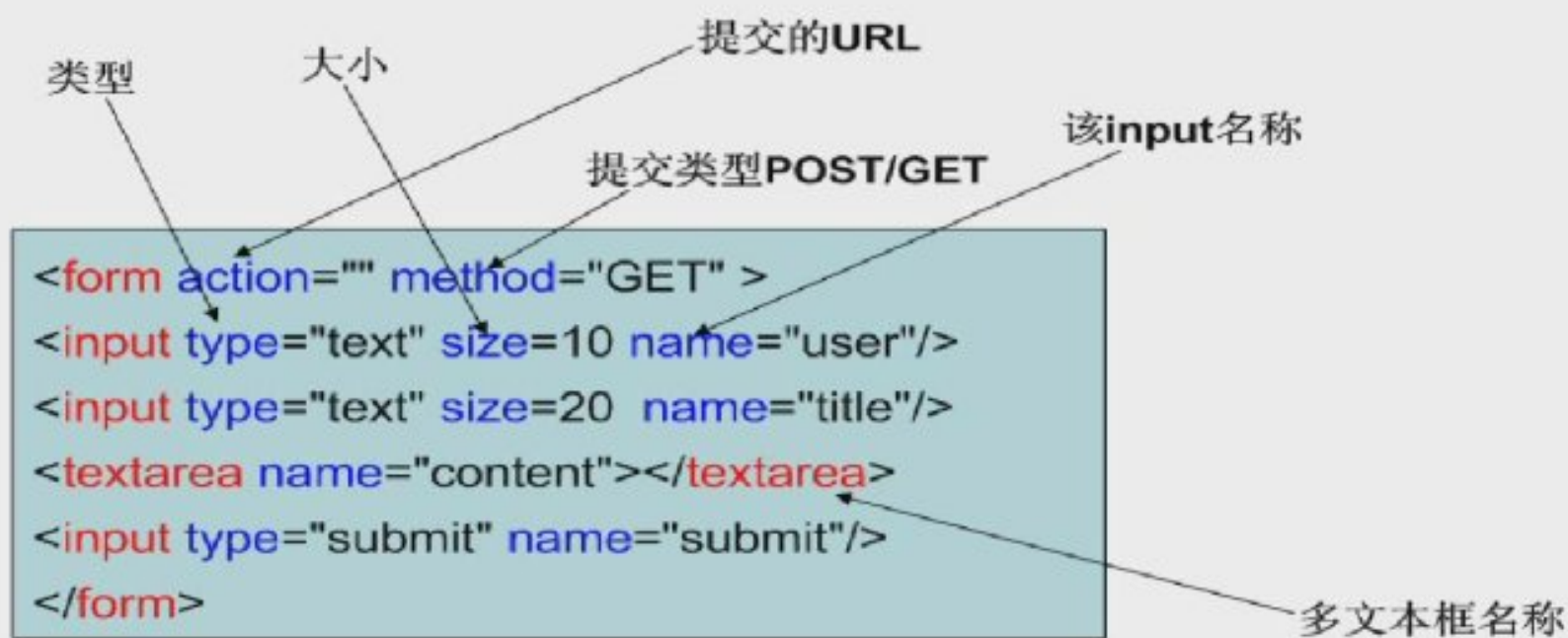
更多知识点可以看

http://www.w3school.com.cn/tags/html_ref_httpmethods.asp

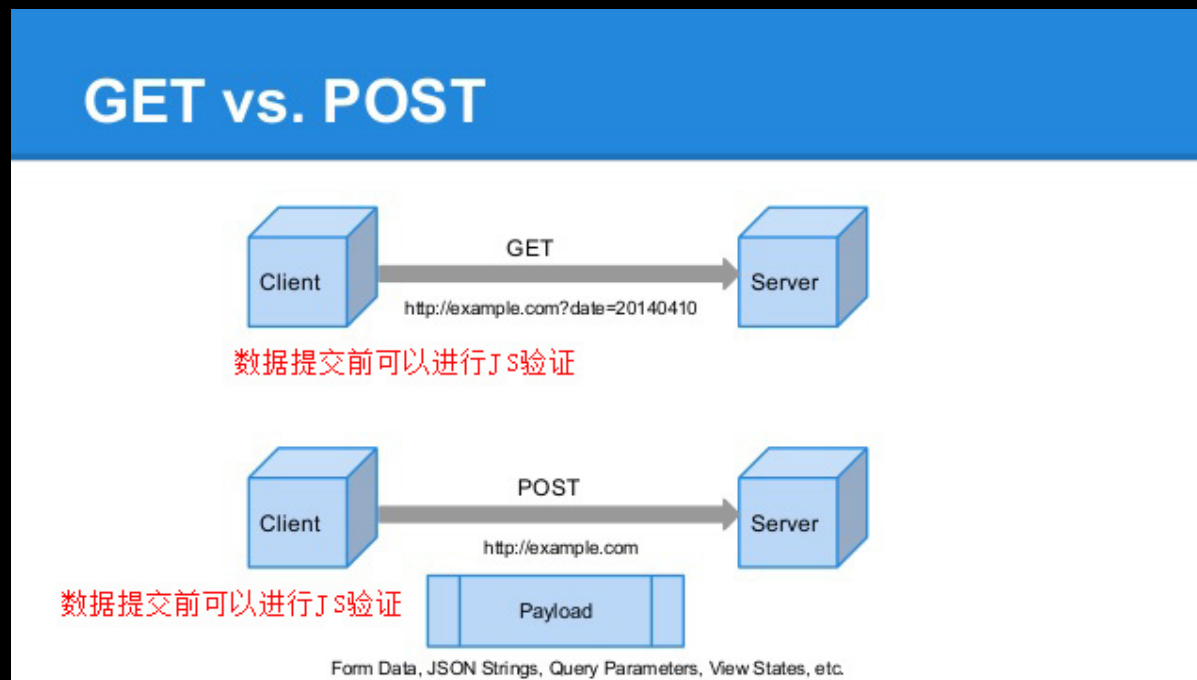


表单HTML详解

2、HTML部分表单标签的使用



为什么会有JS表单验证



为了及时查错，少让用户走冤枉路

为了杜绝垃圾数据，恶意数据

仅仅做好客户端数据验证是不够的，服务器端也需要验证；
服务器端的验证是后端程序员要做的事情，前端程序员一般不管



JS表单验证

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <script>
6       function CheckPost(){
7         return false; return false 表单不会被提交
8         //return true; return true 表单会被提交到action指定的地址
9       }
10    </script>
11  </head>
12  <body>
13    <form action="/test.php" method="post" name="myform"
14      onsubmit="return CheckPost();" >绑定元素属性 注意这里的return
15    <p>姓名: <input type="text" size=10 name="user" /></p>
16    <p>标题: <input type="text" size=20 id="title" /></p>
17    <p>内容: <textarea name="content"></textarea></p>
18    <input type="submit" value="文章发布" />
19  </form>
20 </body>
</html>
```



JS表单验证

1、提交表单Javascript验证

```
<form action="" method="post" name="myform" onsubmit="return CheckPost();">
```

```
<SCRIPT language=javascript>
```

```
function CheckPost()
```

```
{
```

```
  if (myform.user.value=="")
```

```
  {
```

```
    alert("请填写用户");  
    myform.user.focus();  
    return false;
```

```
  }
```

```
  if (myform.title.value.length<5)
```

```
  {
```

```
    alert("标题不能少于5个字符");  
    myform.title.focus();
```

```
  .....  
}
```

执行标签

执行函数

取得title长度

JS自定义函数

取得user值



JS表单验证

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <script>
6       function CheckPost(){
7         if(myform.user.value==""){
8           alert("请输入姓名");
9           myform.user.focus();
10          return false;
11        }
12        if(myform.title.value.length<5){
13          alert("标题不能小于5个字符");
14          myform.title.focus();
15          return false;
16        }
17
18        return true;
19      }
20    </script>
21  </head>
22  <body>
23    <form action="/test.php" method="post" name="myform" onsubmit="return CheckPost();" >
24      <p>姓名: <input type="text" size=10 name="user" /></p>
25      <p>标题: <input type="text" size=20 name="title" /></p>
26      <p>内容: <textarea name="content"></textarea></p>
27      <input type="submit" value="文章发布" />
28    </form>
29  </body>
30 </html>
```



getElementById() 及其他常用属性和方法

获取元素 `var xxx=document.getElementById("xxxxx")`

给元素设置html `xxx.innerHTML="xxxxxxxxxx"`

给元素设置class `xxx.setAttribute("class", "xxxx")`

onFocus事件 鼠标指针插入时

onBlur事件 鼠标离开时



onBlur事件 设置class

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <style>
6       .focus{
7         background:pink;
8       }
9       .unfocus{
10        background:white;
11      }
12    </style>
13
14    <script>
15      function setStyle(tagid){
16        var tag=document.getElementById(tagid);
17        if(tag.value==""){
18          tag.setAttribute("class","focus");
19        }else{
20          tag.setAttribute("class","unfocus");
21        }
22      }
23    </script>
24  </head>
25  <body>
26    <p>用户名:<input type="text" name="username" onBlur="setStyle(this.id)" id="txt_username" /></p>
27    <p>密码:<input type="password" name="pwd" onBlur="setStyle(this.id)" id="txt_pwd" /></p>
28  </body>
29 </html>
```

绑定元素事件属性 参数, this指当前input标签, id指txt_username



动态显示提示信息

```
33 <body>
34   <div id="msg"></div>
35   <form action="/test.php" method="post" name="myform" onsubmit="return CheckPost();" >
36     <p>姓名: <input type="text" size=10 name="user" /></p>
37     <p>标题: <input type="text" size=20 name="title" /></p>
38     <p>内容: <textarea name="content"></textarea></p>
39     <input type="submit" value="文章发布" />
40   </form>
41 </body>
42 </html>
```

file:///C:/Users/lilu/Desktop/why/7-1-1.html

应用 登出 performance_sche... 建议网站 网页快讯库 presurgery是什么... Cus

标题不能小于5个字符

姓名:

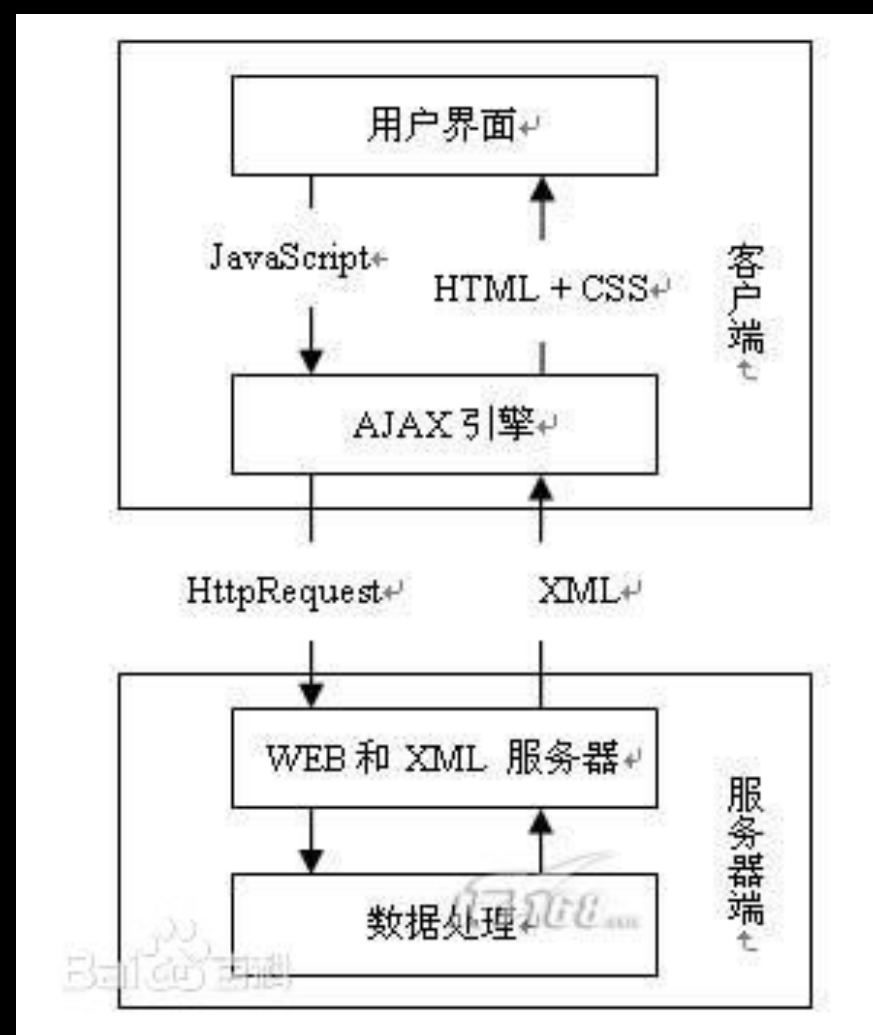
标题:

内容:



神马是AJAX

Asynchronous Javascript And XML” (异步JavaScript和XML)



常见AJAX应用

注册

邮箱地址

lilugirl2005@126.com

Email已存在!

它怎么知道我输入的email已经存在，太神奇了!

昵称

昵称

请输入昵称

密码

密码

5-20位英文、数字、符号，区分大小写

确认密码

确认密码

再输入一次密码

验证码

输入验证码

z5b74

输入验证码



常见AJAX应用

注册

邮箱地址

lilugirl2005@126.com

Email已存在!

它怎么知道我输入的email已经存在，太神奇了!

昵称

昵称

请输入昵称

密码

密码

5-20位英文、数字、符号，区分大小写

确认密码

确认密码

再输入一次密码

验证码

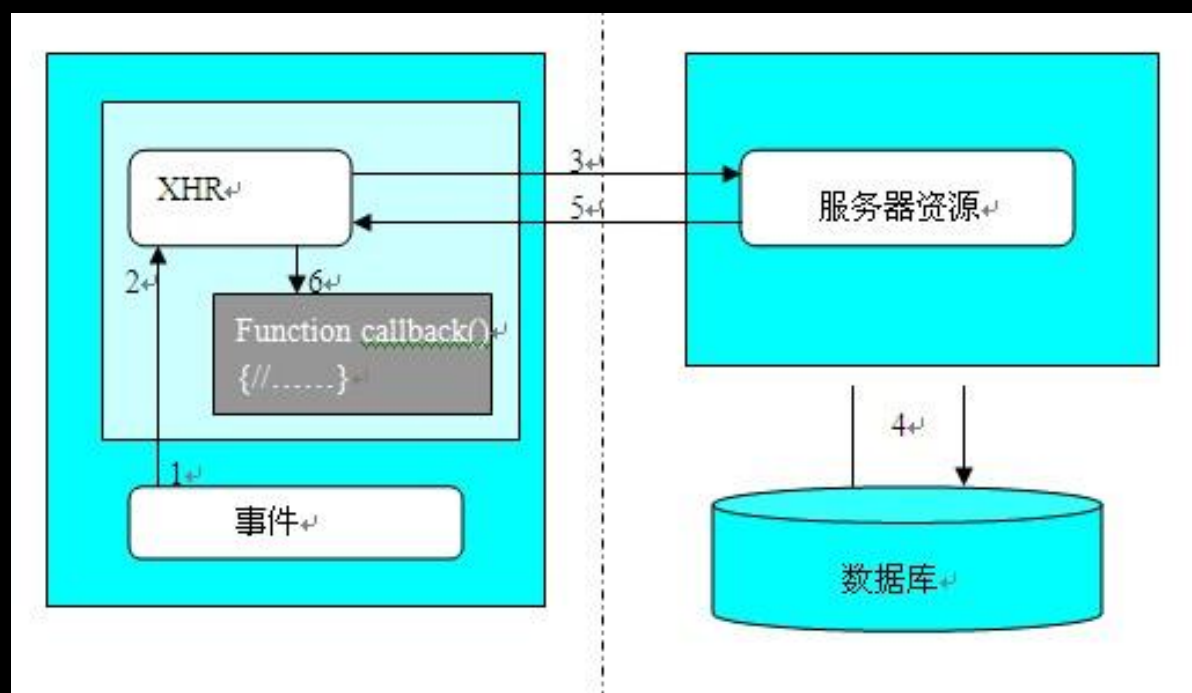
输入验证码

z5b74

输入验证码



AJAX原理



XHR=XMLHttpRequest

由事件触发，创建一个XMLHttpRequest对象，把HTTP方法（Get/Post）和目标URL以及请求返回后的回调函数设置到XMLHttpRequest对象，通过XMLHttpRequest向服务器发送请求，请求发送后继续响应用户的界面交互，只有等到请求真正从服务器返回的时候才调用callback()函数，对响应数据进行处理。



AJAX原生代码 GET

```
var xmlHttp;  
function createxmlHttpRequest() {  
    if (window.ActiveXObject) {  
        xmlHttp = new ActiveXObject("Microsoft.XMLHTTP");  
    } else if (window.XMLHttpRequest) {  
        xmlHttp = new XMLHttpRequest();  
    }  
}  
function doGet(url){  
    createxmlHttpRequest();    创建xmlHttpRequest对象  
    var msg= document.getElementById("msg");  
    xmlHttp.open("GET",url);  
    xmlHttp.send(null);        发送get请求  
    xmlHttp.onreadystatechange = function() {  
        if ((xmlHttp.readyState == 4) && (xmlHttp.status == 200)) {  
            msg.innerHTML=xmlHttp.responseText;  
            msg.setAttribute("class", "show");  
        }  
    }  
    }  
}
```

callback函数 返回服务器结果到页面上



AJAX原生代码 GET

```
function checkuser(){
    var url="checkuser.php";
    url+="?username="+myform.username.value;
    doGet(url);
}

</script>
</head>
<body>
    <div id="msg">
    </div>
    <form name="myform" onsubmit="return CheckPost()">
    <fieldset>
        <legend>登录</legend>
        <p>
            <label for="txt_username">用户名:</label>
            <input type="text" name="username" onBlur="checkuser()" />
        </p>
        <p>
            <label for="txt_pwd">密码:</label>
            <input type="password" name="pwd" />
        </p>
    </fieldset>
    </form>
</body>
```



AJAX原生代码 GET

代码示例 <http://www.1ke.co/backend/teach/js/7-3-1-get.html>

用户名是:lilu
该用户名未注册

登录

用户名:

密码:

登录



AJAX原生代码 POST

```

var xmlHttp;
function createxmlHttpRequest() {
    if (window.ActiveXObject) {
        xmlHttp = new ActiveXObject("Microsoft.XMLHTTP");
    } else if (window.XMLHttpRequest) {
        xmlHttp = new XMLHttpRequest();
    }
}

function doPOST(url,data){
    createxmlHttpRequest(); // 创建xmlHttpRequest对象
    var msg= document.getElementById("msg"); // 发送post请求
    xmlHttp.open("POST",url,true); // 设置头信息 模拟表单提交
    xmlHttp.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
    xmlHttp.send(data);
    xmlHttp.onreadystatechange = function() {
        if ((xmlHttp.readyState == 4) && (xmlHttp.status == 200)) {
            msg.innerHTML=xmlHttp.responseText;
            msg.setAttribute("class", "show"); // 页面显示服务器返回数据
        }
    }
} // callback函数

```

```

function doGet(url) {
    createxmlHttpRequest();
    var msg = document.getElementById("msg");
    xmlHttp.open("GET", url);
    xmlHttp.send(null);
    xmlHttp.onreadystatechange = function () {
        if ((xmlHttp.readyState == 4) && (xmlHttp.status == 200)) {
            msg.innerHTML = JSON.parse(xmlHttp.responseText).message;
            msg.setAttribute("class", "show");
        }
    }
}

```



AJAX原生代码 POST

```
function CheckPost(){
    //这里写一些基本的验证代码
    return false;
}

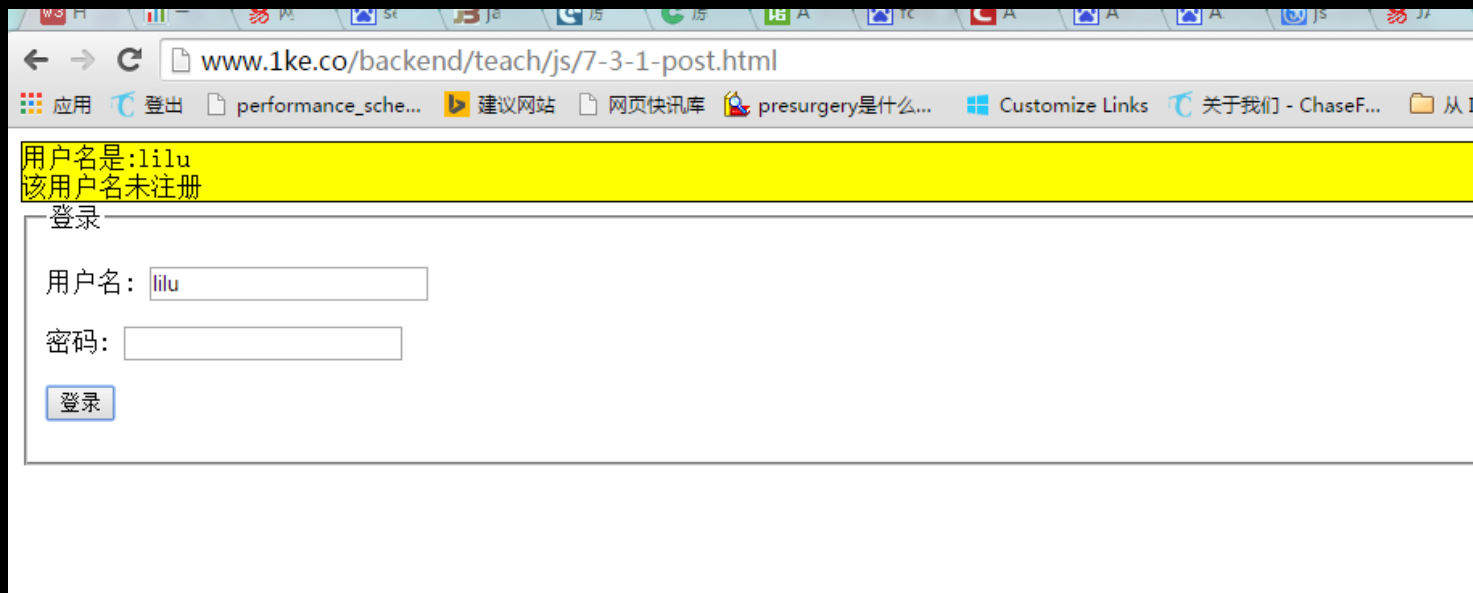
function checkuser(){
    var url="checkuser_post.php";
    data="username="+myform.username.value;
    doPOST(url,data);
}
</script>

</head>
<body>
    <div id="msg">
    </div>
    <form name="myform" onsubmit="return CheckPost()">
        <fieldset>
            <legend>登录</legend>
            <p>
                <label for="txt_username">用户名:</label>
                <input type="text" name="username" onBlur="checkuser()" />
            </p>
            <p>
                <label for="txt_pwd">密码:</label>
                <input type="password" name="pwd" />
            </p>
            <p>
                <input type="submit" value="登录" />
            </p>
        </fieldset>
    </form>
</body>
</html>
```



AJAX原生代码 POST

代码示例 <http://www.1ke.co/backend/teach/js/7-3-1-post.html>



AJAX原生代码

XMLHttpRequest对象的readyState与status:

readyState有五种状态:

- 0 (未初始化): (XMLHttpRequest)对象已经创建, 但还没有调用open()方法;
- 1 (载入): 已经调用open() 方法, 但尚未发送请求;
- 2 (载入完成): 请求已经发送完成;
- 3 (交互): 可以接收到部分响应数据;
- 4 (完成): 已经接收到了全部数据, 并且连接已经关闭。

如此一来, 你应该就能明白readyState的功能,

而status实际是一种辅状态判断, 只是status更多是服务器方的状态判断。

关于status, 由于它的状态有几十种, 我只列出平时常用的几种:

- 100——客户必须继续发出请求
- 101——客户要求服务器根据请求转换HTTP协议版本
- 200——成功
- 201——提示知道新文件的URL
- 300——请求的资源可在多处得到
- 301——删除请求数据
- 404——没有发现文件、查询或URI
- 500——服务器产生内部错误

想了解更多的同学可以自学 <http://www.w3school.com.cn/ajax/index.asp>



补充学习

<http://www.cnblogs.com/purediy/archive/2012/06/10/2544184.html>

http://www.w3school.com.cn/js/js_obj_regexp.asp

http://www.w3school.com.cn/js/js_obj_array.asp

http://www.w3school.com.cn/jsref/met_doc_getelementbyid.asp

http://www.w3school.com.cn/jsref/event_onfocus.asp

<http://www.jb51.net/article/32824.htm>

<http://www.jb51.net/article/42091.htm>

http://www.w3school.com.cn/tags/tag_form.asp

<http://www.w3school.com.cn/ajax/>

<http://www.imooc.com/learn/385>



作业

作业1 完成一个用户注册表单的验证和提醒 要求：
表单包含文本框，下拉框，单选框，复选框；
对表单进行验证，显示提醒信息

作业2 验证成功后，打印document.write所有提交信息
用AJAX 验证部分数据
参考教程 http://www.w3school.com.cn/tags/tag_form.asp
<http://www.w3school.com.cn/ajax/index.asp>

作业3 纯手写js调用公开API <http://www.1ke.co/backend/teach/js/json.php>
在页面上展示数据 显示汽车品牌列表

