# 第一天

## 回顾：

Html，CSS：标记语言

JavaScript：编程语言

**JavaScript：JS，基于客户端的脚本语言**

Day1：变量

Day2：循环，语句

Day3：函数

Day4：系统内部的函数

Day5-Day8：案例项目

## 变量：用来保存东西（数据）

### 语法：var 变量名=值;

var n=23;

var m=24;

Var，是系统内部的关键字，用来申明变量

### 变量名

1. 变量名不能以数字开头
2. 变量名只能包含数字、字母、下划线
3. 区分大小写
4. 变量名不能和系统内部关键字相同
5. 建议：变量名要起得有意义

## 数据类型：

数字类型：number，数字

字符类型：string，字母，汉字，符号“内容”，空字符也是字符类型

空类型：null，表示什么都没有

未定义类型：undefined：表示没有定义

布尔类型：true,false；英文是boolean（bool）

复合数据类型：复杂类型，array数组，object对象

### Typeof，查看数据类型

Alert：（警告）弹出一个窗口（会停止程序的运行）

### Firebug：调试

注意：不要太相信firebug

1. 安装firebug
2. 控制台（点击启用）
3. 刷新
4. 找错

### 运算符：

1. 赋值运算符：符号【=】，表达的意思是把符号右边的数据赋值给左边的变量名，

var n=23

1. 算数运算符：符号【+，-，\*，/，%（取余），++（递增1），--（递减1）】
2. %，取余：取得两个数相除的余数
3. ++：var i=2;i++; 给变量i加1.（++i，自加1）
4. --：var i=2;i--; 给变量i减1.（--i，自减1）

注意：当有赋值运算符参与到++和—的时候，i++和++i的含义就不同了

1. var j=i++; 先把i的值赋值给j，之后i再自加1；
2. var j=++I; 先把i自加1，之后再赋值

练习++：

Var i=6; i=6

Var j=i++; j=6; i=7

Var m=j++; m=6; j=7

Var n=++i; n=8; i=8

Var a=m++; a=6; m=7

Var b=m++; b=7; m=8

Var ab=++b; ab=8; b=8

【i=8, j=7, m=8, n=8, a=6, b=8, ab=8】

### 比较运算符

符号【>, <, >=, <=, ==（等于）, !=（不等于）, ===（全等于/恒等于/绝对等于）, ！==（不全等于）】

==等于：判断两个值是否相同（仅仅是判断值）

===全等于：判断两个数据是否相同（判断值和数据类型）

练习比较运算符：

### 逻辑运算符

符号【！，&&，||】

!; 非，不是

&&; 逻辑与：两个条件的满足情况，如果两个条件同时满足为true，如果两个条件中有任意一个为fasle，结果就为false

||：逻辑或：两个条件满足情况，如果两个条件中有一个为true，结果就为true，只有当两个条件都为false的时候结果为false

### 复合赋值运算符：

符号【+=，-=，\*=，/=】

+=：var i=6; i+=6;【i=i+6】

-=：var i=6; i-=6;【i=i-6】

\*=：var i=8; i\*=2;【i=i\*6】

/=：var i=8; i/=2;【i=i/2】

练习：

M+=3；

N\*=2;

A+=9;

b-=6;

### 字符串运算符：

连接符号两边的内容

当+两边，任意有一边是字符串，就会执行字符串+

符号【+】

23+24；

23+“24”【2324】

## JS的语法规则：

1. 区分大小写
2. 所有的符号都必须是英文状态下的
3. 每行表达出所要表达的意思即可
4. 必须：每行后面必须加上分号（英文状态下的分号）

### JS的注释：

注释就是给某段代码加上标注或者说明，记录

注释的内容不会被代码执行

两种：

1. 单行注释//：符号//后面的内容都是注释内容
2. 多行注释/\*\*/：符号/\*内容\*/中间的都是注释内容

alert：弹出警告窗口

### document.write()：输出到页面上

## 位运算符：了解

二进制：0，1

十进制：0，1，2，3，4，5，6，7，8，9

&：获取两个数的二进制位，比较相同位数上的数字，如果都为1，则为1，否则为0

var n=8&9

8&9=8

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 十进制 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 二进制 | 1 | 10 | 11 | 100 | 101 | 110 | 111 | 1000 | 1001 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 8的二进制 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 9的二进制 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 计算& | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

<<：把某个数字的二进制位数同时左移n位

Var n=9<<2

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 9的二进制 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 左移两位 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 计算成十进制 |  |  | 1\*2^5 | 0 | 0 | 1\*2^2 |  |  |
| 计算成十进制 |  |  |  |  |  |  | 3 | 6 |

n=1\*2^5 + 0\*2^4 + 0\*2^3 + 1\*2^2 + 0\*2^1 + 0\*2^0

=32 + 0 + 0 + 4 + 0 + 0

=36

1024:

千位：1000，【1\*10^3】

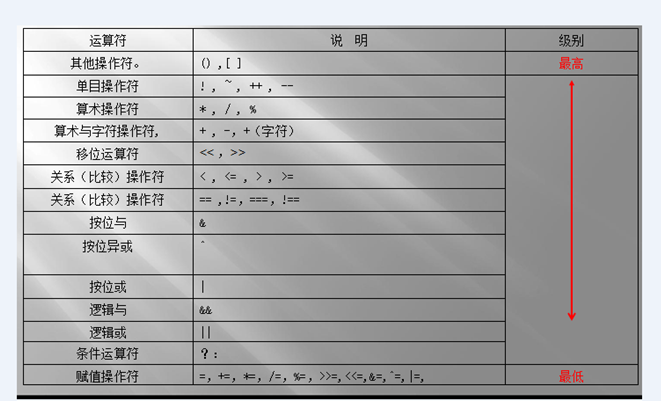
百位：0，【0\*10^2】

十位：20，【2\*10^1】

个位：4，【4\*10^0】

运算符的优先级：

1. 小括号最优先
2. 逻辑与 大于 逻辑或
3. 赋值运算符最后进行
4. 当有多个运算符参与运算的时候，脑子里必须有意识地知道运算符是有优先级的



练习1：

1. 在页面上输出内容【用JS输出】
2. 页面输出h标签【用JS输出】

9\*(2+3)-9\*(8+4)

# 第二天

### 传值：把一个变量传递给另外一个变量

var 变量名=值;

var n=3;

var m=n;

var a={name=”张三”,age:24}

var b=a;

1. 赋值传值：两份数据，两份数据之间不会影响
2. 引用传值：一份数倨，任意一个改动值另外一个都会受到影响

注意：在简单数据的传值中，都是赋值传值；而复杂数据都是引用传值

## 流程控制值if语句：

### 1，if语句

语法：if(条件){语句块【一些代码】}

执行：判断条件是否满足，如果条件满足就会执行语句块，如果条件不满足，就不会执行语句块的内容

注意点：很有可能不执行

### 2，if,else语句

语法：if(条件){语句块1}else{语句块2}

执行：判断条件是否满足，如果条件满足，就会执行语句块1；如果条件不满足，就执行语句块2

注意：这种语句必定会有一个语句块执行

### 3，if,else if,else if语句

语法：if(条件1){语句块1}else if(条件2){语句块2} else if(条件3){语句块3}……

执行：判断条件1是否满足，如果条件1满足，就会执行语句块1；如果条件1不满足，就会判断条件2；如果条件2满足，就执行语句块2；如果条件2不满足，就会判断条件3；如果条件3满足，就执行语句块3；如果条件3不满足，….

注意：这种语句有可能不执行

### 4，if,else if,else if语句

语法：if（条件1）{语句块1}else if（条件2）{语句块2}else｛语句块3｝

执行：判断条件1是否满足，如果满足就执行语句块1，如果不满足就会判断条件2；如果条件2满足就会执行语句块2，如果不满足就会执行语句块3【前面条件不满足就会执行】

注意：这种语句必定会有一个语句块执行

## 练习 if，else语句

学生成绩的案例

如果成绩大于或等于90就输出优秀，

如果成绩大于或等于80就输出良好，

如果成绩大于或等于70就输出一般，

如果成绩大于或等于60就输出及格，

其它就输出不及格

## switch语句

语法：

switch(变量n){

case值:

document.write();

break;

case值:

document.write();

break;

……

default:

document.write();

}

执行：变量n和值1比较，如果相同，就会执行值1后面的内容；如果不相同就会比较变量n和值2是否相同；如果n和值2相同，就会执行值2后面的内容，如果不相同就会继续比较下去；如果没有值和变量n相同，就会执行default的内容

### 小知识点：if语句之间可以互相嵌套

var n=2;

var m=3;

if(n==2){

if(m==4){

alert(“aaa”)

}

}else{

语句块2

}

## 循环

循环三要素：

1. 循环变量的初始化
2. 循环条件的判断
3. 循环变量的改变

### while循环

语法：

var=1; //循环变量的初始化

while(i<=100 //循环条件的判断){

//循环体

document.write(i)

//循环变量的改变

i=i+5;

}

while(){}

执行：

循环变量初始化=》循环条件的判断（true）=》循环体=》循环变量的改变=》循环条件的判断（true）=》循环体=》循环变量的改变=》循环条件的判断（false）=》跳出循环

注意：如果循环条件判断为false，那么就不会执行

练习

### 循环输出1到100之间所有的数字

### 循环输出1到100之间所有的偶数

### 求1到100之间所有数字的和

求和思想：

1. 找一个变量用来记录总和
2. 把前一次的和加上新的值再赋值给变量

### 求1到100之间偶数的个数

记数思想

## do whlie循环

语法：

var i=1;

do{

//循环体

//循环变量的改变

i++;

}while(i<=100)

执行：循环变量的初始化=》立刻执行一次循环体=》循环变量的改变=》循环条件的判断（true）=》循环体=》循环变量的改变=》循环条件的判断（false）=》跳出循环

注意：必定会执行至少一次

### for循环

语法：

for(【循环变量的初始化】;【循环条件的判断】;【循环变量的改变】){

【循环体】

}

执行：

【循环变量的初始化】=》【循环条件的判断】（true）=》【循环体】=》【循环变量的改变】=》【循环条件的判断】（true）=》【循环体】=》【循环变量的改变】=》【循环条件的判断】（false）=》跳出循环

## for循环的案例

## for循环的扩展（1到10，10到1）

for循环的嵌套

for ( ) {

for ( ) {

}

}

### 案例：九九乘法表

双引号中不能套双引号，可用单引号

### 案例：星星金字塔

# 第三天

函数：一段能够自动完成某些功能的代码

### 定义语法：

function 函数名(形式参数1,形式参数2,……){

//一些代码

函数体

}

### 调用语法：

函数名(实际参数1,实际参数2,…….);

注意：函数在写好之后是不会自己去执行，需要调用函数才会执行；

注意：函数执行完成之后会回到调用函数的位置

一般来说函数写好之后不会经常去改动函数内部的代码

### 函数的练习：

求任意一个数为圆的半径，求圆的面积，假设pai为3.14

1. r=6
2. r=9

求已知任意两个值为直角三角形的两条直角边的长度，求斜边的长度

## 函数名：

注意：不要和系统函数以及关键字相同

建议：函数名要起得有意义

## 函数的执行

1. 点到firebug的脚本
2. 点击某个位置添加断点
3. 刷新
4. 点击单步进入

## 作用域

对于函数来说，函数内部定义的变量叫做局部变量；

函数外部的变量叫做全局变量；

### 只针对函数来说的

局部变量：只能在函数内部使用

全局变量：可以放到任何位置去用

### ，平时只会关注局部变量，只要在函数内部定义变量就必须有意识地知道，存在作用域的问题

## 返回值：return

### return：跳出函数，不再执行.

### return 值：跳出函数，同时把值返回调用函数的位置

### 注意：如果有返回值，必须接收返回值

## 函数的位置：

可以先定义，后调用

也可以先调用，后定义

练习：

求1到任意一个数n（n>=1）之间的所有的偶数的个数

九九乘法表，m份数+n\*n乘法表

## 数组

### 属性：length（求得数组的长度）

var arr[23,24,25];

var len=arr.length;

var a=23;

var b=24;

var c=26;

var arr=[23,24,26]

数组是从0开始记录的

数组下标的最大值，从0开始，到数组个数-1结束

如果数组有20个数据，那么最大下标为20-1=19

如果数组有36个数据，那么最大下标为36-1=35

### 取值：arr

|  |  |  |  |
| --- | --- | --- | --- |
| 数组 | 23 | 24 | 26 |
| 下标 | 0 | 1 | 2 |

arr[0]

### 赋值：arr[下标]=29;

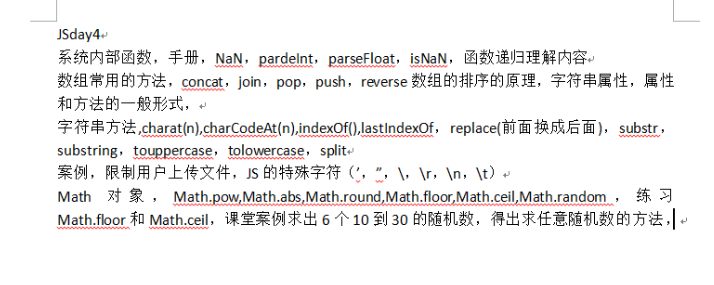
### 练习：

求一个数组中有六个数字，求这些数字的和

var arr=[24,26,22,24,25,29]

求一个数组中六个数字的最大值/平均值

# 第四天



var arr=[值1, 值2, 值3 值4,……]

通过下标去找到数组中某个值【下标：从0开始，到arr.length -1结束】

## 二维数组

var arr1=[11,12,13];

var arr1=[21,22,23];

var arr1=[31,32,33];

var arr=[arr1,arr2,arr3];

var arr=[

[11,12,13],

[21,22,23],

[31,32,33],

]

|  |  |  |  |
| --- | --- | --- | --- |
| 第一行 | 11【0】【0】 | 12【0】【1】 | 13【0】【2】 |
| 第二行 | 21【1】【0】 | 22【1】【1】 | 23【1】【2】 |
| 第三行 | 31【2】【0】 | 32【2】【1】 | 33【2】【2】 |

var n=arr[2][2];

var m=arr[1][2]

### 练习：

### 求一个三行四列的数组的所有项的和

### 求最大值及最大值的下标

## 系统内部的函数

### parseInt：取整（如果字符串开头第一个就是字母【数字以外的符号】，那么就会出现NaN）

parselnt(24.2);

parselnt(“25.2”);

parselnt(“26.27abcd”);

parselnt(“28.ab.29.34”);

parselnt(“31ab”);

parselnt(“ab32”);

### parseFloat：取浮点数

parseFolat(11.11);

parseFolat(“22.22”);

parseFolat(33.33ab);

parseFolat(“ab44.44”);

parseFolat(“0ab.55”);

parseFolat(05ab.55);

NaN: not a number

### isNaN: is not a number（是 “非数字”）

1. 如果判断的数据为数字：false
2. 如果判断的数据为字符：true

## Array对象：

var arr=[22,24,26];

var arr=new Array(22,24,26); // Array中的“A”要大写

### 属性：

length：

arr.length：获取数组的长度

### 方法：

concat() 连接内容或者数组，组成新的数组

join() 加入某个字符n，用n连接数组的每项，组成字符串，如果以空字符串连接，那么就不会加入的内容，直接把数组的每项内容连接

pop() 删除数组的最后一项并且返回

push() 加入新的内容，返回新的长度

reverse() 反转数组顺序

属性和方法的一般形式：

属性：对象.属性名

方法：对象.方法名()

## string对象

var str=”abcdefgh”;

var str=new String(“abcdefgh”); //这里的“String”中的“S”要大写

## 属性：

length:

对象.length;

## 方法：

charAt(n) 找到索引位置上的字符

charCodeAt(n) 找到索引位置上的字符的编码值：a=97，A=65

indexOf(“m”) 找m在字符中第一次出现的位置；如果没有找到，就返回-1

lastIndexOf(“m”) 找m在字符中最后一次出现的位置；如果没有找到，就返回-1

split(“n”) 以字符n分割字符串，返回一个数组（如果string中没有n，则返回一个数组），以空字符串分割会把字符串每个字符都当做数组的一项内容，如果不存在n那么就会把整个字符串当做数组的内容

substr(n,m) 截取字符串：把字符串从n（下标）的位置开始截取，截取m个；如果只有一个参数n，那么就是从n开始，截取到最后

substring(n,m) 截取字符串：把字符串从n（下标）的位置开始截取，截取到m（下标）的位置；注意：能够取到n，但是取不到m；如果只有一个参数n，意思就是从n开始截取到最后

toLowerCase() 把字符串中的字母全部转换成小写形式

toUpperCase() 把字符串中的字母全部转换成大写形式

## 练习

检测用户上传的文件是否符合要求

jpg，gif，png

document.getElementById(“id名字”)

onclick

## Math对象

Math.pow(n,m); 求n的m次方

Math.abs(n); 求n的绝对值

Math.round(n); 求n四舍五入的值

Math.floor(n); 求n的向下取整的值：不大于n的最大整数（小于等于）

Math.floor(2.3)==2

Math.floor(2.5)==2

Math.floor(2)==2

Math.floor(2.9)==2

Math.ceil(n); 求n的向上取整的值：不小于n的最小整数（大于等于）

Math.ceil(2.3)==3

Math.ceil(2.6)==3

Math.ceil(2)==2

Math.ceil(2.1)==3

Math..random(); 获得一个o到1之间的随机数（16位左右的小数），注意：能取到0，但是不能取到1

1. 求0到20之间的随机数：Math.random()\*20
2. 求0到60之间的随机数：Math.random()\*60
3. 求0到90之间的随机数：Math.random()\*90

求0到n之间的随机数：Math.random()\*n

1. 求10到20之间的随机数：
2. 想办法变成0到某某数字：0到10再加上10
3. 再变换成10到20之间的随机数 Math.random()\*10+10
4. 求30到60之间的随机数：Math.random()\*30+30
   1. 先把整体-去30，就变成0到30 ，再+30
   2. Math.random()\*30+30
5. 求60到90之间的随机数
   1. 先把整体-去60，就变成0到30 ，再+60
   2. Math.random()\*30+60

求n到m之间的随机数（n<m）（取不到m）：

Math.random()\*(m-n)+n

### 练习：

1. 在页面输出6个10到30之间的两位数（不包括30，则向下取整Math.floor；或者取整parseInt）
2. 要求取到30（四舍五入）

1和0的问题：

得到0的概率：0-0.5之间

得到1的概率：0.5-1.5之间

得到10的概率和得到30的概率要小于其它数

1. 要概率相同
   1. 求9.5~30.5之间的四舍五入整数：用 var n = Math.round(Math.random() \* 21 + 9.5);
   2. 求10到31之间的向下取整数，用var n = Math.floor(Math.random() \* 21 +10);
   3. 求9到30之间的向上取整数，用var n=Math.ceil(Math.random()\*21+9);

### 公式

求n到m之间的随机数（n<m）（取到n和m）：

Math.floor(Math.random()\*(m-n+1)+n)

## JS特殊字符

‘：单引号

“：双引号  
 \：转义字符

\r：回车

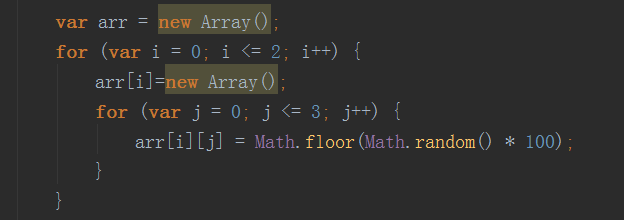
\n：换行

练习：

1. （二维数组，逻辑，偏难）定义一个二维数组（3行4列），输入任意12个数值，求最大值，最小值，两个下标均为奇数的数的和
2. （二维数组）定义一个二维数组（自定义，3行n列），求得所有数值的各以及平均值（注意：列数不固定）

定义二维数组

注意：var= new Array(); 定义的是一个一维数组



# 第五天-案例项目

## Date对象：

1. var n1=new Date(); //创建当前时间日期对象（当前电脑默认时间）
2. var n2=new Date(“1999/9/9 6:6:6)”; //创建一个指定的时间日期
3. var n3=new Date(1999,9,9,9,9,9); //创建一个指定的时间日期对象
4. var n4=new Date(2343523445632); //创建一个指定的时间日期对象

toLocalString 以当前本地格式显示

getFullYear(); //获取年份

getMonth(); //获取月份

getDate(); //获取日期

getHours(); //获取小时

getMinutes(); //获取分钟数

getSeconds(); //获取秒数

getMillSeconds(); //获取毫秒值

getDay(); //获取星期 （如果是星期天，则返回0）

getTime(); //获取从1970年开始到（创建的）时间日期的毫秒值

setFullYear(); //设置年份

setMonth(); //设置月份

setDate(); //设置日期

setHours(); //设置小时

setMinutes(); //设置分钟

setSeconds(); //设置秒数

setMillSeconds(); //设置毫秒数

~~setDay(); //设置星期~~ 没有这一项，因为星期由日期决定，不由个人设置

## 练习

计算到现在为止你所生活了多少天

## 对象：一类中的具体的事例

var obj={name:”张三丰”, age:24,taiji:function ab(){}, zuofan:function(){}}

属性：name, age

方法：taiji, zuofan

在JS角度看，标签都是对象

IE: JavaScript; jsctipt

LiveScipt=>javascript=>ECMAscript, BOM, DOM

## 标签：HTML属性，CSS属性

HTML：对象.HTML 属性=值

CSS：对象.Style 属性=”值”

## 获取对象的方法：

id: document.GetElementById(“id名”)

object HTMLInputElement

## 事件：

用户的动作

onclick：点击

onmouseover：鼠标放上

onmouseout：鼠标离开

### 练习：

### 修改div样式

### 改变图片大小

### 开关灯

### 修改div的内容

innerHTML：对象中所有的内容（文本内容和标签内容），指的是双标签，容器标签

innerText：对象中所有的文本内容

## 案例：网页版计算器

在页面所获取的内容都是字符串

## 事件：

### 鼠标事件：

onclick：点击

onmouseover：鼠标放上

onmouseout：鼠标离开

ondblclick：双击

onmousedown：鼠标按下

onmouseup：鼠标抬起

onmousemover：鼠标移动

### 表单事件：

onfocus：获得焦点

onblur：失去焦点

onsubmit：提交事件

onchange：当发生改变的时候

onreset：重置事件

### 键盘事件

onkeyup：键盘抬起

onkeydown：键盘按下

onkepress：键盘按键一次

### 窗口事件 onload事件

页面加载完成之后立刻执行的事件(不需要人为点击，就自动执行)

两种方式：

1. <script>window.onload=init;</script>

init是函数名，不要加()

1. <body onload=”init()”></body>

这里的init要加()

### Event：保存事件发生时的相关的信息

当事件发生的时候，event

event.clientX：事件发生时的X的坐标

event.clientY：事件发生时的Y的坐标

event.target：事件源

event必须通过实际参数的形式传递给函数，才可以用

### onchange 发生改变时触发的事件

## 练习：

### 窗口事件

### 案例：移动瞄准案例

# 第六天

## 对象：一类中的具体的实例

<标签>

document.creatElement(“标签名”)

document.body.appendChild(对象);

document.body：是body标签对象

document.documentElementHtml：标签对象

注意：margin不会触发事件

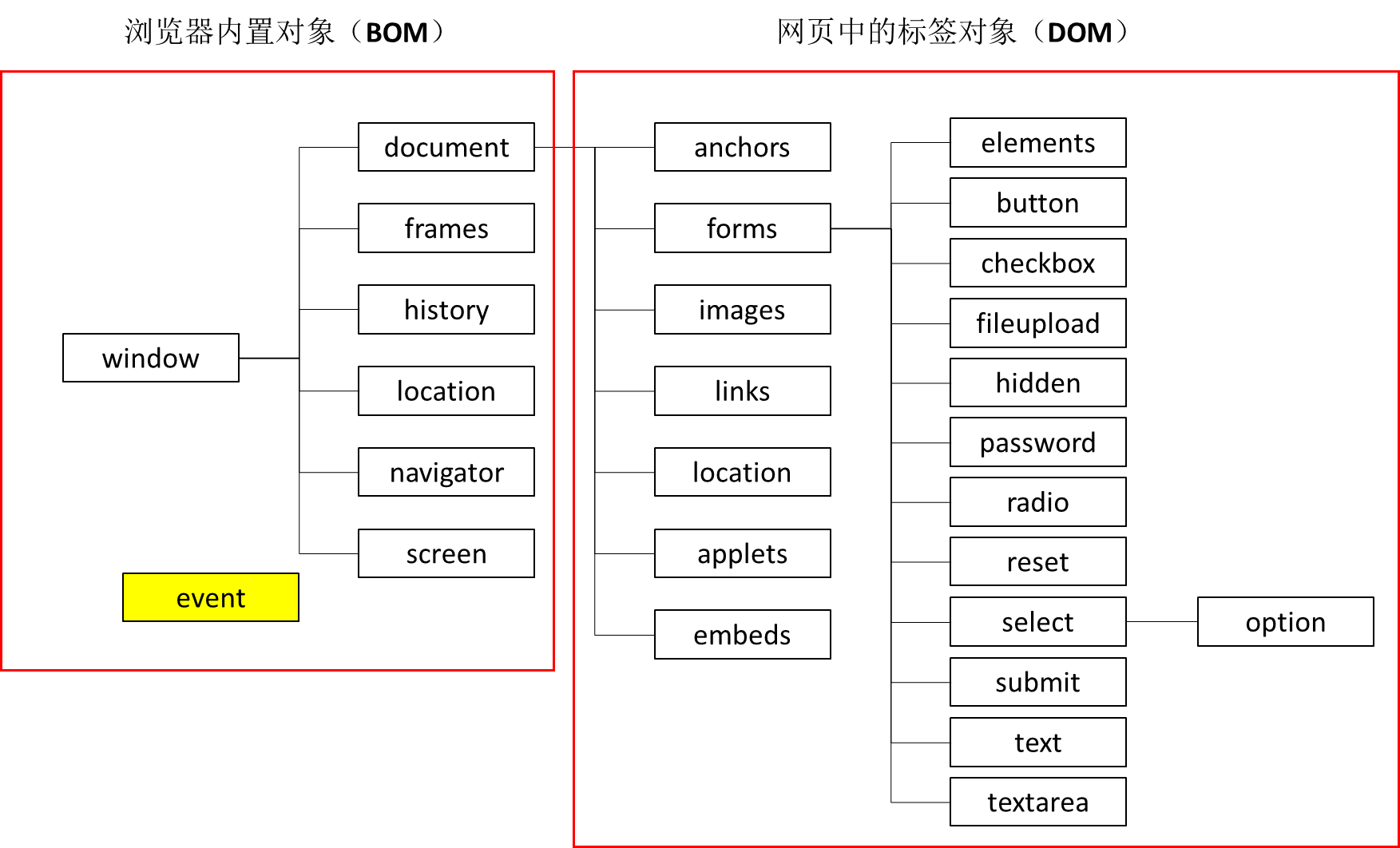
案例：点出满天小星星

1. 让页面点出星星
2. 让星星随机大小并且随机位置显示

## this：代表的是当前事件的事件源

4a++；结果为NaN，not a number

“44”++：结果为数字，可加



## ECMAsrcipt，BOM，DOM

发现：

1. window对象最高级（最高级）
2. BOM：浏览器对象模型：browser object model
3. DOM：文档对象模型：document object model
4. BOM 网页一打开就会存在
5. DOM 去操作的代码部分
6. document是连接BOM和DOM

document有下级，其它都没有下级【有的是多窗口】

## window：window的方法

### window的三个弹出框

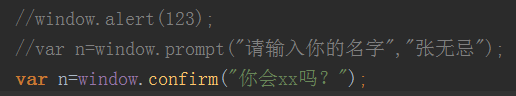
1. alert()：弹出一个窗口
2. prompt()；一个可以让用户输入的窗口,加入提示性语句，也可以加入默认值；

如果用户不填写信息，点击确认确认得到的是空字符串；

如果用户点击取消得到的是null

1. confirm()；弹出一个可以让用户选择的窗口；

如果点击确认得到的是true；如果点击取消得到的是false



### 练习：

## window.open：打开一个新窗口

window.open(“链接”, “name”, “设置”);

1. width：设置窗口宽度.
2. height：设置窗口的高度
3. left：新窗口到左端的距离
4. top：新窗口到顶部的距离
5. scrollbar：滚动条【yes,no; 1,0】
6. toobar：工具栏【yes,no; 1,0】
7. location：地址栏

window.close(); 关闭窗口

## 定时器

### 一次性定时器：var abcd = window.setTimeout(“js代码”,时间t)

执行：是在时间t之后执行代码【只会执行一次】

时间：以毫秒为单位

### 反复性定时器：var dsq = window.setInterval(“js代码”,时间t)

执行：是每过时间t就会执行一次js代码【n次】

时间：以毫秒为单位

清除定时器：

清除一次性定时器：window.clearTimeout(abcd);

清除反复性定时器：window.clearInterval(dsq)

注意：要想清除定时器，必须定时器名字，匿名定时器无法清除

# 第七天

## 图片切换案例

1. 定时器：反复性定时器
2. 页面加载完成之后都会去切换图片：onload事件
3. 鼠标放到图片上，图片停止【鼠标离开图片，图片就会自动切换】
4. 鼠标放到列表list上，图片停止到对应的图片，选中的列表list背景颜色要变换
5. 鼠标离开列表，图片自动切换，对应的列表list背景颜色也自动切换

## TAB浏览案例（标签式浏览）

要求：鼠标放到各新闻类别的tab上，下面的新闻标题部分自动切换

原理：onmouseover（鼠标放上）时，其它的tab不显示，只有选中的tab显示

## 找对象的方法：

id：document.getElementById(“id名”);

标签：document.getElementsByTagName(“标签名”);

获取的是对象的集合/ 相当于一个数组

扩展 某对象.getElementsByTageName(“标签名”);

name：document.getElementsByName(“name名”);

form表单上的 name 集合/ 数组

className：document.getElementsByClassName(“class名”);

集合/数组【只在firefox浏览器中能用】

document.images; 获取img对象/ 数组

document.links; 获取link对象/ 数组

document.forms; 获取表单对象/ 数组

document.body; body标签对象

document.doucumentElement; HTML对象

event; 事件信息对象

this; 当前对象

注意：id比较常用

id > this，event > 标签

# 第八天

## Location对象

location.url; 返回url信息【可以获取url信息；也可以给其赋值，实现页面跳转】

location.assign(); 加载新的文档【跳转页面】

location.reload(); 重新加载当前的文档【刷新】

location.replace(); 用新的文档替换当前的文档【跳转】

### location.assign();和location.replace();的区别：

location.assign(); 会产生历史记录

location.replace(); 不会产生历史记录

### 练习案例：

实现页面自动跳转

## History对象

history.length; 获取浏览过的url数量

history.back(); 返回历史记录的前一个页面

history.forward(); 加载历史记录中的下一个页面

history.go(n); 跳转到历史记录中指定中的页面，如果是-1，实际上就是history.back的功能

## Screen对象

screen.height; 获取屏幕的高度

screen.width; 获取屏幕的宽度

screen.availHeight; 获取除去任务栏的高度

screen.availWidth; 获取除去任务栏的宽度

## Navigator对象

**Navigator 对象包含有关浏览器的信息。**

navigator.appName; 返回浏览器的名称【除了IE是MSIE，其他一般都是Netscape】

navigator.appCodeName; 返回浏览器的代码名称【大都是mozilla】

navigator.appVersion; 返回浏览器的版本号和平台信息

navigator.Agent; 返回由客户机发送服务器的 user-agent 头部的值。【浏览器的大部分信息】

### 练习：

查看用户所用的浏览器是什么

## DOM：描述网页各个组成部分之间的关系

var obj=document.getElementById(“id名”);

火狐浏览器中空白处也算一个节点

parentNode：父节点

childNodes：子节点【数组】

firstChild：第一个子节点

lastChild：最后一个子节点

nextSibling：下一个兄弟节点【注意：必须是同父级关系】

previousSibling：上一个兄弟节点【注意：必须是同父级关系】

### 练习案例：

### 星星案例：

1. 网页打开后随机大小、位置出现星星
2. 点击星星，让星星消失
   1. 绑定一个onclick事件：

对象.事件=事件处理函数

【注意，没有函数没有”()”】

obj.onclick=abcd;

* 1. removeChild(obj); 删除节点

【注意：要想删除某个节点，必须找到它的父节点】

【注意：在绑定事件中，this可以直接使用】

1. 游戏功能添加
   1. 游戏开始和暂停
   2. 当页面上星星个数大于20个时，游戏结束
   3. 游戏进度条