# 重点内容

Bom编程 四个重要对象：windows 、location、history、screen

DOM编程：DOM（document Object Model）文档对象模型编程，document代表一个html页面

事件编程：点击事件等

对象的理解：自定义对象，内置对象

# 基础知识

JS是Netscape公司的产品，JS只需解释就可以执行，Java需要先编译成字节码文件，再执行

尽管 ECMAScript 是一个重要的标准，但它并不是 JavaScript 唯一的部分，一个完整的 JavaScript 实现是由以下 3 个不同部分组成的：①ECMAScript②Dom③BOM

也就是：JavaScript语法基础、使用JS操作网页（DOM）、使用JS操作浏览器（BOM）

变量： var 变量名=值;

js是弱类型语言，使用var来定义任何数据类型

js中变量的类型是由变量的值决定的，所以只定义不赋值的变量，就是一个未赋值变量（undefined），未赋值的变量不能使用

var a = 20; 值来决定类型：number

var b = 3.14; 值来决定类型：number

var c = 'c'; string类型

var e = true; boolean类型

var f = new Object(); object类型

typeof(变量)： 查看变量的数据类型

document.write("c的数据类型是:"+typeof(c)+"<br/>");//c的数据类型是:string

document.write("e的值为："+e +",e的数据类型是:"+typeof(e)+"<br/>");//e的数据类型是:boolean

document.write("f的值为："+f +",f的数据类型是:"+typeof(f)+"<br/>");//f的数据类型是:object

js的数据类型分类有4个：

a）number： 无论整数还是小数，都是number类型

b) string: 无论字符还是字符串，都是string类型

c) boolean

d) object 对象类型

类型转换parseInt（）注意字符串转number的时候要开头是数字，不然转换不了

第一种情况 var a = "10"; 字符串就是数字的时候

document.write("a的类型是(转换前)："+typeof(a)+"<br/>");// a的类型是(转换前)：string

a = parseInt(a);//转换

document.write("a的类型是(转换后)："+typeof(a)+",值为："+a+"<br/>"); a的类型是(转换后)：number,值为：10

第二种情况 a = "22a45b"; 字符串带有数字和字母的时候

a = parseInt(a);//转换 打印：33

第三种情况 a = "cc22a45b"; 开头是字母后续还有数字的时候：打印NaN

运算符 注意 可以number加布尔值：因为 true用1代表 false用0代表

var a = 10; var b = false;

document.write("结果为："+(a/b)+"<br/>");打印：Infinity 属性用于存放表示正无穷大的数值

document.write("结果为："+(a+b)+"<br/>"); 打印：10

if表达式 跟java不同之处

布尔值：true： 成立 ； false:不成立 java只能条件是true或false，下面3个都不支持

number：非0： 成立； 0： 不成立

string: 非空字符串： 成立 ； 空字符串： 不成立

object: 非null: 成立； null：不成立

if(null){ 可这样写

alert("条件成立");

}else{ alert("条件不成立");}

for-in作用 遍历数组 和 对象 类似java增强型for循环

遍历数组

var arr = [10,20,30,40];

for(var i in arr)

document.write(arr[i]+",");

遍历对象

//定义对象

function Person(name,age){

this.name = name;

this.age = age;

}

//创建对象

var p = new Person("eric",20);

//遍历对象

for(var i in p){

document.write(p[i]+"<br/>");

}

# 函数

调用函数步骤 js是没有重载的，同名的函数后面会覆盖前面的

简单两步骤

第一步 声明一个函数

function add(a,b){ / /注意 形参a和b这里不允许var 关键词

docement.write(”a加b结果为：”+(a+b));

}

第二步 调用

add(10+20);

**有返回值：是return关键字**

第一步 声明一个函数有返回值

function add(a,b){ / /注意 形参a和b这里不允许var 关键词

var result=a+b;

return result; //这里使用关键字即可

}

第二步 调用

var s=add(10+20); //因为js是弱编程全部用var接收，这里随意一个变量名称var接收即可

document.write(”返回值是：”+s);

结论 第二步调用 如果赋值少于形参个数会变成 NaN，多了会丢失掉，不会报错

1、js中的形式参数和实际参数的数量可以不一致，依然可以调用

2、js的每个函数中都隐藏了一个叫arguments的数组：这个数组用于接收函数调用时传递过来的实际参数值。arguments数组接收完实际参数后，会逐一的依次从左到右赋值给形式参数,如果实际参数数量大于形式参数，则丢失剩下的实际参数

# js对象

每个对象都包含有原型属性(prototype) 没听懂

什么是原型？

1）js的每个对象中都包含有了原型属性（prototype）

2）如果往某个对象的原型属性添加了方法，那么添加的这个方法就会自动地添加到当前对象的方法中。

3) 原型的作用：给**内置对象**追加方法的

//给内置对象Array追加方法max

Array.prototype.max = function(){ }这样就为a对象追加了一个方法

## 自定义对象

定义对象

定义对象方式一：有参数的构造函数

function 对象名称(形式参数){

定义属性

定义方法

}

创建对象：

var 变量 = new 对象名称(实际参数);

注意 都是需要this的

function Person(name,age){ //this: 表示当前调用的对象

//定义属性 赋值

this.name = name;

this.age = age;

//定义方法

this.say = function(){

alert("这个对象的方法");

}

}

//创建对象操作

var p = new Person("狗娃",12);

p.say() //调用方法

遍历 for-in

定义对象的方式二：使用Object对象。 Object对象可以作为任意对象的模板

//创建对象

var p = new Object();

//追加属性

p.name = "狗蛋";

p.age = 16;

//追加方法

p.say = function(){

alert("狗蛋的函数");

}

定义对象方式三： 无参数的构造函数

//定义对象

function Person(){

}

//创建对象

var p = new Person();

//追加属性

p.name = "狗剩";

p.age = 14;

//追加方法

p.say = function(){

alert("狗剩的函数");

}

定义对象方式四：使用字面量的形式创建对象。 json语言（了解）

//创建人对象

var p = {

//定义属性（属性名：属性值）

"name":"铁蛋",

"age": 20,

//定义方法

"say":function(){

alert("铁蛋的函数");

}

};

//查看属性值

document.write(p.name);

document.write(p.age);

//调用对象方法

p.say();

//使用for-in遍历对象

for(var i in p){

document.write(p[i]+"<br/>");

}

## 内置对象

String

跟java差不多也是有字符串常量池，也可以new

var str =new String(”hello”);

valueOf（）:该方法返回对象的内容

document.write("结果："+(str1.valueOf()==str2.valueOf()));现在相当于equalse()方法对比内容

常用方法：

charAt(): 返回指定索引的内容

indexOf(): 返回首次出现指定字符的索引位置

lastIndexOf(): 返回最后出现指定字符的索引位置

fontcolor(): 直接给字符串添加颜色

replace(): 替换指定的字符串

split(): 使用指定字符串切割字符串，返回字符串数组

substring(start,end); 截取字符串，start：开始索引， end:结束索引

substr(strat,[length]) 截取字符串， start：开始索引 length: 截取的字符串长度

Number

定义Number对象

方式一：

var num1 = new Number(20);

var num2 = new Number(20);

方式二：

var num1 = 20;

var num2 = 20;

document.write((num1==num2)+"<br/>");

document.write(num1.valueOf()==num2.valueOf());

Boolean

var b1 = new Boolean(true);

var b2 = new Boolean(true);

document.write((b1==b2)+"<br/>");

document.write(b1.valueOf()==b2.valueOf());

Math 都是静态方法

常用的方法：

1）ceil()： 向上取整。 如果有小数部分的话，直接+1 Math.ceil(num)

2）floor(): 向下取整。如果有小数部分的话，直接丢失小数部分，保利整数位 Math.floor(num)

3) round(): 四舍五入取整。满5进一

4）random(): 生成一个随机的0-1的小数 .包含0，不包含1

5）max(): 返回最大值

6）min(): 返回最小值

Date

//创建日期

var date = new Date(); //取当前系统日期时间 java: SimpleDateFormat对象 yyyy-MM-dd 格式化

//document.write(date); //默认格式

//格式： 2015年06月01日 15时12分30秒

//年份

document.write(date.getFullYear()+"年");

//月

document.write((date.getMonth()+1)+"月");

//日

document.write(date.getDate()+"日");

//星期

document.write("星期"+date.getDay()+"&nbsp;");

//时

document.write(date.getHours()+"时");

//分

document.write(date.getMinutes()+"分");

//秒

document.write(date.getSeconds()+"秒");

网页时钟

<script type="text/javascript">

function genNewDate(){

第一步 生成一个系统的当前时间，格式：2015-06-01 15:22:30

var date = new Date();

第二步 拼接字符串

var dateStr =date.getFullYear()+"-"+(date.getMonth()+1)+"-"+date.getDate()+"

&nbsp;"+date.getHours()+":"+date.getMinutes()+":"+date.getSeconds();

第三步 把日期字符串放到span中

//得到span对象

var dateTip = document.getElementById("dateTip"); //注意js代码放页面上面会导致页面还没加载完获取不了对象

//设置span的innerHTMl属性给span赋值

dateTip.innerHTML=dateStr;

}

第四步 让函数隔几秒进行运行一次

window.setInterval("genNewDate()",1000);//方法名、毫秒值

</script>

Array

注意：

1）数组的长度会随着元素的添加而变化，不用担心出现索引位置越界的异常。

2) js的数组可以存放任意类型的元素。

方式一：创建数组

var arr = new Array(3); 指定数组长度

var arr = new Array(); 不指定数组长度、默认0个元素

var arr = new Array(10,"hello",true); 指定具体内容

方式二： 创建数组 不需要new，直接指定内容

var arr = [10,"java",false];

给数组赋值

arr[0]=10;

遍历 length属性： 表示数组的长度

for(var i=0;i<arr.length;i++){

document.write(arr[i]+",");

}

常用的方法：

join(字符串): 使用指定的字符串，把数组的所有元素连接在一起，最终组成一个新的字符串

var arr = ["java","net","php","ios"];

var str = arr.join("-");

document.write(str);

reverse(): 反转数组中的元素

arr = arr.reverse();

for(var i=0;i<arr.length;i++){

document.write(arr[i]+",");

}

# 事件编程

java中的GUI编程事件三个要素：

事件源： 按钮JButton JFrame

事件：KeyEvent WindowEvent

监听器：KeyListener WindowListener

GUI编程具体的步骤：

1）创建事件源

2）编程监听器

3）在事件源上注册监听器

javascript事件编程的三个要素：

1）事件源：html标签

2）事件 ：click dblclick mouseover。。。。

3）监听器： 函数

javascript事件分类：

点击相关的：

单击：onclick 双击： ondblclick

焦点相关的：

聚焦：onfocus 失去焦点： onblur

选项相关的：

改变选项： onchange

鼠标相关的：

鼠标经过： onmouseover 鼠标移除： onmouseout

页面加载相关的：onload一般用在body

页面加载： onload这个事件是在加载完标签后再触发。通用用在body标签中，所以加载完body标签的内容触发

例子 选择省份列出城市

js代码

function testChange(){

//alert("选项改变了");

//获取选中的内容

var jiguan = document.getElementById("jiguan").value;

//alert(jiguan);

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

//每次先清空城市select的内容

city.innerHTML = "";

if(jiguan=="广东"){

//把一些option选项放入到城市的select中

var arr = ["广州","珠海","深圳"];

for(var i=0;i<arr.length;i++){

city.innerHTML += "<option value='"+arr[i]+"'>"+arr[i]+"</option>";

}

}

if(jiguan=="广西"){

//把一些option选项放入到城市的select中

var arr = ["柳州","桂林","南宁"];

for(var i=0;i<arr.length;i++){

city.innerHTML += "<option value='"+arr[i]+"'>"+arr[i]+"</option>";

}

}

HTMl代码

请选择籍贯

<!--onchange :表示这个select中的选项发送改变 -->

<select onchange="testChange()" id="jiguan">

<option value="广东">广东</option>

<option value="广西">广西</option>

<option value="湖南">湖南</option>

</select>

城市

<select id="city">

</select>

# BOM编程

BOM就是浏览器对象模型编程，通过javascript引擎提供的四个浏览器对象，操作浏览器，这叫BOM编程

它有四个重要对象：windows 、location、history、screen

window对象(重点) window代表的是一个浏览器窗口

注意：因为window对象使用非常频繁，所以当调用js中的window对象的方法时，可以省略对象名不写。

open(): 在一个窗口中打开页面

<input type="button" value="open()" onclick="testOpen()"/>

function testOpen(){

参数一： 打开的页面

参数二：打开的方式： \_self: 本窗口 \_blank: 新窗口（默认）

参数三： 设置窗口参数。比如窗口大小，是否显示任务栏

window.open("02.广告页面.html","\_blank","width=300px;height=300px;toolbar=0");//可以少参数的

}

定时器

setInterval(): 设置定时器（执行n次）

clearInterval(): 清除定时器

连续执行让定制的方法跳转页面，然后进行清除定时器操作

<script type="text/javascript">

function testOpen(){ //第一步 要被执行的方法

window.open("a.html","\_blank","width=300px;height=300px;toolbar=0");

}

var taskId;

function testInterval(){第二步 设置方法定时器在页面设置时间：指定方法要被执行，返回一个唯一的id

/\* 定时器： \*/

taskId=window.setInterval("testOpen()",5000);

}

function testClearInterval(){ 第三步 设置一个onclick点击事件，进行清除

/\*清除任务 需要id \*/

window.clearInterval(taskId);

}

</script>

setTimeout(): 设置定时器(只执行1次)

clearTimeout(): 清除定时器

效果和上面两个方法一样 只执行一次，不写了

提示框

alert(): 提示框

confirm(): 确认提示框 返回布尔值

function testConfirm(){

返回值就是用户操作true: 点击了确定，false： 点击了取消

var flag = window.confirm("确认删除吗？一旦删除不能恢复，请慎重！");

if(flag){

alert("确定删除，正在删除中....");

}else{

alert("取消了操作");

}

}

propmt(): 输入提示框

function testPrompt(){

输入提示框

var flag = window.prompt("请输入你的U顿密码");

if(flag){

alert("密码正确，转账中...");

}else{

alert("取消了操作");

}

alert(flag);//打印：你输入的内容

}

location对象 location对象表示一个窗口中的地址栏，它属于windows

href属性： 代表的是地址栏的URL，可以获取和设置URL。URL表示统一资源定位符

reload方法： 刷新当前页面

function testHref(){

alert(window.location.href); //获取超链接的 href值

window.location.href="02.广告页面.html";//通过修改location对象的href属性来实现页面的跳转

}

function testReload(){ //刷新当前页面

window.location.reload();

}

//function testRefresh(){

//定时刷新

window.setTimeout("testReload()",1000);

//} history对象表示窗口的历史记录栏

forward(): 前进到下一页

back(): 后退上一页

go(): 跳转到某页（正整数：前进 负整数：后退） 1 -2

function testForward(){

//window.history.forward();

window.history.go(1);

window.history.go(-1); 后退

}

screen对象 代表是一个屏幕 指的是整个显示屏

availHeight和availWidth是排除了任务栏之后的高度和宽度

document.write(window.screen.availWidth + "<br/>");

document.write(window.screen.availHeight + "<br/>");

document.write(window.screen.width + "<br/>");

document.write(window.screen.height + "<br/>");

# DOM编程

DOM（document Object Model）文档对象模型编程，document代表一个html页面

在浏览器加载完html页面：javascript引擎把整个html页面的所有html标签封装成对象

获取标签对象

获取页面的指定的所有标签 ：返回数组

all: 获取所有标签对象

forms： 获取form标签对象

images: 获取img标签对象

links: 获取a标签对象

//var nodeList = document.all; //返回标签对象数组

//var nodeList = document.forms; //返回所有标签form数组

//var nodeList = document.images; //返回素有images数组

var nodeList = document.links;//返回a标签对象数组

//alert(nodeList.length);

//遍历标签对象

for(var i=0;i<nodeList.length;i++){ //nodeName：获取标签名称

alert(nodeList[i].nodeName);

}

节点关系获取标签对象

父节点： parentNode属性 父亲节点就一个

var aNode = document.links[0]; //得到a标签（获取第一个a标签对象）

//alert(aNode.nodeName); //打印标签名字 A

var formNode = aNode.parentNode; //得到父标签

alert(formNode.nodeName);

子节点： childNodes属性

第一个子节点： firstChild属性

最后一个子节点： lastChild属性

下一个兄弟节点： nextSibling属性

上一个兄弟节点： previousSibling属性

通过 id 、name、标签名字

1)document.getElementById("id属性值"); 最常用

注意： 1）使用该方法获取的标签一定要有id属性。2）在同一个html页面中不要出现两个同名的id

2)documetn.getElementsByName("name属性值"); 获取同name属性名的标签列表

注意： 1）使用该方法获取的标签一定要有name属性

3)document.getElementsByTagName("标签名") 获取相同标签名的标签列表

修改标签对象属性

常用的需要修改的属性：

innerHTML属性：修改标签体内容 <span>xxxxxx</span> <div></div> <select></select>

innerText: 设置标签内容的文本

value属性： 修改value属性值。 input type="text"

src属性： 修改图片的src属性。 <img src=""/>

checked属性：修改单选或多项的默认值。 <input type="radio/checked" checked=""/>

<script type="text/javascript">

var mySpan = document.getElementById("mySpan");//获取span标签对象

mySpan.innerHTML = "<h3>标题</h3>";//设置标签内的HTML内容， 在IE和火狐可以

//mySpan.innerText="标签"; /\*innerText在IE可以，在火狐不行\*/

//获取input对象

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

userName.value = ""; //设置value值操作

function changeImg(){

var img = document.getElementsByTagName("img")[0]; //得到标签对象

img.src="mm.jpg"; //设置src属性值操作

}

标签对象添加方法

添加相关的：

document.createElement("标签名") 创建节点对象

setAttribute("name","value"): 设置节点的属性

插入操作：

appendChild("标签对象") ; 添加子节点对象

insertBefore("新标签对象","指定的对象") 在指定的对象前面添加子节点对象

删除：

removeChild("标签对象"); 删除子节点