Web程序设计

第三章： JavaScript编程基础

第三章： JavaScript编程基础

第三章： JavaScript编程基础

1、JavaScript简介

1.1JavaScript介绍

●JavaScript 是世界上最流行的编程语言。

它可用于 HTML 和 web，更可广泛用于服务器、PC、笔记本电脑、平板电脑和智能手机等设备。

● JavaScript是一种解释性的脚本语言，采用在HTML文本中嵌入小程序段的方式，开发过程非常简单。

● JavaScript可以直接对用户或者客户的输入做出响应，而不需要经过Web服务器程序，这减少了客户浏览器与服务器之间的通信量，提高了速度。

● JavaScript是一种与平台无关的解释性脚本语言，依赖于浏览器，而与操作系统无关，只要计算机能运行浏览器，而且该浏览器支持JavaScript，就可以执行JavaScript脚本。

第三章： JavaScript编程基础

1、JavaScript简介

1.2 JavaScript与Java的区别

1）JavaScript与Java都是与平台无关的语言，它们都具有很强的实用性。

2）两者处理方式不同。Java程序必须经过编译，形成独立的字节码，然后在相应的Java虚拟机上执行，正由于Java虚拟机的存在，才使Java能够实现跨平台。JavaScript脚本是嵌入在HTML文本中，不需要编译，通过浏览器逐行解释执行。

3）在Java程序中，如果需要某个变量，在使用之前必须进行声明（强变量模式），而JavaScript不必事先声明就可以使用变量（弱变量模式）。

第三章： JavaScript编程基础

1、JavaScript简介

1.2 JavaScript与Java的区别：

4）Java采用面向对象的程序设计方法，编程时需要创建对象，而后编程。JavaScript是基于对象编程，它可以将HTML的标记、浏览器相关对象、自定义对象作为处理对象，基于它的属性和方法编程。

5）JavaScript是一种嵌入语言，它通过在HTML文本中使用标记：<script>…</script>来插入JavaScript小程序。Java以类为单位建立独立程序文件

6）JavaScript是一种脚本语言，学习起来比Java简单得多.

第三章： JavaScript编程基础

1、JavaScript简介

❖1.3JavaScript的实现

完整的JavaScript编程包括三个组成部分：

ECMAScript（核心部分）

是JavaScript的基础部分，包括数据类型、控制语句、保留字、函数、对象等。

DOM （文档对象模型）

JavaScript 操作 HTML 文档的接口，提供了访问 HTML 文档的途径以及操作方法。

BOM（浏览器对象模型）

提供了独立于内容而在浏览器窗口之间进行交互的对象和方法。

第三章： JavaScript编程基础

1、JavaScript简介

1.3 JavaScript脚本的使用

- HTML 中的脚本必须位于 <script> 与 </script> 标签之间；
- 脚本可被放置在 HTML 页面的 <body> 和 <head> 部分中；
- <Script>格式如下：

1）<script text="type/JavaScript">

脚本

</script>

2）<script src="filename.js"></script>

其中：filename.js是独立命名的JavaScript文件，其扩展名必须为.js，文件中不含<script>标记

2、JavaScript基础

2.1 标志符

JavaScript标志符命名规范：

- 必须以字母开头
- 也能以 $ 和 _ 符号开头（不过不推荐这么做）
- 名称对大小写敏感（y 和 Y 是不同的变量）
- 长度没有特定要求。

第三章： JavaScript编程基础

2、JavaScript基础

2.2 数据类型与运算符

（1）常见的数据类型一共有7种：

① 数字类型：包括整数和浮点数。整数可以为正整数、0或者负整数，浮点数可以包括小数点，如8.33，或者“E”（也可为“e”） 如7E-2等。

② 字符串类型：字符串数据应加上单引号或者双引号，例如“hello”。

③ 对象类型：这是JavaScript的重要组成部分，将在后面的小节中介绍。

```
var obj_person={firstName:“Marry”, lastName:“Eich”};
```

④布尔类型：可以为true和false两个值。

⑤数组类型：var name=[“zhangsan”,”lisi”,”wangwu”];

⑥Null（空）、Undefined（未定义）。数据类型可以typeof查看

2、JavaScript基础

2.2 数据类型与运算符

（1）常用运算符

① 算数运算符:+、-、*、/、%、**、++、--。

② 赋值运算符=、+=、-=、*=、/=、%=、**=。

③ 关系：>、<、>=、<=、==、!=、===、（绝对等,值与类型均相等） 、！== （不绝对相等，值和类型有一个不相等，或两个都不相等）。

④ 逻辑运算符：！、&&、||。

2 JavaScript语言基础

2.3、保留字

与其他编程语言一样，JavaScript也有一些保留字，这些保留字不可以用作变量、函数名、对象名等,如var、abstract、boolean、break、int 等。

第三章： JavaScript编程基础

2 JavaScript语言基础

2.4、常用语句

- 函数定义语句

```
function 函数名称（形参）
    {
        函数执行部分
        return 表达式
    }
```

函数调用的格式:

函数名称（实参）

函数调用位置:

1）在脚本中被调用

2）通过事件驱动被调用

3）在超链接点击被调用

补充:不包含在任何函数中的变量为全局变量

2 JavaScript语言基础

2.4、常用语句

- 条件语句

```
if (条件)
{ 语句1 }
else
{ 语句2 }
```

如果条件成立，则执行语句1，否则执行语句2。

2 JavaScript语言基础

2.4、常用语句

- 分支语句

```
switch (expression)
{    case label1: 语句1;
       case label2: 语句2;
       case label3: 语句3;
       ……
       default : 语句n; }
```

如果表达式expression的值与任何一个label都不匹配，将执行default后面的语句n。

2 JavaScript语言基础

2.4、常用语句

- 循环语句

for语句：

```
for (变量初始化; 条件; 更新变量)
  { 语句 }
```

for…in语句：

```
for (变量 in 对象或数组)
    { 语句 }
```

while语句、do while语句：

```
while (条件)          do
    { 语句 }          {  语句  } while(条件);
```

2 JavaScript语言基础

2.5、事件

在客户端处理时需要通过表单控件的事件驱动机制驱动和客户端脚本来实现。

事件过程的调用方式

事件过程按其名称被识别，事件过程的命名规则是： on事件名，如Click事件过程名为onclick。事件过程的定义与调用有如下2种方式。

ICEC

2 JavaScript语言基础

2.5、事件

（1）通过<body>等标记块实现

在body标记块中用onload 等事件调用

格式如 <body onload = 函数名（）>

```
<HTML>
 <HEAD><TITLE>调用事件过程</TITLE>
  <SCRIPT LANGUAGE=javascript>
   function mysub()
{      aa = " onload事件调用";  alert(aa); }
</SCRIPT>
 </HEAD>
 <BODY onload=mysub()> </BODY>
</HTML>
```

2.5、事件

(2) 在对象定义标记中调用事件过程

在对象定义标记中设置事件过程属性可以调用函数。

```
<HTML>
  <HEAD><TITLE>调用事件过程</TITLE>
    <SCRIPT LANGUAGE=javascript>
      function mysub()
{   aa = "这是按钮的单击事件 "; alert(aa); }
</SCRIPT>
  </HEAD>
  <BODY><H3>调用事件过程</H3><HR>
    <INPUT type=button onclick=mysub() value=确定>
  </BODY>
</HTML>
```

2.5、事件

（3） 在超链接中调用事件过程

在对象定义标记中设置事件过程属性可以调用函数。

```
<HTML>
 <HEAD><TITLE>调用事件过程</TITLE>
  <SCRIPT LANGUAGE=javascript>
   function mysub()
{  aa = "这是按钮的单击事件“; alert(aa); }
</SCRIPT>
 </HEAD>
 <BODY><H3>调用事件过程</H3><HR>
   <a href=“javascript:mysub();”> 调用函数</a>
 </BODY>
</HTML>
```

2 JavaScript语言基础

2.5、事件

- 鼠标事件：onMouseDown、onMouseUp、onMouseOut、OnClick、onMouseOver、onMouseMove。
- 键盘事件：onKeydown、onKeyup、onKeypress。
- 焦点事件：onFocus、onBlur。
- 调整窗口尺寸事件：onresize。
- 加载和卸载窗口事件：onLoad、unLoad。

第三章： JavaScript编程基础

名称	说明
Focus事件	当控件收到焦点时，触发事件
Blur事件	当控件失去焦点时，触发事件
Click事件	当用户单击鼠标左键，然后抬起按键时，触发Click事件；
Dblclick事件	当用户双击鼠标左键，将触发Click事件；
Mouseover事件	当鼠标移到控件（或对象）的上方时，将触发该事件
Mousemove事件	当鼠标移到控件（或对象）的上方时，将触发该事件
Mouseout事件	当鼠标从控件（或对象）内部移出时，将触发该事件
Mousedown事件	按下鼠标键时，将出发该事件
Mouseup事件	按下鼠标键，再松开时，将出发该事件
Keypress事件	当按下和松开某个ASCII字符键时，拥有焦点的控件将会响应，事件过程将获得按键对应的ASCII码
Keydown事件	当按下键盘上的某个键时，拥有焦点的控件将会响应，事件过程将获得按键对应的键值
Keyup事件	按下键盘上的某个键，当释放该键时，拥有焦点的控件将会响应，事件过程将获得按键对应的键值

第三章： JavaScript编程基础

```
<html>
 <head>
 <script  language=javascript>
              function ss()
             { if (f1.r1(0).checked==true)
                alert(f1.r1(0).value);
                else
                  alert(f1.r1(1).value); }
            </script>
            </head>
            <body > <form  name="f1">
            <input type="text" name="t1"><br>
            <input type="text" name="t2"><br>
            <input type="radio" name=r1 value="男" checked>nan<br>
            <input type="radio" name=r1 value="女">女<br>
            <input type=button value="执行" onclick=ss()  title="this is a
                example">
            </form>
            </body>
            </html>
```

2 JavaScript语言基础

2.6、对象——简介

JavaScript是基于对象的语言,常见对象：

- String对象:字符串对象,只有一个属性：length属性，表示字符串中包含的字符数目。
- Date对象:如果要在JavaScript程序中获取当前时间，就要用到Date对象。使用Date对象，必须首先使用关键字new来创建。
- Math对象：数学对象，有一个很重要的属性：PI（圆周率）。除此之外，它还有许多有用的数学计算方法。
- 数组对象Array :提供对数组类型数据的操作.常用属性是length,还有一些对数组元素操作的方法，比如pop,push等
- Window对象是每个HTML网页的顶层对象，在调用Window对象的方法时，不必显式调用Window对象。几个重要方法：open()、alert()、confirm()、prompt()、close()等。

2 JavaScript语言基础

❖ 2.6、对象——串对象(String)使用

String对象：内部动态性。

访问属性和方法时， 需要声明对象。

基本使用格式：objectName.属性/方法

1. 串对象的属性

该对象常用的只有一个属性，即length。它表明了字符串中的字符个数，包括所有符号。

第三章： JavaScript编程基础

2 JavaScript语言基础

❖ 2.6、对象——串对象(String)使用

串对象的方法

说明：String对象的方法共有三十多个。主要用于有关字符串在Web页面中的显示、字体大小、字体颜色、字符的搜索以及字符的大小写转换。

1）字符搜索：

格式：indexOf [charactor,fromIndex]

❖ 作用：从指定formIndtx位置开始搜索charactor第一次出现的位置。

2 JavaScript语言基础

2.6、对象——串对象(String)使用

2）返回字符串的一部分子串：

- 格式：substring(start,end) ； charAt(n)
- 返回从start开始到end的字符; 返回第n个字符，n:0-length

3）字符串大小写转换

- toLowerCase()：小写转换
- toUpperCase()：大写转换

ICEC

2 JavaScript语言基础

2.6、对象——串对象(String)使用

4）有关字符显示的控制方法

- big()：用大字体显示；
- italics()：斜体字显示
- bold()：粗体字显示
- blink()：字符闪烁显示
- small()：字符用小体字显示
- fixed()：固定高亮字显示
- fontcolor(color)：字体颜色方法；
- fontsize(size)：控制字体大小等
- 详细内容参加：https://www.runoob.com/jsref/jsref-obj-string.html

2 JavaScript语言基础

2.6、对象——数学函数 Math对象 使用

- 功能：提供除加、减、乘、除以外的一引些自述运算。如对数，平方根等 。
- 静动性：静态对象

1.主要属性

- Math中提供了10个属性，它们是数学中经常用到的常数e、以10为底的自然对数LN10、以2为底的自然对数LN2、3.14159的PI、1/2的平方根SQRT1-2、2的平方根为SQRT2等。

第三章： JavaScript编程基础

2 JavaScript语言基础

2.6、对象 ——数学函数 Math对象 使用

主要方法

- 绝对值：abs(x)
- 正弦余弦值：sin(x),cos(x)
- 反正弦反余弦 :asin(x), acos(x)
- 正切反正切：tan(x),atan(x)
- 四舍五入取整：round(x)
- 平方根：sqrt(x)
- 对一个数进行舍入取整：floor(x)
- 求自然对数：log(x)
- 求e的指数：exp(x)
- 基于几方次的值：pow(base,exponent)
- 返回0~1之间的随机数：random()

2 JavaScript语言基础

2.6、对象 ——日期对象Date 使用

简介

- 功能：提供一个有关日期和时间的对象。
- 静动性：动态性，即必须使用new运算符创建一个实例。
- Date对象没有提供直接访问的属性。只具有获取和设置日期和时间的方法。

例：var now=new Date() ; //将系统当前日期及时间作为对象now的初始值；

第三章： JavaScript编程基础

2 JavaScript语言基础

2.6、对象 ——日期对象Date 使用

1）获取日期的时间方法

- getYear()： 返回年数
- getMonth()：返回当月号数 //月份的号从0开始计数
- getDate()：返回当日号数
- getDay()：返回星期几
- getHours()：返回小时数
- getMintes()：返回分钟数
- getSeconds()：返回秒数
- getTime()：返回毫秒数 //日期起始值:1970年1月1日 00:00:00。

2 JavaScript语言基础

2.6、对象 ——日期对象Date 使用

2）设置日期和时间：　时间设定调用时不需要返回值。

- setYear()：设置年
- setDate()：设置当月号数
- setMonth()：设置当月份数
- setHours()：设置小时数
- setMintes()：设置分钟数
- setSeconds()：设置秒数
- setTime ()：设置毫秒数
- 例：now.setDate(2022);　　//将now的年份设定为2022年

2 JavaScript语言基础

2.6、对象 ——数组对象Array 使用

- 功能：提供对数组类型数据的操作。
- 静动性：动态性，即必须使用new运算符创建一个 实例
- 属性：
- 1）constructor 所建立对象的函数参考
- 2）prototype　能够为对象加入的属性和方法
- 3）length　获取数组元素的个数,即最大下标加1

第三章： JavaScript编程基础

2 JavaScript语言基础

2.6、对象 ——数组对象Array 使用

使用：可以采用new和Array对象来创建数组。

基本语法：

- 1)arrayObj=new Array()　　//说明此时创建的数组元素不固定
- 2)arrayObj=new Array(size)　　//此时创建的数组的个数为size
- 3)arrayObj=new Array(元素1，元素2，…,元素n)，存入数组的元素值及其个数

第三章： JavaScript编程基础

2 JavaScript语言基础

2.6、对象 ——数组对象Array 使用

❖方法：

1）concat(array1,arrayn)：将两个或两个以上的数组值连接起来，合并后返回结果；

2）join(string)： 将数组中元素合并为字符串，string为分隔符。如省略参数则直接合并，不再分隔；

3）pop()： 移除数组中的最后一个元素并返回该元素；

4）push(value)：在数组的末尾加上一个或多个元素，并且返回新的数组长度值；

5）reverse()：颠倒数组中元素的顺序,反向排列；

第三章： JavaScript编程基础

2 JavaScript语言基础

2.6、对象 ——数组对象Array 使用

❖方法：

6）shift()：移除数组中的第一个元素并返回该元素；

7）sort(compare Function)：在未指定排序号的情况下,按照元素的字母顺序排列，如果不是字符串类型则转换成字符串再排序,返回排序后的数组；

8）splice()：为数组删除并添加新的元素；

9）toString()：将数组所有元素返回一个字符串,其间用逗号分隔；

10）unshift(value)：为数组的开始部分加上一个或多个元素，并且返回该数组的新长度；

11）valueOf()：返回数组对象的原始值；

第三章： JavaScript编程基础

2 JavaScript语言基础

2.7、对象 ——全局函数

即JavaScript 的内部函数，可作用在JavaScript的所有内部对象上

1)parseInt

作用：将字符串转换为整数

格式：parseInt(“字符串”);

parseInt(“字符串”，“进制”);

2）parseFloat

作用：将字符串转换为实数

格式：parseFloat(“字符串”)

3）String

作用：将对象转换为字符串

格式：String（object）

4) eval

作用：函数可计算某个字符串，并执行其中的的 JavaScript 代码

格式：eval(“字符串”)

3. JavaScript的DOM编程

3.1 简介：

DOM（Document Object Model，文档对象模型）是DHTML（动态HTML）的一种方式，利用HTML、CSS、JavaScript技术进行动态编程；

DOM 是 W3C（万维网联盟）的标准，定义了访问 HTML 和 XML 文档的标准；

DOM W3C 文档对象模型 （DOM） 是中立于平台和语言的接口，它允许程序和脚本动态地访问和更新文档的内容、结构和样式。

第三章： JavaScript编程基础

3. JavaScript的DOM编程

3.2 DOM功能

- JavaScript 能够改变页面中的所有 HTML 元素内容
- JavaScript 能够改变页面中的所有 HTML 属性
- JavaScript 能够改变页面中的所有 CSS 样式
- JavaScript 能够对页面中的所有事件做出反应

ICEC

3. JavaScript的DOM编程

3.2 DOM的实现

- 通过 id 找到 HTML 元素

var x=document.getElementById(“id名”);--------简单变量

var array=document.getElementsByName(“name名”); --数组

var array= document.getElementsByTagName(”标记名”); --数组

- 改变 HTML 内容

document.getElementById(id).innerHTML=new HTML；

- 改变 HTML 属性

document.getElementById(id).attribute=new value；

第三章： JavaScript编程基础

3. JavaScript的DOM编程

3.2 DOM的实现

- 改变 HTML 样式

document.getElementById(id).style.property=new style ；

- 创建新的 HTML 元素

1）var 元素变量名=document.createElement(“html标记");

2）var 结点变量=document.createTextNode(“内容");

3）元素变量.appendChild(结点变量);

4）var 已有元素=document.getElementById(“元素id”);

5）已有元素.appendChild(新元素变量名);

ICEC

3. JavaScript的DOM编程

3.2 DOM的实现

❖ 删除已有的 HTML 元素

在母标记中删除子节点的过程：

1）var 母节点变量名=document.getElementById ("母标记id名");

2）var 子元素变量=document.getElementById ("子标记id名");

3）母节点变量名.removeChild(子元素变量);

注：DOM只能删除已有母节点中的子节点

ICEC

4. JavaScript的 BOM编程

4.1 BOM简介

JavaScript的BOM(Browser Object Model)编程技术提供了JavaScript与浏览器对话的方式，即可以通过访问浏览器对象实现动态程序设计的一种方式。

常用浏览器对象有：window、document、navigator、location、history、screen、cookies等；

其中最常用的是window对象，它也是其他对象的母对象。

ICEC

4. JavaScript的 BOM编程

4.2 window对象

简介：

- 所有浏览器都支持 window 对象。它表示浏览器窗口。
- 所有 JavaScript 全局对象、函数以及变量均自动成为 window 对象的成员。
- 全局对象是 window 对象的属性。
- 全局函数是 window 对象的方法。
- HTML DOM 的 document 也是 window 对象的属性之一

ICEC

window对象的方法

名称	说明
Alert方法	用于显示一个对话框，包括信息图标、提示信息和“确定”按钮。
Close方法	用于关闭当前浏览器窗口。父window对象调用Close方法时，会关闭所有的子窗口
Confirm方法	用于显示一个对话框，包括问号图标、提示信息、“确定”按钮和“取消”按钮
open方法	用于打开一个新的浏览器窗口，并且加载由其URL参数指定的HTML文档
Prompt方法	用于显示一个输入对话框，包括提示信息和输入框
SetTimeout方法	用于定时执行某个函数或命令

第三章： JavaScript编程基础

window方法

（1）alert方法

```
<html>
<head>
<script type="text/javascript">
function disp_alert()
{ alert("我是警告框！ ！ ") }
</script>
</head>
<body>
<input type="button" onclick="disp_alert()" value="显示警告框" />
</body>
</html>
```

第三章： JavaScript编程基础

window方法

（2） close方法

window.close();

```
<html>
<head>
<script language="JavaScript" type="text/javascript">
    function cl()
    { window.opener='';
    window.close(); }
    </script></head>
<body>
<input type="submit" onclick="cl()" name="Submit" value="提交" />
    </body>
</html>
```

window方法

（3） confirm方法

```
<html>
<head>
<script type="text/javascript">
function show_confirm()
{ var r=confirm("Press a button!");
if (r==true)
  { alert("You pressed OK!"); }
else
  {  alert("You pressed Cancel!"); } }
</script>
</head>
<body>
<input type="button" onclick="show_confirm()" value="Show a confirm
    box" />
</body>
</html>
```

第三章： JavaScript编程基础

window方法

（4） open方法

```
<html>
<head>
<script LANGUAGE="JavaScript">
    function openwin() {
window.open ("page.html")}
 </script>
</head>
<body>
<input type="button" onclick="openwin()" value="打开窗口">
</body>
</html>
```

第三章： JavaScript编程基础

window方法

（5）prompt方法

```
 <html>
<head>
<script type="text/javascript">
function disp_prompt()
  {   var name=prompt("请输入您的名字","Bill Gates")
  if (name!=null && name!="")
    { document.write("你好！ " + name + " 今天过得怎么样？") } }
</script>
</head>
<body>
<input type="button" onclick="disp_prompt()" value="显示提示框" />

</body>
</html>
```

ICEC

4. JavaScript的 BOM编程

4.3 history对象

作用：包含浏览器的历史

- 方法：history.back() - 与在浏览器点击后退按钮相同，返回上一URL;
- history.forward() - 与在浏览器中点击按钮向前相同，前进下一URL;
- History.go(n)-根据n的正负觉得前进或后退

ICEC

4. JavaScript的 BOM编程

4.4 navigator对象

作用：对象包含有关访问者浏览器的信息

属性：

```
<script> txt = "<p>Browser CodeName: " + navigator.appCodeName + "</p>"; txt+= "<p>Browser Name: " + navigator.appName + "</p>"; txt+= "<p>Browser Version: " + navigator.appVersion + "</p>"; txt+= "<p>Cookies Enabled: " + navigator.cookieEnabled + "</p>"; txt+= "<p>Platform: " + navigator.platform + "</p>"; txt+= "<p>User-agent header: " + navigator.userAgent + "</p>"; txt+= "<p>User-agent language: " + navigator.systemLanguage + "</p>";
</script>
```

5.JavaScript的正则表达式

5.1简介：

在计算机科学中，是指一个用来描述或者匹配一系列符合某个句法规则的字符串的匹配字符串。在很多文本编辑器或其他工具里，正则表达式（ **RegExp**）通常被用来检索和/或替换那些符合某个模式的文本内容。许多程序设计语言都支持利用正则表达式进行字符串操作。

详细参加： https://www.runoob.com/jsref/jsref-obj-regexp.html

5.JavaScript的正则表达式

5.2 正则表达式 RegExp）实现：

正则表达式的检索实现包括定义模式及执行两个过程：

1)定义 RegExp

格式：var 模式对象=new RegExp（模式[,属性]）

说明：模式对象是创建的正则表达式对象，用于匹配模式用；

模式是待检索匹配的字符，即匹配的模式，写在两个/之间。

属性是指对模式匹配的范围或方式进行说明，(i:不区分大小写;g:全局范围匹配；m：多行匹配）

例：var e=new RegExp(/[A-Z]/) :匹配模式为英文大写字母

第三章： JavaScript编程基础

5.JavaScript的正则表达式

5.2 正则表达式 RegExp）实现：

2）RegExp的执行

RegExp有三个方法执行匹配，分别是test()、exec()、compile().

* test("待查找字符串")

作用：用于检索指定指定字符串中是否含有模式字符串；

返回值：true| false;

例：e.test(“abABC”):查找字符串”abABC”是否符合模式e.

* exec("待查找字符串")

作用：用于检索制定字符串中的模式；

返回值：找到的字符串或是null

第三章： JavaScript编程基础

5.3、javascript正则表达式案例

（案例1）验证输入身份证号码

```
<script language="javascript">
function check(myform){
    if(myform.number.value==""){
    alert("请输入身份证号码地址!");myform.number.focus();return;}
    if(!checkeNO(myform.number.value)){
    alert("您输入的身份证号码不正确!");myform.number.focus();return;}
    myform.submit();}
function checkeNO(NO){
    var str=NO;
     //在JavaScript中，正则表达式只能使用"/"开头和结束，不能使用双引号
    var Expression=/\d{17}[\d|X]|\d{15}/;
    var objExp=new RegExp(Expression);
    if(objExp.test(str)==true){
            return true; }else{ return false; } }
</script>
```

第三章： JavaScript编程基础

5.3、javascript正则表达式案例

（案例1）验证输入身份证号码

表单的设计：

```
<form action="" method ="post" name= "form1">
身份证号码:
<input name="number" type= " text " >
<INPUT  type="button" value="注 册" name="Submit"
    onClick="check(form1)">
<input name="button1" type="reset" value="重置">
</form>
```

5.3、javascript正则表达式案例

（案例2）验证输入字符是否为汉字

```
<script language="javascript">
function check(myform){
    if(myform.realname.value==""){
    alert("请输入真实姓名!");myform.realname.focus();return;          }
    if(checkrealname(myform.realname.value)){
    alert("您输入真实姓名不正确!");myform.realname.focus();return; }
    myform.submit(); }
function checkrealname(realname){
    var str=realname;
     //在JavaScript中，正则表达式只能使用"/"开头和结束，不能使用双引号
    var Expression=/[^\u4E00-\u9FA5]/;
    var objExp=new RegExp(Expression);
    if(objExp.test(str)==true){
    return true; }else{ return false; } }
</script>
```

5.3、javascript正则表达式案例

（案例2）验证输入字符是否为汉字

表单的设计：

```
<form action="" method ="post" name= "form1">
姓名:
<input name="realname" type= " text " >
<INPUT  type="button" value="注 册" name="Submit"
    onClick="check(form1)">
<input name="button1" type="reset" value="重置">
</form>
```

第三章： JavaScript编程基础

6、javascript案例

（案例1）检查表单元素是否为空

表单的设计：

```
 <form name="form1" method="post" action="">
留言人：<input type="text" name="lyr" size="20"> <br>
留言主题：<input type="text" name="lyzt" size="66"><br>
留言内容：<textarea name="lynr" cols="64"
    rows="10"></textarea><br>
<input name="button" type="button" value="提交"
    onClick="check(form1)">
<input name="button1" type="reset" value="重置">
</form>
```

6、javascript案例

（案例1）检查表单元素是否为空

Javascript设计方法1：

```
<script language="javascript">
 function check(){
  if(form1.lyr.value==""){
  alert("留言人不能为空"); form1.lyr.focus();return;
  }
 if (form1.lyzt.value==""){
 alert("留言主题不能为空");form1.lyzt.focus();return;
 }
 if (form1.lynr.value==""){
 alert("留言内容不能为空");form1.lynr.focus();return;
  }
```

6、javascript案例

（案例1）检查表单元素是否为空

Javascript设计方法2 ：

```
<script  language="javascript">
 function check(Form){
 for(i=0; i<Form.length;i++){
          if (Form.elements[i].value==""){
 alert(Form.elements[i].name + "不能为空");
 Form.elements[i].focus();
return;
} }
Form.submit();
}
</script>
```

第三章： JavaScript编程基础

6、javascript案例

（案例2）验证输入E-mail是否正确

Javascript的设计：

\\ 调用checkmail()函数判断E-mail是否正确，显示相应提示信息：

```
<script language="javascript">
function check(myform){
    if(myform.e_mail.value==""){
    alert("请输入Email地址!");myform.e_mail.focus();return; }
    if(!checkemail(myform.e_mail.value)){
    alert("您输入Email地址不正确!");myform.e_mail.focus();return; }
    myform.submit(); }
</script>
```

6、javascript案例

（案例2）验证输入E-mail是否正确

Javascript的设计：

```
\\ 检测E-mail地址是否正确的函数checkmail()：
<script language="javascript">
function checkemail(email){
    var str=email;
    var Expression=/\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*/;
//验证e-mail的正则表达式
    var objExp=new RegExp(Expression);
    if(objExp.test(str)==true){
          return true; } else { return false;
    } }    </script>
```

第三章： JavaScript编程基础

6、javascript案例

（案例2）验证输入E-mail是否正确

表单的设计：

```
<form action="" method ="post" name="form1">
E-mail:
<input name="e_mail" type= " text " >
<INPUT  type="button" value="注 册" name="Submit"
    onClick="check(form1)">
<input name="button1" type="reset" value="重置">
</form>
```

第三章： JavaScript编程基础

9、javascript案例

（案例3）验证两次输入密码是否一致

```
<script language="javascript">
function check(myform){
    if(myform.username.value==""){
    alert("请输入用户名!");myform.username.focus();return; }
    if(myform.PWD.value==""){
    alert("请输入密码!");myform.PWD.focus();return; }
    if(myform.PWD1.value==""){
    alert("请确认密码!");myform.PWD1.focus();return; }
    if(myform.PWD1.value!=myform.PWD.value){
    alert("您两次输入的密码不一致，请重新输入!"); myform.PWD.focus();re
          }
    myform.submit(); }
</script>
```

第三章： JavaScript编程基础

6、javascript案例

（案例3）验证两次输入密码是否一致

表单的设计：

```
<form action="" method ="post" name= "form1">
用户名:<input name="username" type= " text " > <br>
密码：<input name="PWD" type= “password" > <br>
重复密码：<input name="PWD1" type= “password" > <br>
<INPUT  type="button" value="注 册" name="Submit"
    onClick="check(form1)">
<input name="button1" type="reset" value="重置">
</form>
```

6 利用网页编辑器制作网页

如果要开发的网页非常复杂，如网站的主页等，那么它的HTML脚本就会非常庞杂。如果脚本都由人来一行一行的编写，很容易出错，开发工作量也很巨大。为此，一些公司推出了网页编辑器，利用可视化的界面来编写HTML网页。

目前，常见的网页编辑器主要有两种：Frontpage和Dreamweaver。后者是针对专业网页设计师特别设计的视觉化网页开发工具，利用它可以很容易制作出跨越平台限制和跨越浏览器限制的充满动感的网页。

Web程序设计

下　课！！