JS视频笔记

事件 用户操作

document.getElementById()处理兼容性问题，不能直接用id9

网页换肤，两套css链接<link>中的href link可以加id

function skin1(){

var ol = document.getElementById("");

ol.href = 'css1.css';

}

function skin2(){

var ol = document.getElementById("");

ol.href = 'css2.css';

}

再调用函数

任何标签都可以加id

任何标签的任何属性js怎么写

显示隐藏if(div显示){div隐藏}else{div显示}

传参时传字母" 'abx' "，单引号。

function setcolor(color){

。。。。.background=color

}

调用....setcolor(yellow);

函数就是个占位符。函数里有一部分定不下来，需要传参。

设置一个css, .class名，用js加入到html中，变量

名.className="class名";

第一种操作属性的方法 .

第二种操作属性的方法 [' ']好处：参数可以变

例子：setText(name,value){

oDiv.style[name] =value;

}

调用 setText(width,'100px');

能用点的都能用[]，[]不一定会适用点。

字面量，看到的东西，自然之道是什么。'ab'

变量 var a = ?;a是不一定固定值。

不带引号，就只能当成变量，会报错。

带引号是字符串。

元素.style.属性=xxx;是修改行间样式

之后再修改className不会有效果

要么统一使用style，要么统一使用class

样式优先级

\* < 标签 < class < id < 行间

提取行间事件

提取事件

为元素添加事件：

如果函数没有名字，则为匿名函数。

Function 名字(){ }

oBtn.onclick = 名字;

oBtn.onclick = function(){ }匿名函数

Js读一行加载一行。

Window.onload 页面加载完成时发生。

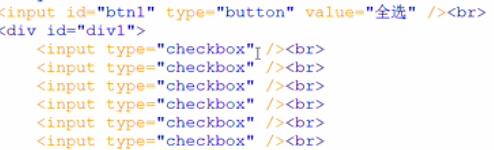


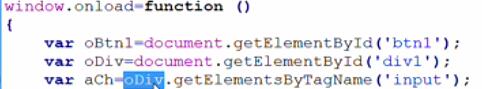
行为，样式，结构三者分离

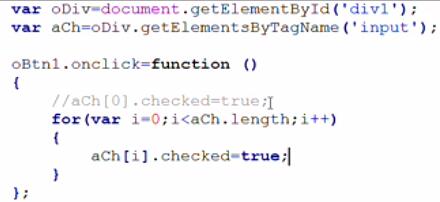
Js css html三者分离

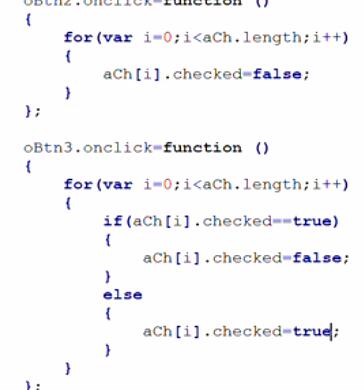
获取一组元素

getElementsByTagName









改变<div>背景，**aDiv.length适应div个数**

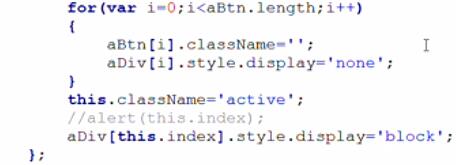
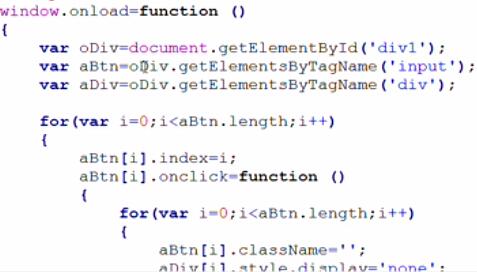


数组，Length属性

Js中一次只能给一个元素设样式。

This 当前发生事件

选项卡先清空所有按钮，再选中当前按钮，内容实现，先隐藏所有DIV，在现实当前DIV



innerHTML

索引值的使用

什么时候使用索引值，需要知道第几个的时候

Html中添加index，会被过滤，js中添加不会过滤。

Alert(‘abc’+(12+5)+’def’);输出abc17def

Alert(‘abc’+12+5+’def’);输出abc125def

JS组成

1.ECMAScript 翻译 解释器

2.DOM Document Object Model HTML documentL(对象)

操作HTML的能力

3.BOM Bromser Object Model 浏览器 window(对象)

兼容性

ECMA 几乎没有兼容性问题

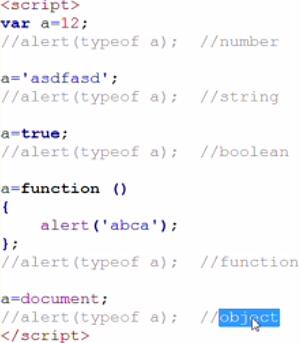
DOM 有一些操作不兼容

BOM 没有兼容问题（完全不兼容）

类型：typeof运算符

用法：返回值

常见类型



变量本身没有类型

Undefined 1.没有定义

2.虽然定义，没有给东西

一个变量应该只存放一种类型数据

显示类型转换（强制类型转换）

字符串转数字parseInt()：输出为整数

parseFloat();可输出为小数，也可以是小数

从左到右碰到不是数字，则跳出。

NaN: Not a Number非数字

任何数和NaN相加都为NaN

NaN和NaN不相等

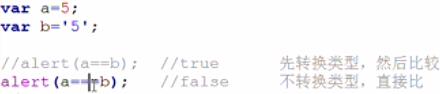
isNaN();判断是否是NaN





隐式类型转换

双等号，先转换类型，然后比较

三等号，不转换类型，直接比较

+ 1.字符串连接 2.数字相加

- 1.数字相减（隐式类型转换）

变量的作用域和闭包

变量的作用域

局部变量，全局变量

闭包：子函数可以使用父函数的局部变量

命名规范

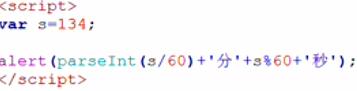
可读性

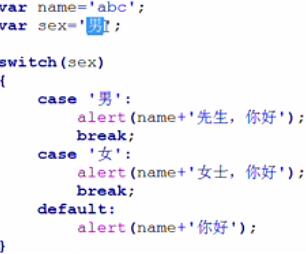
（给变量命名需要前缀，给函数命名不需要前缀）匈牙利命名法：类型前缀，首字母大写

% 求摩

隔行变色（求摩应用）





Switch应用

Break整个循环中断

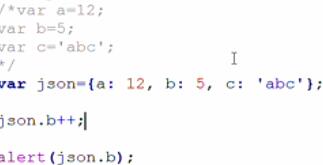
Continue 本次循环中断，继续下次循环

真:true 非零数字 非空字符串 非空对象

假:false 数字零 空字符串 空对象 undefined

Json

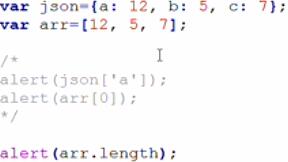
存储数据

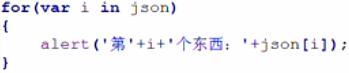


Json没有length

Arr有length属性，数组循环一般用length循环

Json循环





深入JavaScript

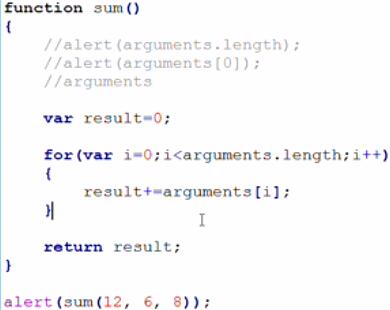
函数返回值

函数的执行结果，也可以没有return

Return 只能有一个

函数传参

Arguments 可变参 不定参



Css取样式设置样式



argunment html css

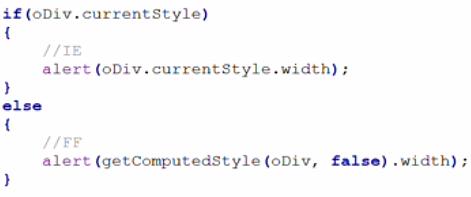
获取非行间样式 currentStyle 只兼容IE或者高的谷歌

获取非行间样式

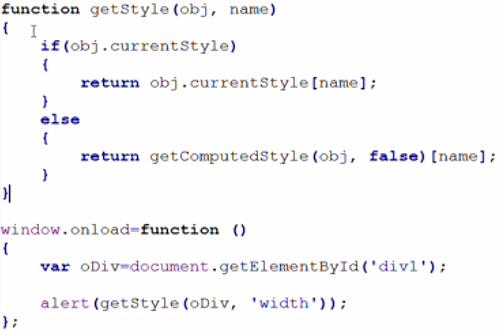
getComputedStyle可兼容获取元素，第二个参数随意值（false）（IE9之前不可用）

可兼容获取元素

都可兼容获取元素



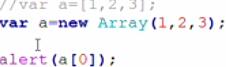
自己封装一个函数，调用函数使用



复合样式 background border

单一样式 width,backgroundColor

数组写法 建议使用第一种



数组中应该只存一种类型。

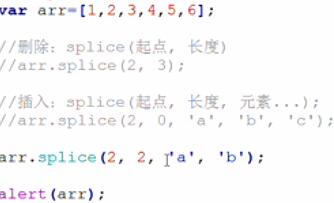
Push() 尾部添加

Pop() 尾部减去

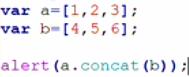
Shift()，从头部弹出

Unshift() 从头部添加





Concat（数组2）数组链接

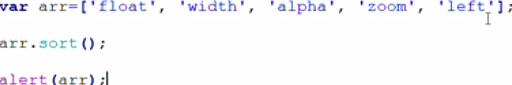


Join(分隔符) 用分隔符，组合数组元素，生成字符串，字符串split

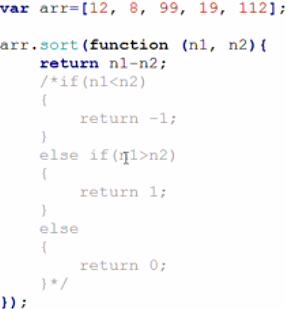
join

数组排序

字母排序



数字排序



定时器的使用

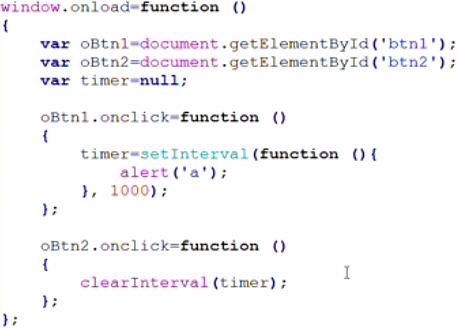
setInterval 间隔性

SetTimeout 延时性

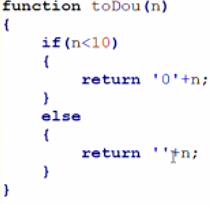
clearInterval 关闭定时器

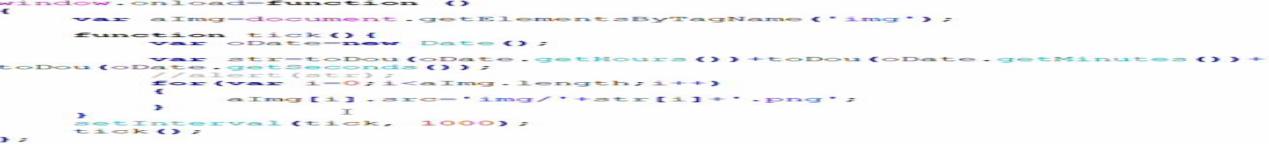
clearTimeout

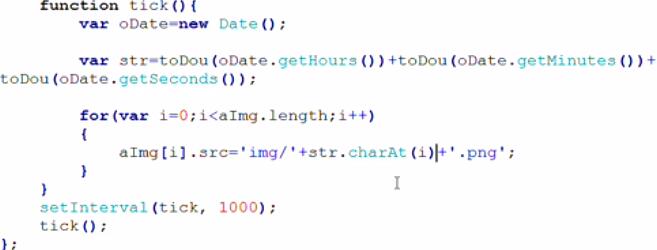
开始关闭定时器



开始关闭htmljpg

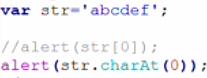




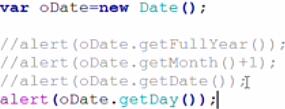




获取字符串某一位的东西



Dare对象用法

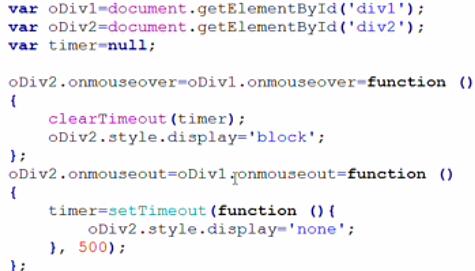


延时弹出框

延时弹出框css

..css

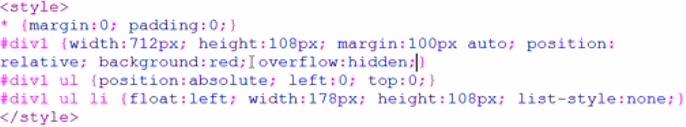


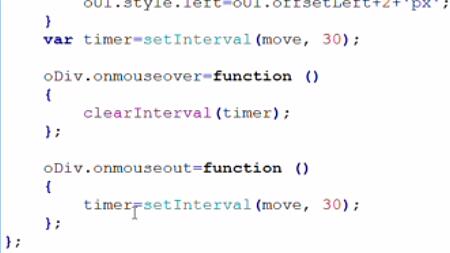
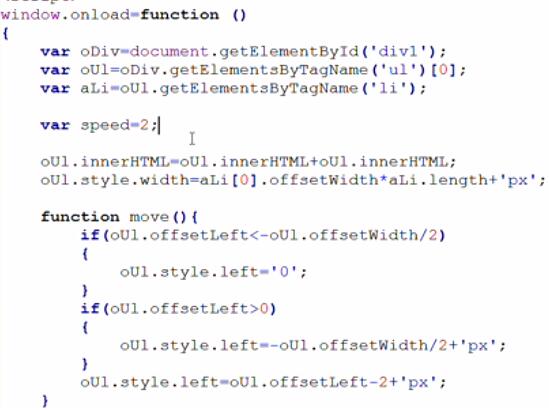
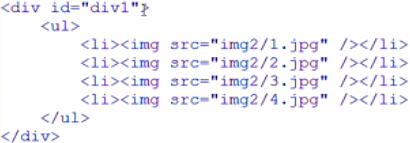


offsetLeft获取左边距，综合考虑能影响元素所有左边距属性。



无缝滚动



无缝滚动html1

DOM基础

DOM document 操作浏览器上的元素

childNodes（子节点）算空的文本节点 在IE6—8不会出问题。

Children 只取第一层

Asdasd文本节点

<span>.fasdf元素节点</span>

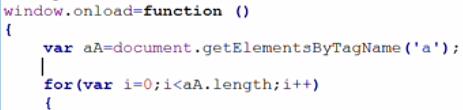
nodeType==3文本节点

nodeType==1元素节点





父节点parentNode 不会改变





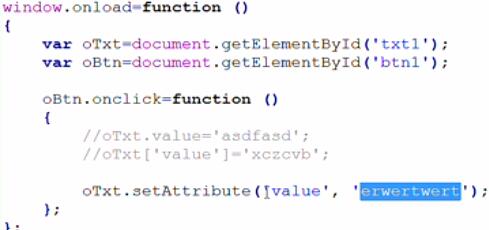
offsetParent 获取有定位的父级，根据样式会改变

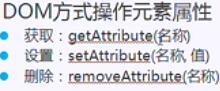


firstChild

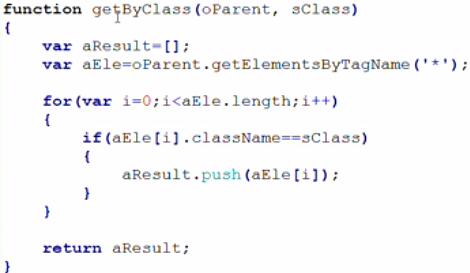


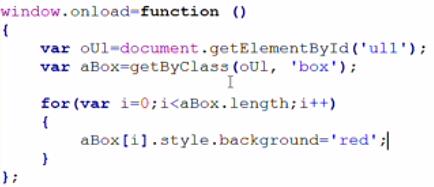
元素的属性操作(一般不用)

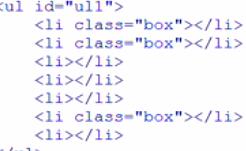
DOM元素属性操作html



用className选取元素







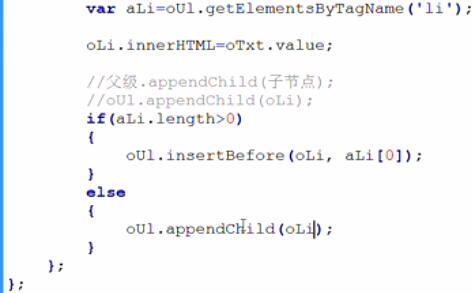
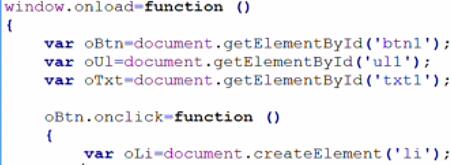
DOM操作

创建元素createElement

父级.appendChild（子节点） 追加一个节点

插入DOM元素

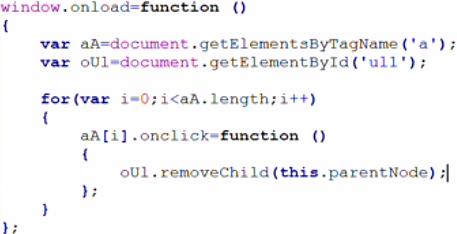
父级.insertBefore（子节点，哪个位置前）

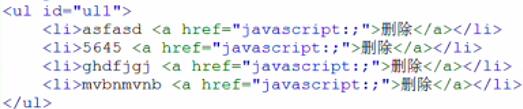




删除DOM元素

父级.removeChild(子节点)

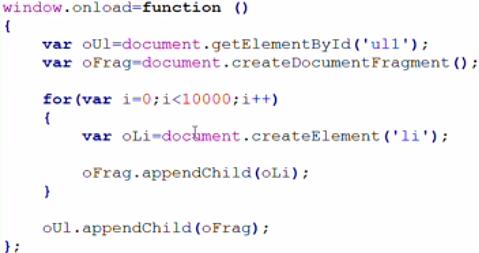




文档碎片

文档碎片理论上可以提高效率（IE6-7才会有些效果，高级浏览器不会有很多影响，基本没有用了，甚至可能会降低性能）

creatDocumentFragment 创建文档碎片



<ul id=”ul1”></ul>

表格

Tbody浏览器会自动加上，最好主动写上。

tBodies获取tbody

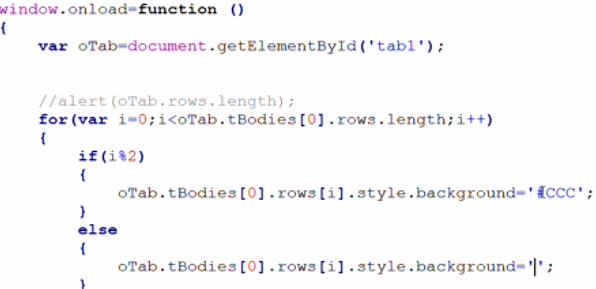
Rows行

Cells取行中单元格

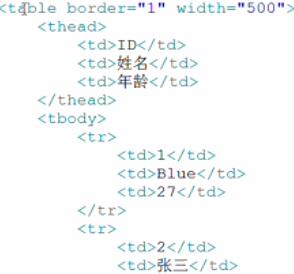
只能有一个标头，可以有很多tbody

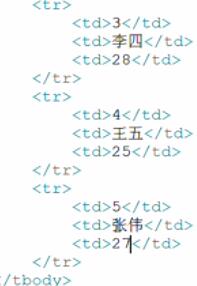


表格隔行变色

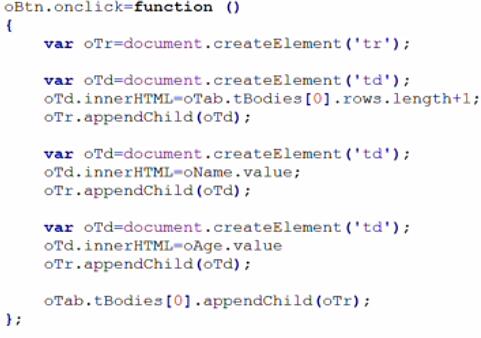
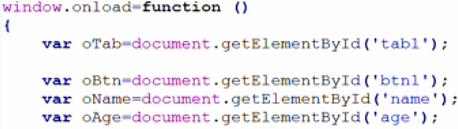
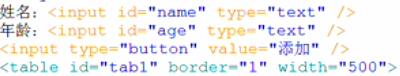






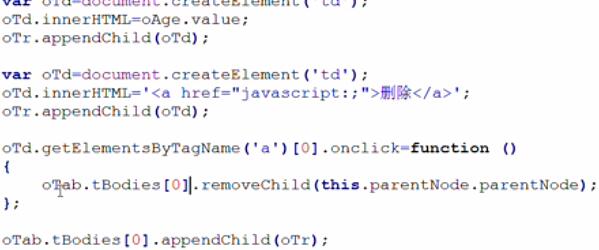


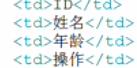
表格添加



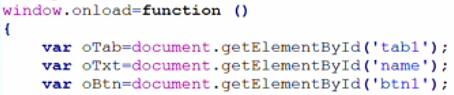
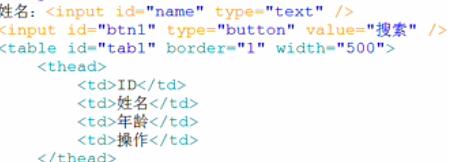
表格删除







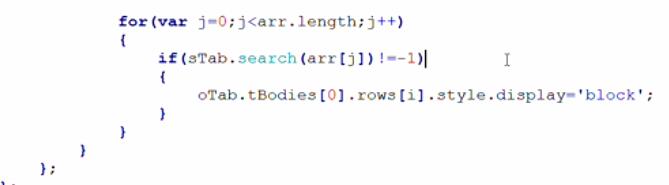
表格搜索





Search 找到并且返回下标，找不到返回-1.

表格筛选

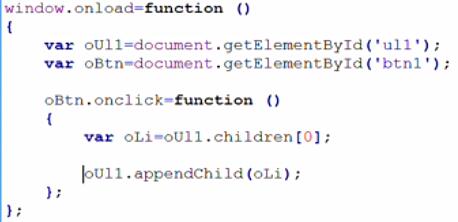




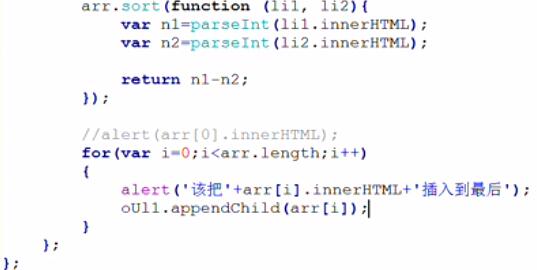
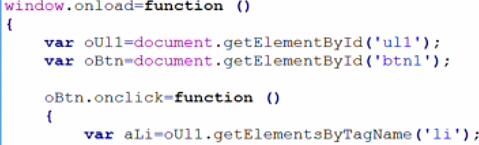
Appendchild

1. 先把元素从原有父级上删除，
2. 添加到新的父级

移动

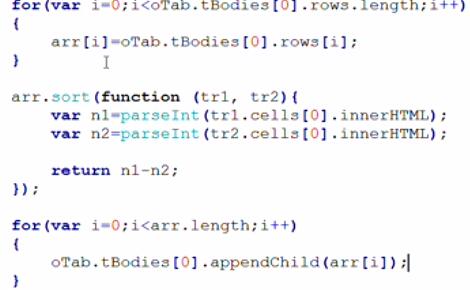
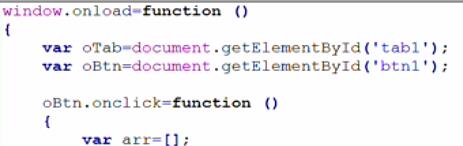


排序



表格排序

同表格隔行换色html一样



运动

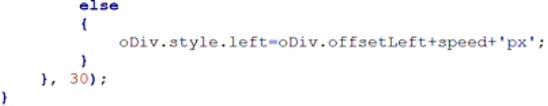
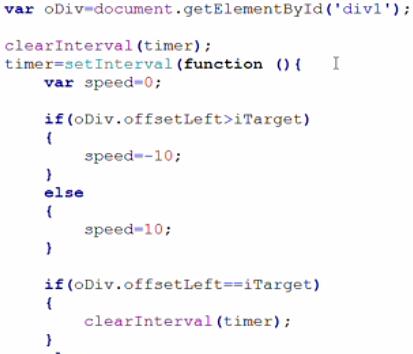
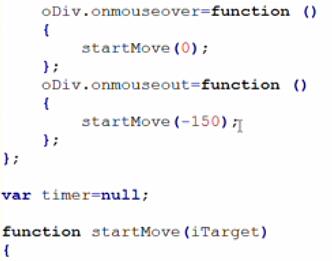
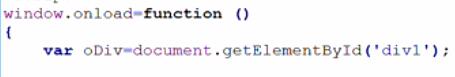


运动框架：

在开始运动时，关闭已有定时器。clearInterval(timer)

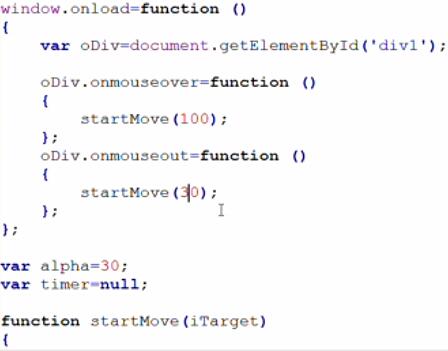
把运动和停止隔开(if/else).

分享到例子：

分享到html

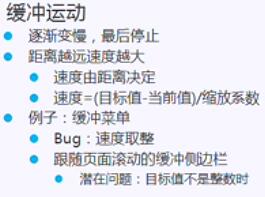
淡入淡出例子

淡入淡出css淡入淡出htnl



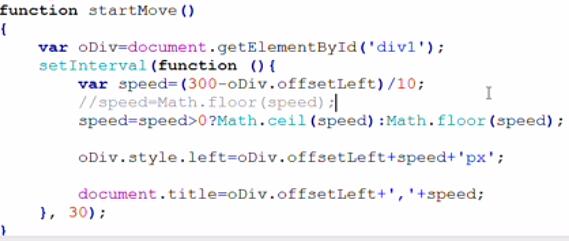


缓冲运动



缓冲运动例子：

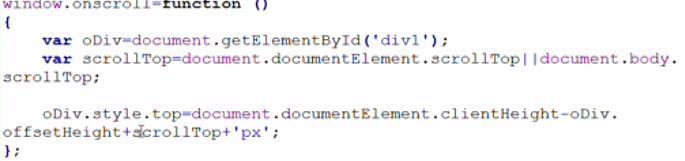
缓冲运动css缓冲运动html



右侧悬浮框（缓冲菜单）

右侧悬浮框css

右侧悬浮框html

右侧悬浮框Js1

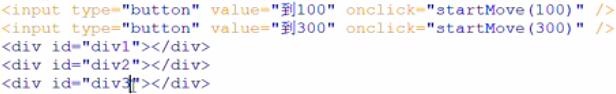
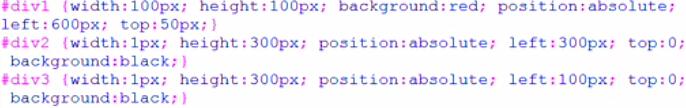
右侧炫富框js4

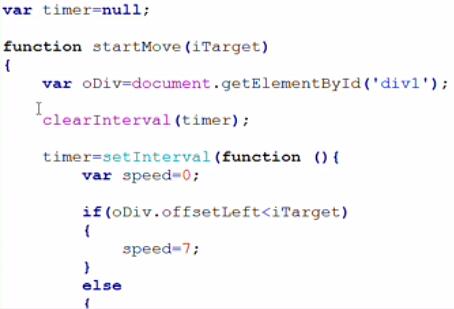
对联悬浮框

除了下面代码，与以上代码相同

对联悬浮框js

匀速运动的停止条件

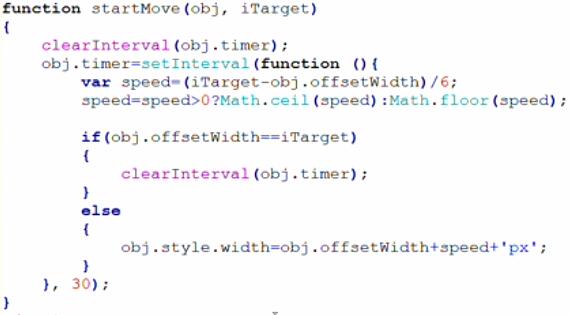
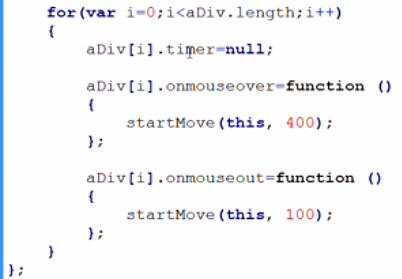
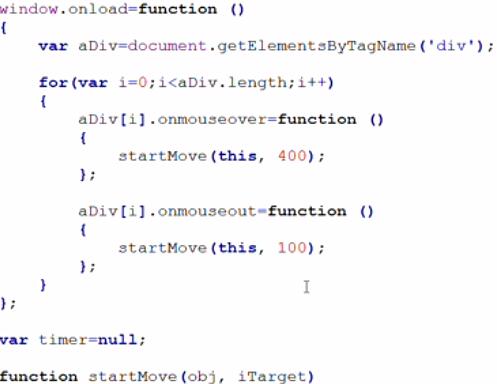






多个div变宽

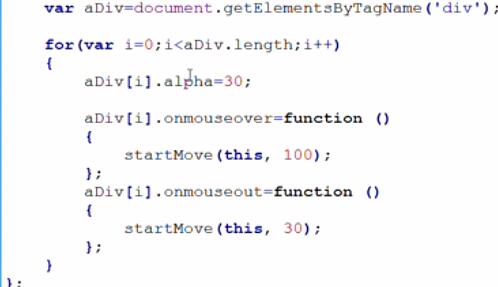
拉长css拉长html



多个淡入淡出

多个淡入淡出css

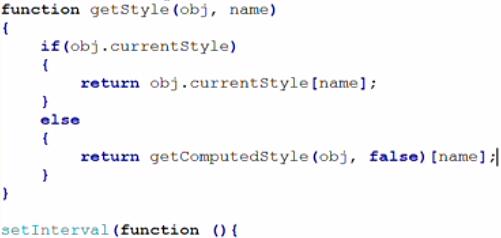


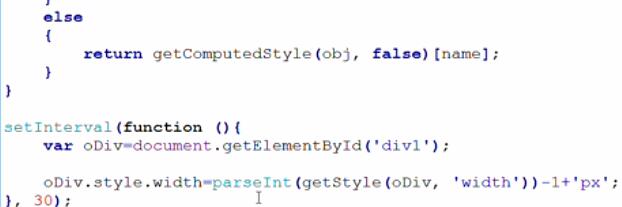


Offsetwidth bug



解决方法1（行间）



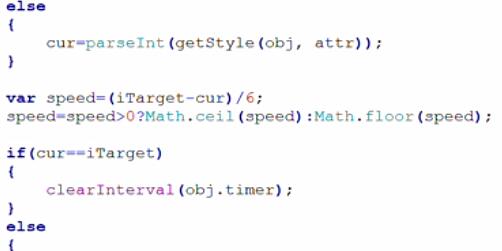


Math.round四舍五入

运动框架

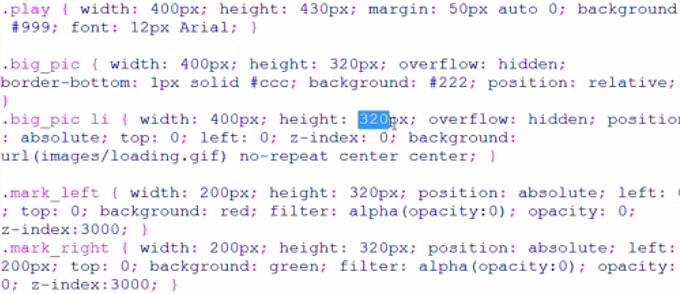
<div id=”div1”></div>

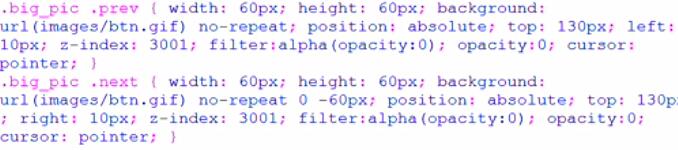
运动框架css

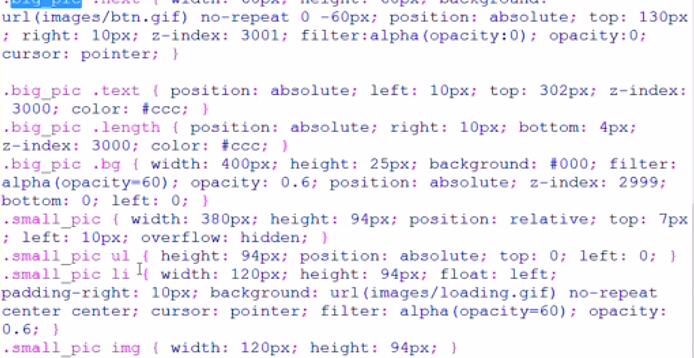


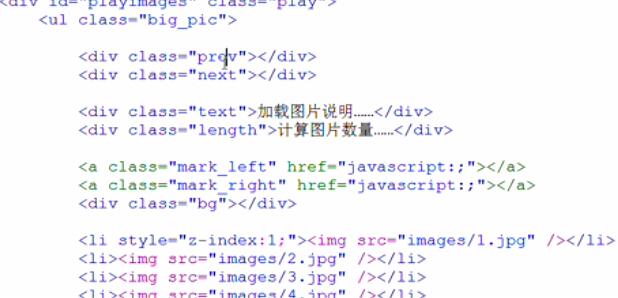
仿Flash动画 所有的东西一定不要公用 18

Flash css1

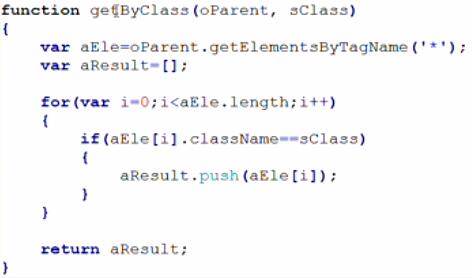


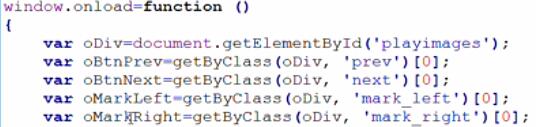












Flashjs1(1)

Flash js1(3)

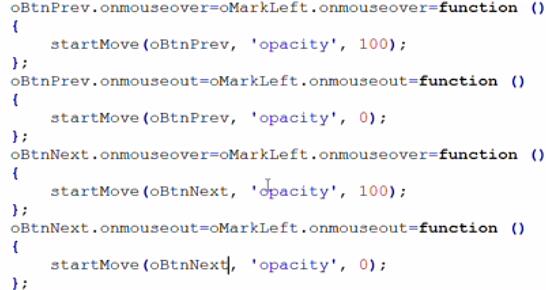
Flash js1(6)

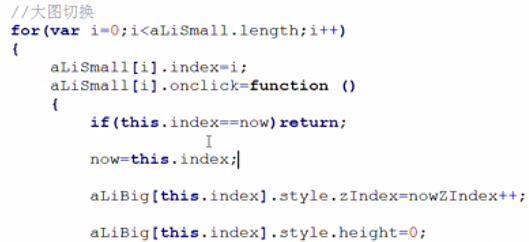
Flash js1(2)

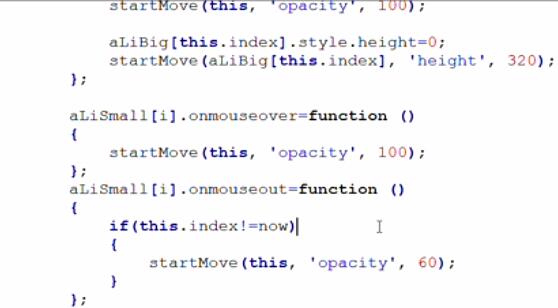
Flashjs1(4)

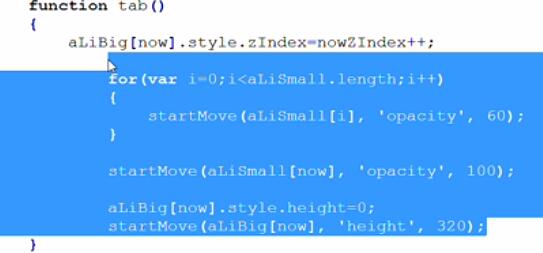
Flash js1(5)

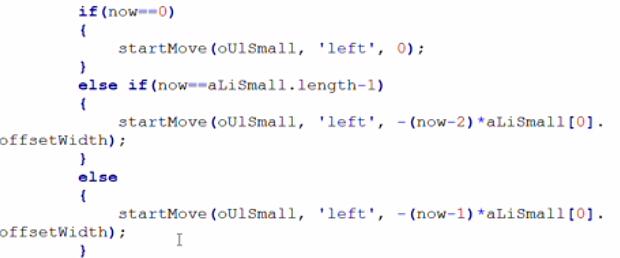
Flash js1(4)



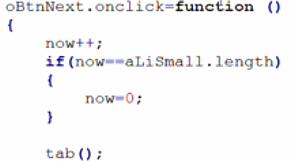
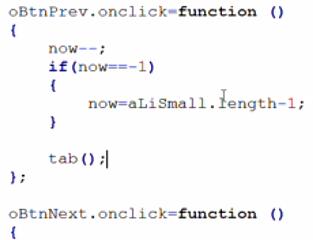




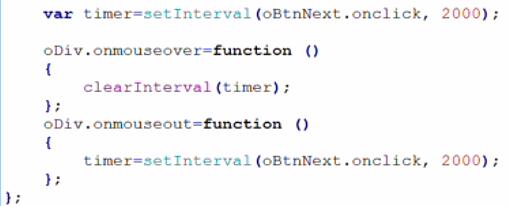




}



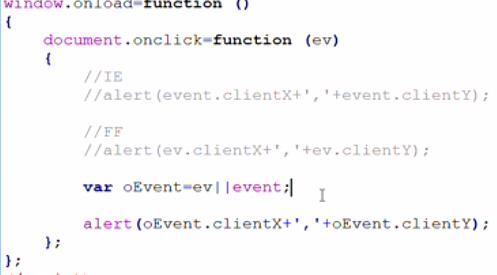
}



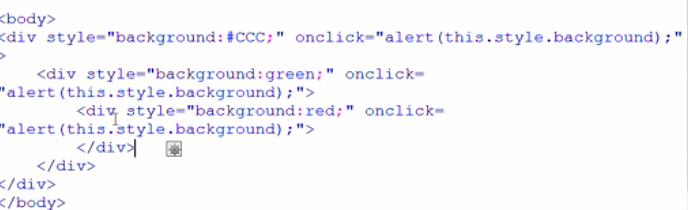




点击（兼容性写法）



事件冒泡



仿select

仿select html仿selectcss



oEvent=ev||event;获取事件

oEvent.cancelBubble取消冒泡



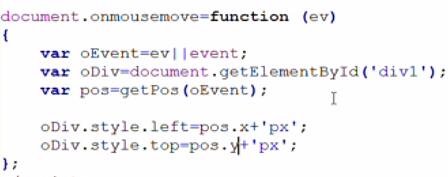
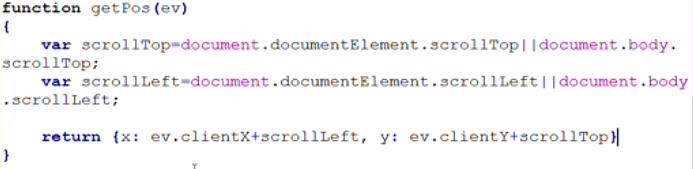
鼠标坐标

用clientX,Y一定要用scrollTop

clientY 鼠标可视区位置

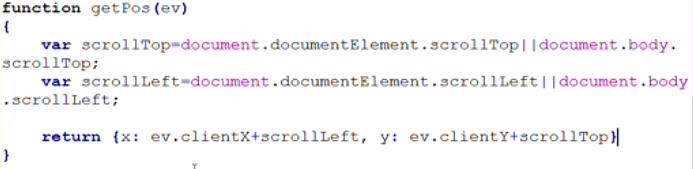
<div id=”div1”></div>

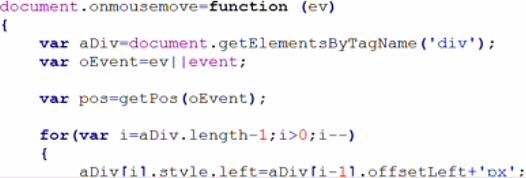
鼠标坐标

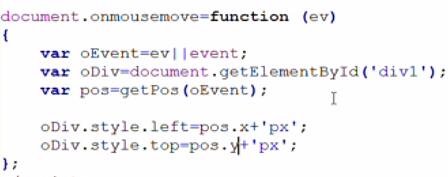


一串div跟着鼠标走

第一个div跟着鼠标走,从后往前循环，后一个div跟着前一个div走





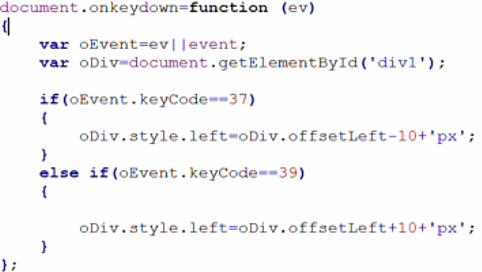




键盘控制移动

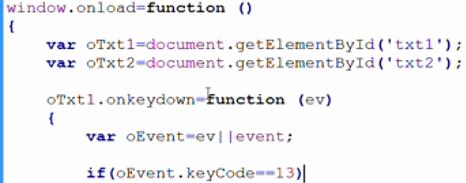
<div id=”div1”></div>

键盘控制移动css



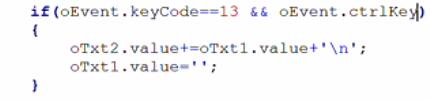
回车提交

回车提交html



回车提交js2

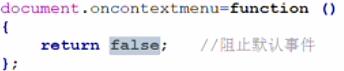
Ctrl+回车提交



默认行为

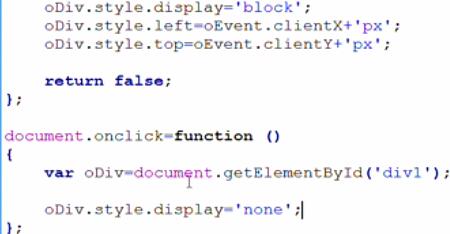
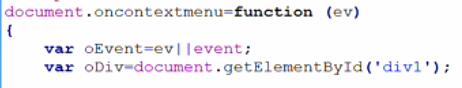
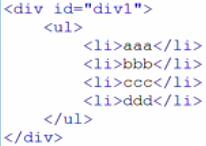
浏览器自带的行为。

阻止默认事件



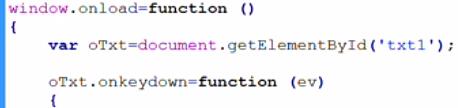
Oncontextmenu 点右键菜单的时候

自定义右键菜单

右键菜单css

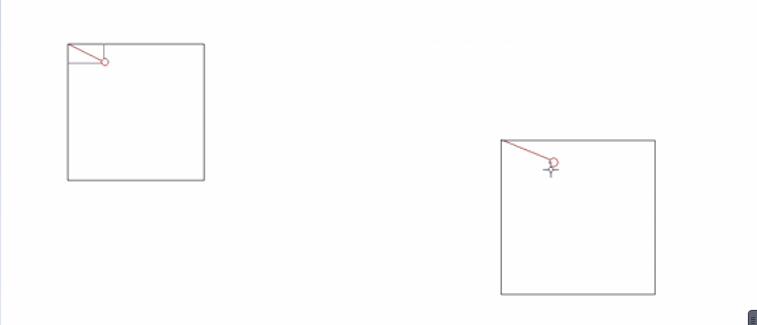
input只允许输入数字的文本框

<input type=”text” id=”txt1”/>

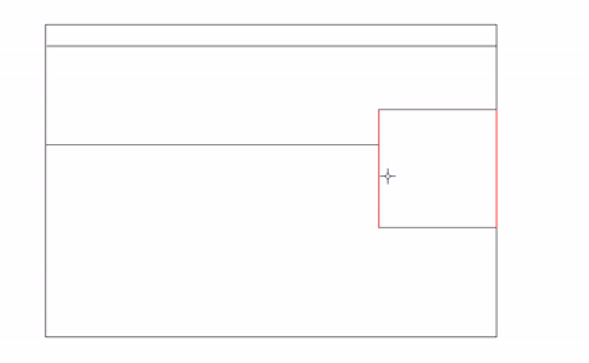




拖拽

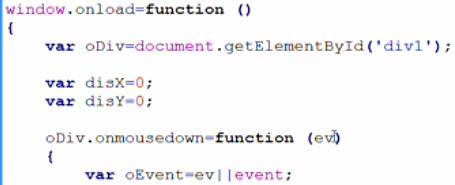


右边界限



<div id=”div1”></div>

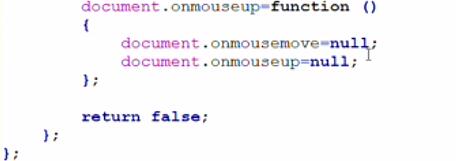
拖拽css



拖拽js5

防止用户处可视区





Return false; 除掉火狐bug

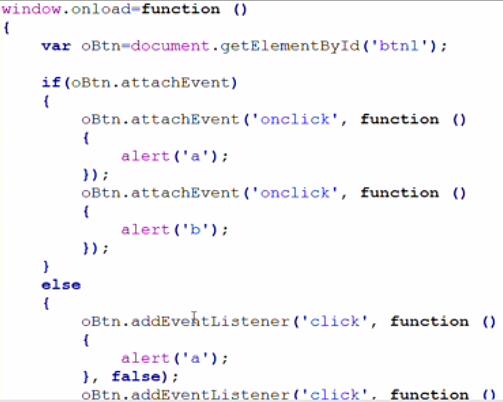


绑定事件

attachEvent(事件名，函数) （IE下）onclick

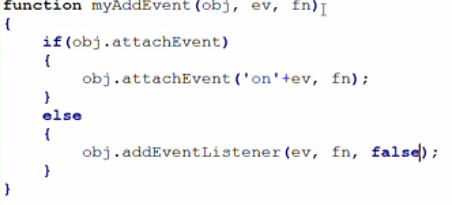
addEventListener(事件名，函数，flase) (火狐，谷歌，IE9)click

为什么给document加事件，因为div太小鼠标就跑了，div被甩掉。

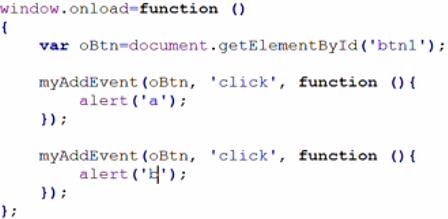




封装绑定事件函数



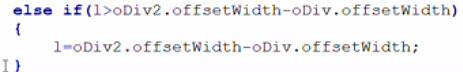
调用封装绑定事件函数

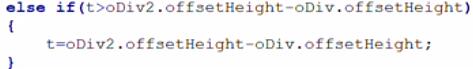




高级拖拽

不能拖出指定对象的Div

高级拖拽改css高级拖拽改js1



磁性吸附



事件捕获

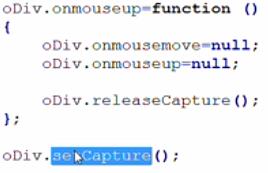
<input oid=”btn1” type=”buttn” value=”按钮”/>

setCapture 不会选中文字，事件捕获只有IE兼容，整个网页的所有事件都会聚集在一个div上

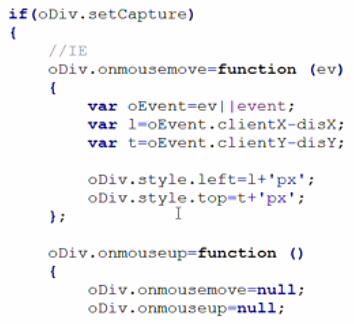
releaseCapture();取消事件捕获

oDiv.setCapture();集中在oDiv上。

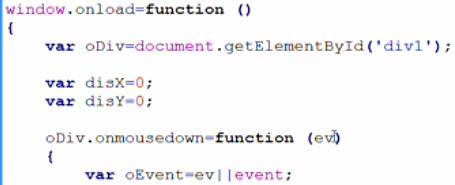
Document改为oDiv（IE下）



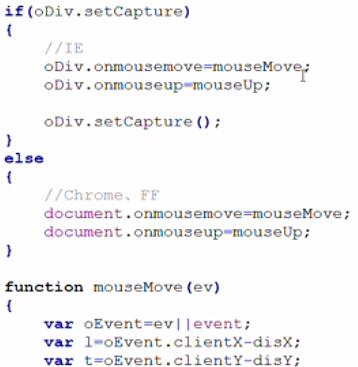
用if else处理兼容 把document换成oDiv,同上整理

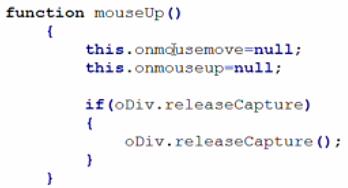
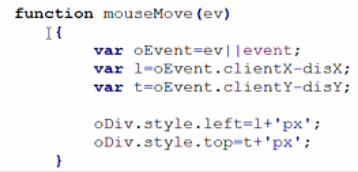


合并事件捕获(完美拖拽)



拖拽js5





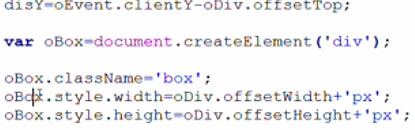
合并事件捕获js4

带框的拖拽

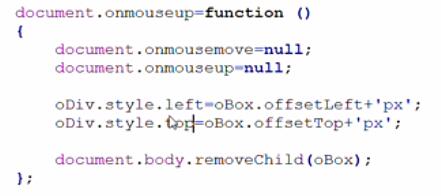
与DOM配合的带框拖拽

添加.box css

带框拖拽css



带框拖拽ls4



自定义滚动条

<div id=”parent”>

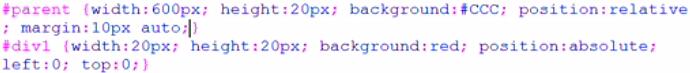
<div id=”div1”></div>

</div>

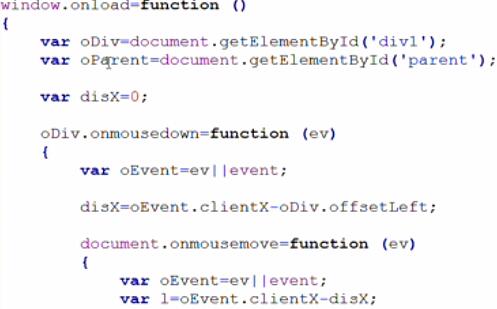
<div id=”div2”>

<div id=”div3”></div>

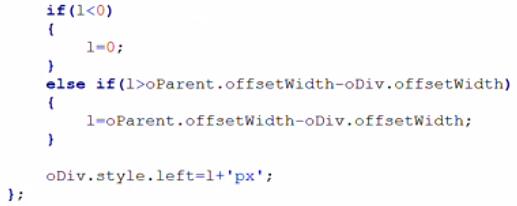
</div>



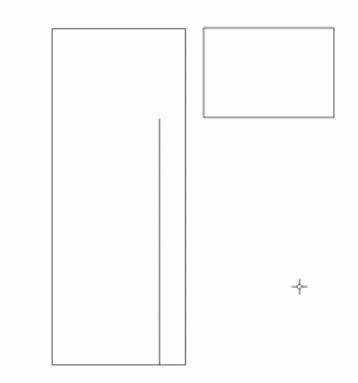
自定义滚动条css2

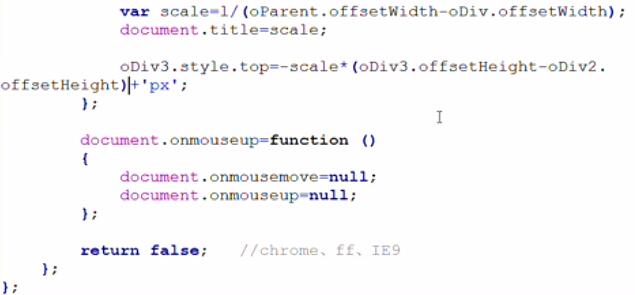


上下不能拖拽，左右不能出父级。

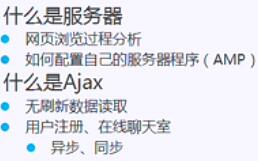


滚动范围是div3-div2的高度。





AJAX基础



让js可以读取或者发送服务器上的数据（不用刷新页面）

什么是服务器

PC，手机也能作为服务器

Locallhost 本地主机

自己配一个服务器，把文件存在www下

Function alax(url,fnSucc,fnFild){}

第一个参数，从哪去取

第二个参数，读成功，回来的时候，会主动访问自己写的一个函数

不成功返回fnFile

放在服务器下的文件，不能用中文命名

统一存储编码都是UTF-8或者其他。

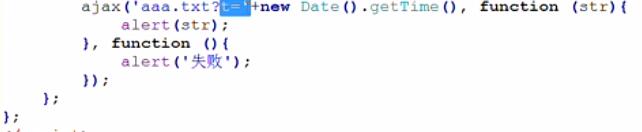
缓存

根据url来缓存的

域名后的?后 get数据



清除缓存处理 加一个没用的t



Eval()；解析成js

读取数组,json数据



读取数据并生成ul



Ajax原理(id 给前台用，name给后台用)

Get请求

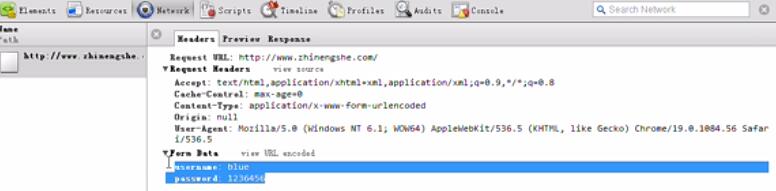
Get方式：放入url

名字=值&名字=值...

gei数据

Post请求

post请求



（网址长度有限）

Get:通过网址

容量小

安全性非常差

有缓存

用于获取数据

Post:不通过网址，http

容量大（2G）

安全性好一点

没有缓存

向服务器传递信息

HTTP请求方法

GET——用于获取数据（如：浏览帖子）

POST——用于上传数据（如：用户注册）

GET、POST的区别

get是在url里传数据：安全性、容量

Ajax原理

1. 创建Ajax对象
2. 连接到服务器
3. 发送请求
4. 接收返回值

全局变量是window属性

Js中没有定义的的变量--报错

没有定义的属性--不报错

Open(方法,文件名,异步传输)

1.Get？Post? 2.读哪个文件 3.

同步 多件事一起 js中事件一件一件来

异步 一件一件来 js中多个事情一起来（Ajax）

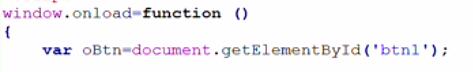
如果同步的话，网速很慢，其他标签什么的都不会显

onreadystatechange和服务器交互的时候发生的事件

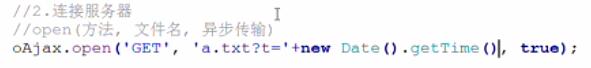
创建Ajax

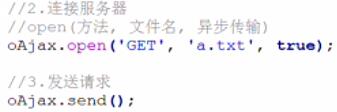
ActiveXobject();针对IE6兼容

XMLHttpRequest();针对非IE6

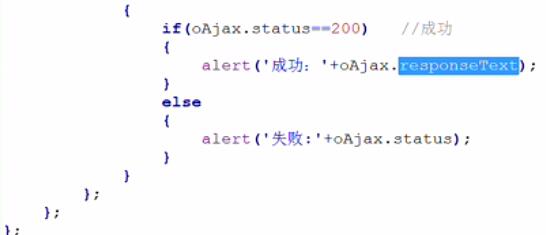


制止缓存





Statue 200成功，除200以外都是失败



封装Ajax





面向对象（）

不了解原理的情况下，使用它内部的功能。（使用对象时，只关注对象提供的功能，不关注其内部细节。比如：jQuery）

对象：

黑盒子，不了解内部结构，知道表面的各种操作（按钮）duixian

对象是一个整体，对外提供一些操作。

面向对象时一种通用思想，并非只要编程中能用，任何事情都可以用

面向对象

1. 用对象
2. 写对象

JS中的面向对象

面向对象编程（OOP）的特点

抽象：抽象最主要的特性，跟问题相关的特性抽出来

封装：看不到里面的东西，用好表面的功能就行了。

继承：代码重用，从父类上继承出一些方法，属性，又有自己的属性（遗传）

多重继承 多态

对象的组成

方法——函数：过程，动态的

属性——变量：状态，静态的

变量能做的事情，属性也能做。

变量：自由的，不属于任何人

属性：属于一个对象的

函数：自由

方法：属于对象

This:当前发生事件的对象（之前）

当前的方法属于谁，this就是谁

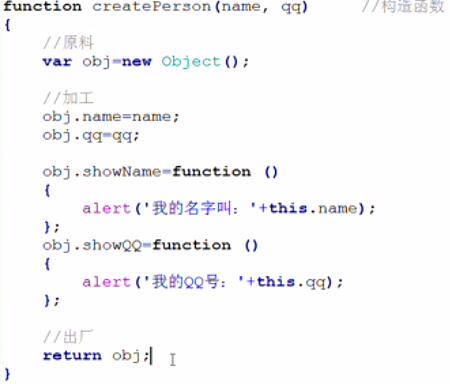
不能再系统对象中随意附加方法，属性，否则会覆盖已有方法，属性。

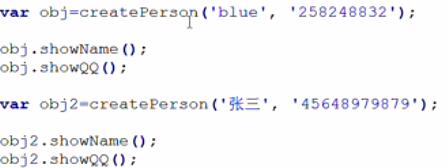
Date 日期 Array 存东西 RegExg 正则 Object 没有功能

第一个面向对象程序

构造函数（就是普通函数）工厂方式

Var obj = new Object();创造空白对象





工厂方式

没有new (解决方法如下)



面向对象this js

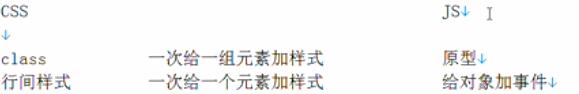
obj.showName();

obj.shoeQQ();

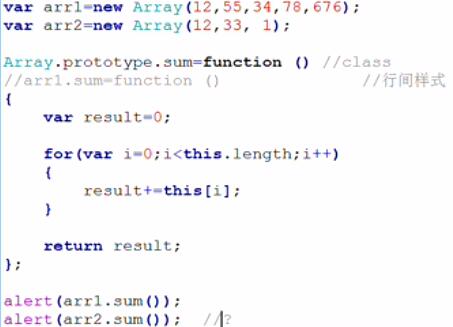
函数重复（解决方法——原型）

原型：





原型应用（一次给一组加东西）



类：模子

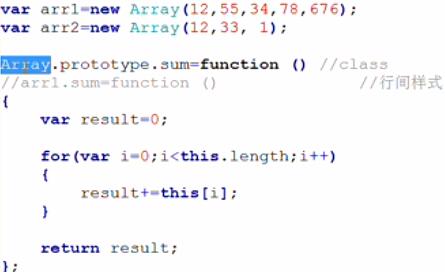
对象：产品（成品）

Var arr=new Array(1,2,3); arr对象 array类

混合方式

用构造函数加属性

用原型加方法



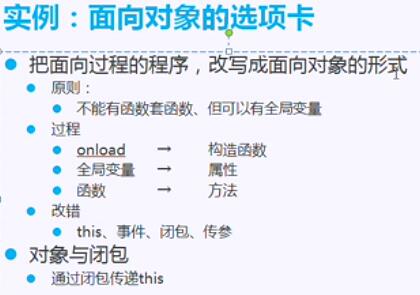


面向对象写选项卡（面试可能会考）









Window.onload 初始化整个程序

全局变量 添加this

原型添加方法

fnClick不是函数了，是方法，fnClick前面加this

aBtn前也需要加this

Json面向对象的方法

Json：整个程序里只有一个，写起来比较简单

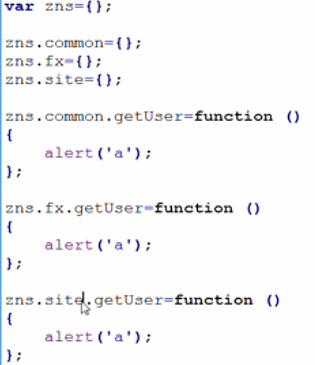


Json命名空间（单体）

把方法包在一个Json里

有人管它叫——命名空间

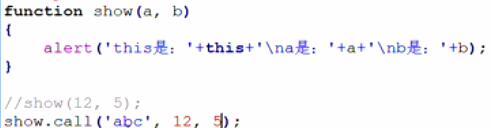
在公司里，把同一类方法，包在一起



对象：属性+方法

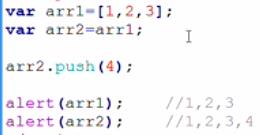
继承：父级的属性+方法

Call用法

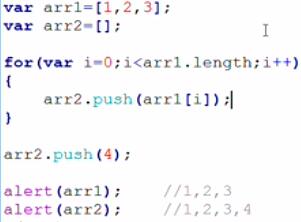


引用

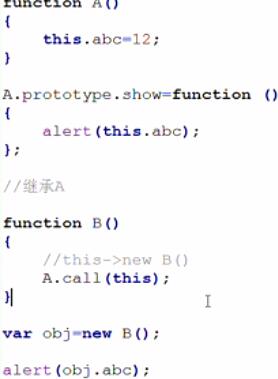
Arr1,arr2指向同一空间

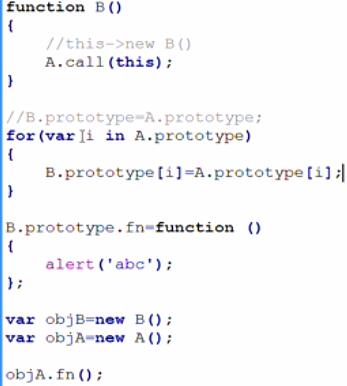


解决方法（arr1,arr2指向的不是同一空间）



Call继承父级方法



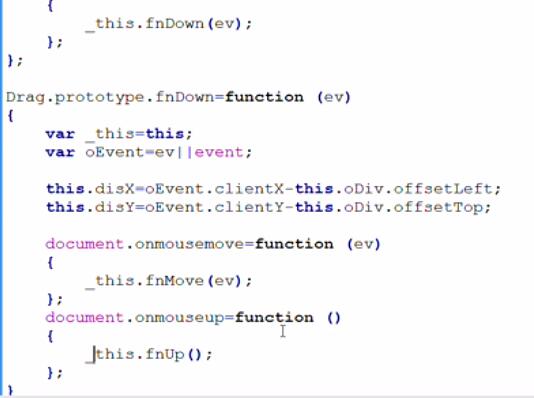


使用继承

面向对象拖拽（把这个程序封装成一个js库）



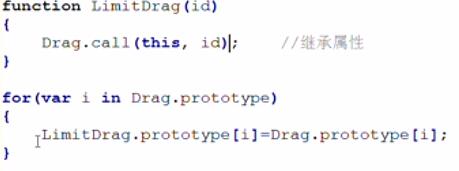
面向对象拖拽js改





继承面向对象拖拽（css没写）





给limitDrag加功能



系统对象



宿主：运行环境（浏览器）

本地对象+内置对象（JS语言本身具备的，跟环境无关）

静态对象：不能new没直接用的 代表 Math

非静态对象：必须new,实例化才能用的对象

BOM

Document.write();先清空，再写

Open用法



Close （必须你开的，你才能关）

BOM close

window.navigator.userAgent判断当前浏览器类型，操作系统类型

Window.location 当前网页地址

可以读取可以赋值

BOM location 

BOM尺寸及坐标



可视区大小



srocllTop可视区距离顶部的距离

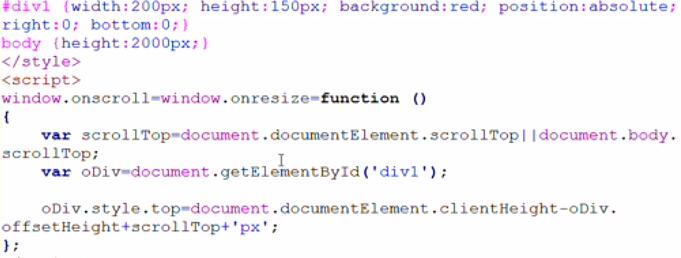
滚动（给body加一个高度）



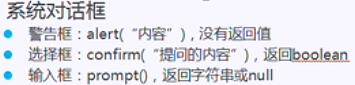
Window.onresize当窗口改变大小是发生

userAgent>IE6 fixed

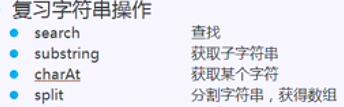
IE6 运动



系统对话框



正则表达式



Search 返回位置，没有找到返回-1

Substring 不包括结束位置，可以不写第一个参数，行头开始湖区字符串

chatAt 返回字符串的某一位

Split 把字符串切成数组

正则表达式：是个规则，模式

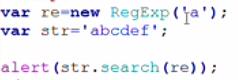
强大的字符串匹配工具

是种正常人很难读懂的文字

JS风格正则表达式

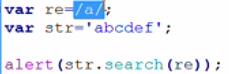
Var re = new RegExp(‘a’,’i’);

第二个参数i，忽略大小写,ignore



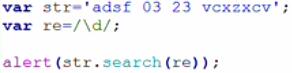
Per风格

Var re = /a/i;



正则转义

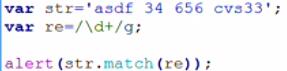
\d 数字 （找数字） \d一位数字 \d\d两位数字



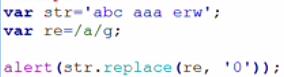
Match 把所有匹配的东西都找出来

g global 找到全部，全局匹配

量词：个数 若干（多少都可以）

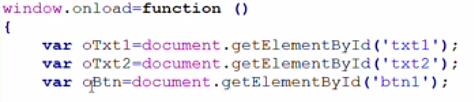


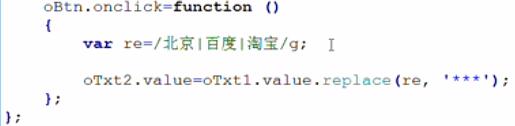
正则replace



敏感词过滤 | 或者

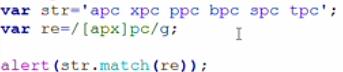
敏感词过滤html





正则元字符 （方括号）

[abc]pc 或者 apc bpc cpc



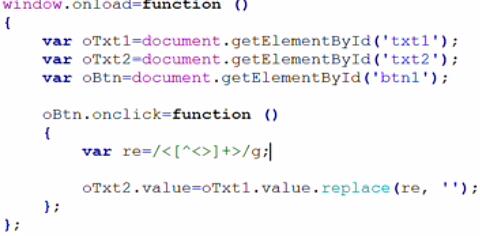
\d 相当于[0-9]

[^a-z0-9]匹配除了英文和数字以外的

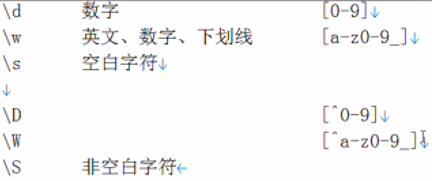
过滤html标签

Var re = /<[^<>]+>/g; 除了两个<>以外的所有字符

敏感词过滤html

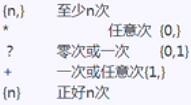


转义字符



量词 ： 个数

+ ：多个



手机号：[1-9]\d{7}

Qq号：[1-9]\d{4-10}

固定号码:(0\d{2,3}-)?[1-9]\d{7}

正则邮箱

正则邮箱



Text 字符串一部分符合要求，就返回true

^行首 $行尾

/d == [0-9]

<.+> 任意个任意字符