# 浏览器发展

html(超文本标记语言)

css(层叠样式表)

给文档添加各种样式

dom(文档对象模型)

是一套对文档的内容进行抽象和概念化的方法.

是一种api;

利用dom给文档添加交互能力

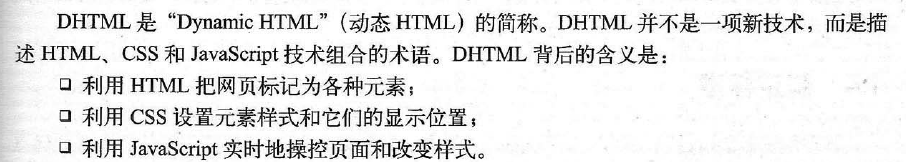
bom(浏览器对象模型)

设定浏览器属性的办法

----------------------------------

1997年 Netscape ie 争夺 扩展了js 的dom

dhtml(动态html)



现如今扩展为dom脚本程序设计

--------------------------------

浏览器

webkit引擎

--Safari苹果浏览器

--chrome 谷歌浏览器

gecko

--firefox 火狐浏览器

trident

--ie ie浏览器

# js语法

语法:语言结构上各方面的规则,称之为语法;

语句(statement):脚本的一系列指令

需要根据语法编写语句;

--注释语句(comment)

变量:会发生变化的东西

--命名规则

--赋值:把值存入变量的操作

声明(declare):

数据类型

说明:所有的变量实际上是某种类型的对象(实例);

--类型声明(typing)

----强类型(strong typed)

----弱类型(weakly typed)

--标量(sclar):任何时候只能有一个值;标准变量

--数组:

----关联数组:用字符串代替数字下标

对象(Object):是指一种类型

--内建对象

js提供的对象

--用户自定义对象

用户自己定义的对象;

--宿主对象

浏览器提供的预定义对象比如document form image elment

操作

# dom

d:document

dom把文档装换为一个文档对象

o:object

浏览器预定义对象,bom(浏览器对象模型 window)

m:mode 或是map

dom代表着加载到当前浏览器的网页,浏览器提供了加载网页的地图,js可以读取这张地图;

地图用树结构的方式,层层表示关系;

节点:文档的组成成分

每一个节点在获取时,都要表示为一个对象,然后获取他们的值(用nodeValue属性)

节点类型:

--元素节点:dom的原子;

标签:元素的名字;

--文本节点:文本,总是包含在元素节点内部

--属性节点:用来对元素做出更具体的描述

css说明:

把文档视为一颗文档树,子类也会继承父类的css;

id 和class属性:

dom 规范中的获取对象的方法定义:getElementById() 等

typeof:

可以区分 字符串 数值 函数 布尔值 和对象;

获取和设置属性:getAttribute()

dom

DOM 是 W3C（万维网联盟）的标准。

DOM 定义了访问 HTML 和 XML 文档的标准:

w3cDOM是中立于平台和语言的接口,它允许程序和脚本动态的访问和更新文档的内容,结构和样式.

W3C DOM 标准被分为 3 个不同的部分：

* 核心 DOM - 针对任何结构化文档的标准模型

实现了DOM的脚本或语言可以在实现了DOM的解释器里使用DOM的规范访问文档;比如js和php

* XML DOM - 针对 XML 文档的标准模型

是关于如何获取 修改 添加 删除 xml元素的标准;

* HTML DOM - 针对 HTML 文档的标准模型

是关于如何获取 修改 添加 删除 html元素的标准;

HTMLElement接口的参与形式与上述的 Node节点接口非常地相似。一旦引入了OO设计观，只要实现了HTMLElement接口的对象，就可以定义它为元素了。一份HTML文档有相当多的都是元素，当然还有，表示文档本身的HTMLDocument和集合容器HTMLCollection等等。

dom core

element.getAttribute("title","")

html dom

element.title = "";

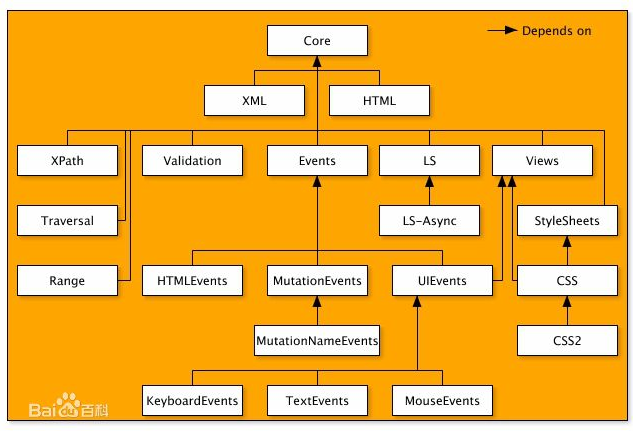
================================================================

* XML和HTML

XML与HTML相似，但两者是有区别。XML和HTML都来自于SGML，它们都含有标记，有着相似的语法，HTML和XML的最大区别在 于：HTML是一个定型的标记语言，用固定的标记来描述和显示数据

DHTML

* DOM 1 2 3



**常见的Core DOM方法如下**：

1、创建节点  
createElement()  
createTextNode()  
2、复制节点  
cloneNode()  
3、插入节点  
appendChild()  
insertBefore()  
4、删除节点  
removeChild()  
5、替换节点  
replaceChild()  
6、查找和设置节点属性  
setAttribute()  
getAttribute()  
7、查找节点  
getElementById()  
getElementsByTagName()  
hasChildNode()

**常见的Core DOM属性如下**：

node.childNodes  
node.firstChild  
node.lastChild  
node.nextSibling  
node.previousSibling  
node.parentNode



js实现了dom-core 和html-dom

用面向对象设计模式思考,

html 对象继承DOM Element;

html 对象的属性为DOM Attribute 对象 继承与此;

在 W3C DOM Core 中，Attr (attribute) 对象从 Node 对象继承所有属性和方法。在 DOM 4 中，Attr 对象不再从 Node 继承。

# 案例研究:js图片库

事件:

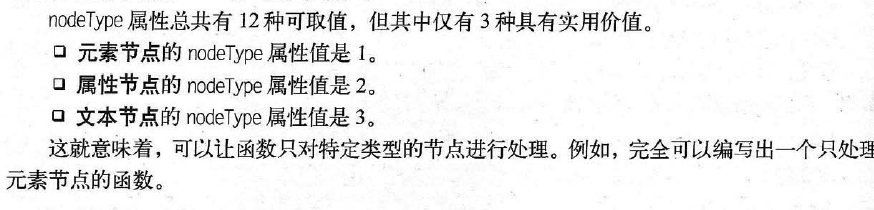
事件返回结果影响;比如 超链接的onclick 返回为false时不跳转连接

节点的获取

body.childnodes属性返回了所有的节点,这个节点包含了每一行,比如空格和换行符也是一个节点

节点类型的判断

nodetype



节点值的操作

nodeValue

文本作为节点,默认的位置永远是第一个 如 <**p** id=**"shuoming"**>  
 <**p**>123</**p**>  
 这里是图片说明  
</**p**>

这里包含三个节点,而不是两个;

firstChild lastChild 快捷操作;

# 最佳实践

弹出新窗口:

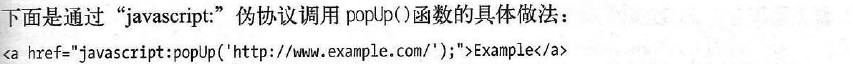
Windows

window.open(**"images/Chrysanthemum.jpg"**,**"新标签页面"**,**"width=320,height=640"**);

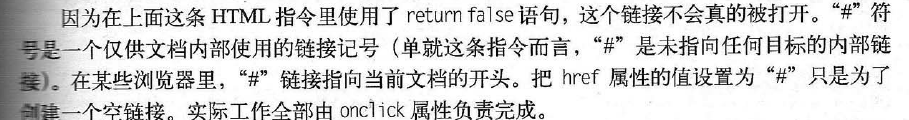
"javascript:" 伪协议



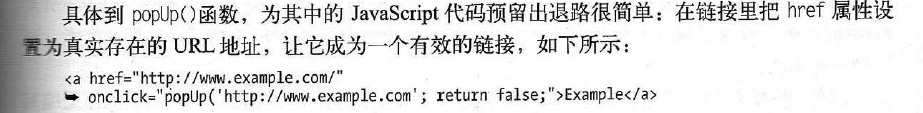
方式一



方式二



方式三 推荐用方式三,为了平稳退化



标记良好的内容就是一切

渐进增强原则

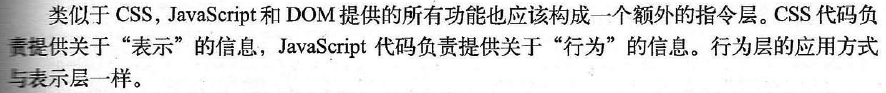
--css js 修饰原始html

--向后兼容

平稳退化原则

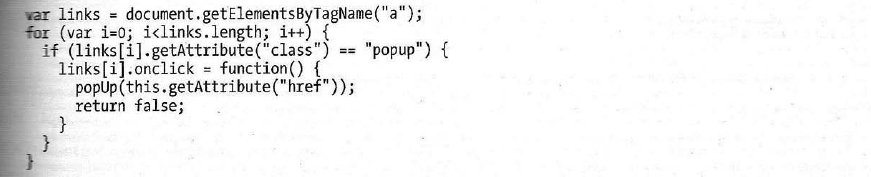
--屏蔽js时

--浏览器兼容某函数



分离javascript:

将超链接上的js分离到外部文件中;



性能考虑

--尽量减少dom的操作

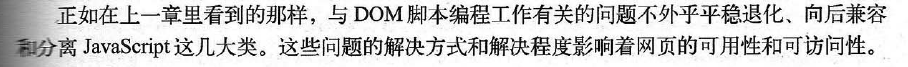
--尽量减少html中的标记

--合并和放置脚本,尽量减少脚本的数量,将脚本放置在文档后面可以更快显示

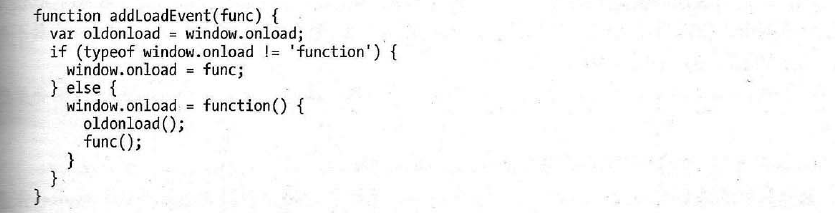
--压缩脚本:通过删除多余字符和修改变量名



# 案例研究:图片库改进版

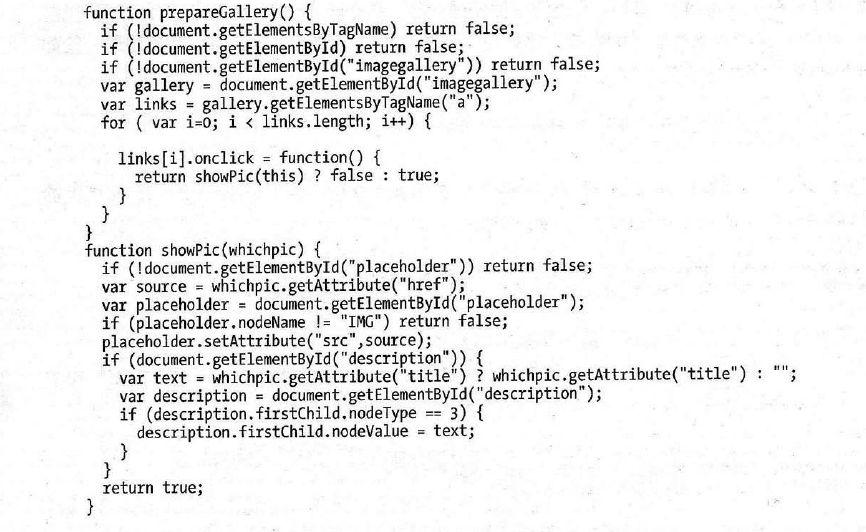


添加函数到onload



减少js对html 的id或class依赖

增加对文档的多种可能发生的情况进行判断和处理;



键盘访问

onpresskey

tab键将焦点切换到超链接上也会触发onclick

dom-core 和html-dom

dom-core 提供的方法任何语言都可以实现



# 动态创建标记

传统方法

document.write();

在html文档中,需要写入内容的地方执行上面语句,这就不能将js与文档分离,避免这样些

innerHTML属性

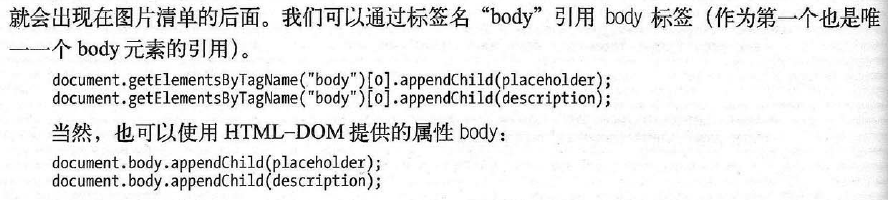
js精细操作dom,innerHTML不关心细节;

dom方法

document.createElement("p");

**document**.createTextNode(str);

body的引用



在已有元素前插入一个新元素;



编写insertafter 函数

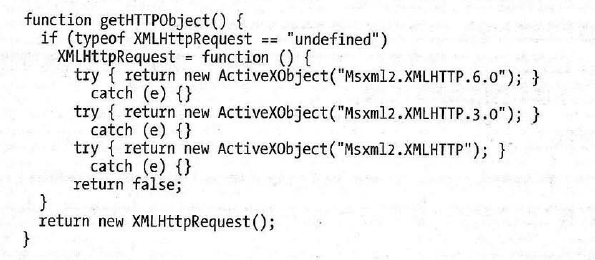


appendchild默认在最后节点后添加新元素;

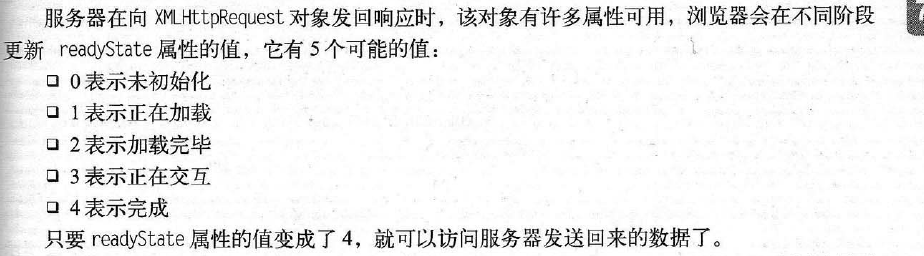
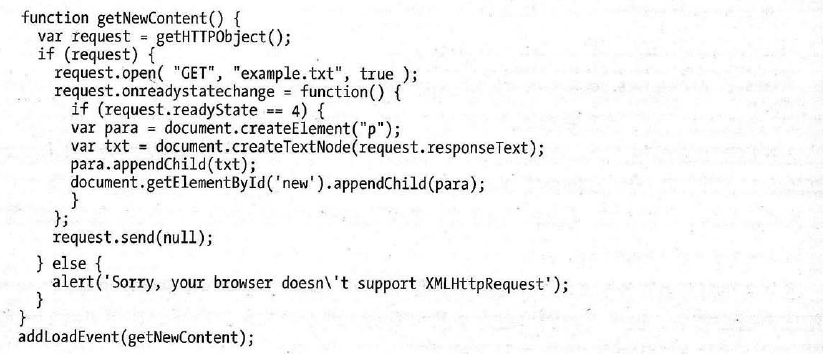
异步请求

ajax 局部刷新数据,搜索引擎不能抓取相关内容;

xmlhttprequest 对象



获取新内容



同源策略:ajax请求只能是在html同一域名下的http请求

平稳退化上的冲突:

在传统html项目上使用ajax;

以ajax为核心开发网站,



hijax:渐进增强的使用ajax;



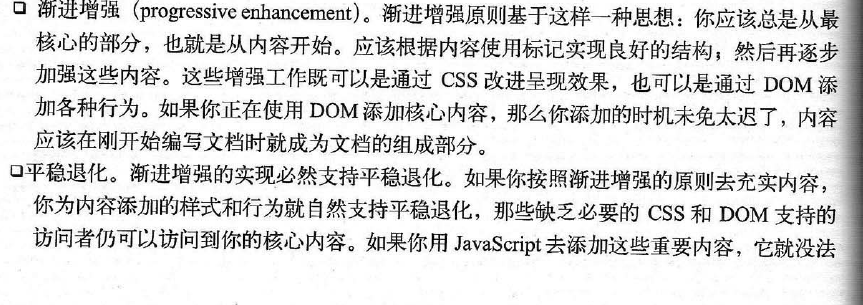
提交表单

提交表单触发submit事件, 通过onsubmit 事件处理函数捕获该事件,那么就会替换掉默认的操作(提交这个页面);

当浏览器不支持js时,表单可以继续提交,只不过会刷新整个页面的形式沟通;

# 充实文档的内容

搜索引擎几乎不支持js





渐进增强

平稳退化

网页

1.编写合适的标签(选择严格的doctype )html xhtml html5的选择

每个浏览器对标签的解读有时也是不一致的,

每个浏览器对标签都有自己的默认的样式

标签的属性有的也是包含了需要展示的信息的

2.使用css减少每个浏览器的样式差异,即取消默认值,显式指定样式;

3.使用js修改样式或是添加功能;(核心内容不用js创建)

添加文档的辅助性内容,比如创建快捷键的清单,创建缩略词的清单等;

4.兼容性修改,避免由于兼容性问题导致的js报错;

缩略语标签:

<abbr> 和<acronym>

浏览器之间的竞争,相互排斥标签

范例:检索和添加信息

将文档中的同一规则的信息检索出来

向文档中添加新的结构文档信息

# css-dom

三位一体的网页

结构层

标签,元素

表示层

css

行为层

dom ,javascript

网页的默认表示层和行为层,

如文本的默认样式

鼠标悬停的默认显示title

表示层和行为层的重叠

css 中的属性设置代替dom事件响应

如:hover属性

dom操作css样式的属性

文档里的每个元素就是一个对象

告诉我们节点树上的位置属性

parentNodes,nextSibling,firstChild等

元素本身信息的属性

nodeName,id,nodetype

样式属性

style

element.style.fontFamily:

css样式中,属性命名font-family,在dom 规范中,属性名称为fontFamily;

dom style 不能读取到外部css的样式;

js操作文档样式

有规律的样式展示,

生成的元素样式

大量的规律性的样式赋值

class属性

如果用js直接操作元素样式,建议直接改变元素的class来改变css样式;

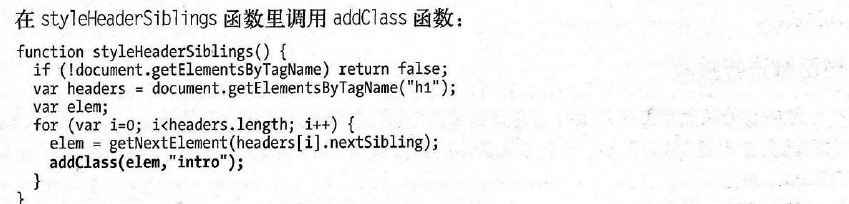
如果是追加元素的样式 ,ele.classname += " intro";//注意前面会多一个空格,(需要首先确定元素原先就含有class)

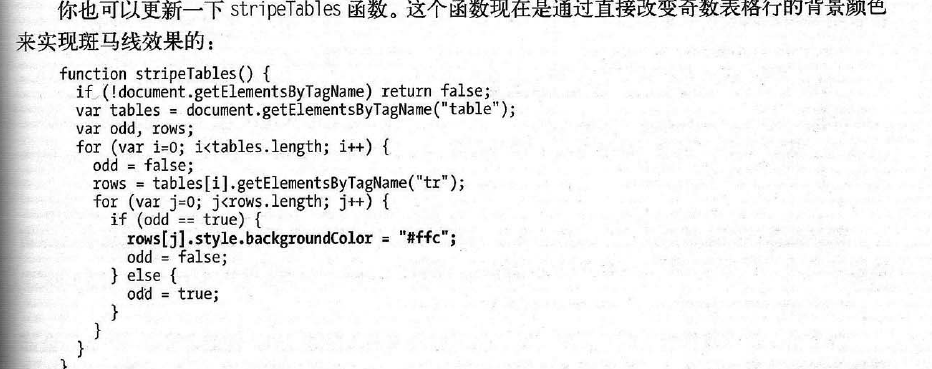
css样式可以实现的,用css;可以控制className的,改变classname达到效果;最后用dom设置具体的样式

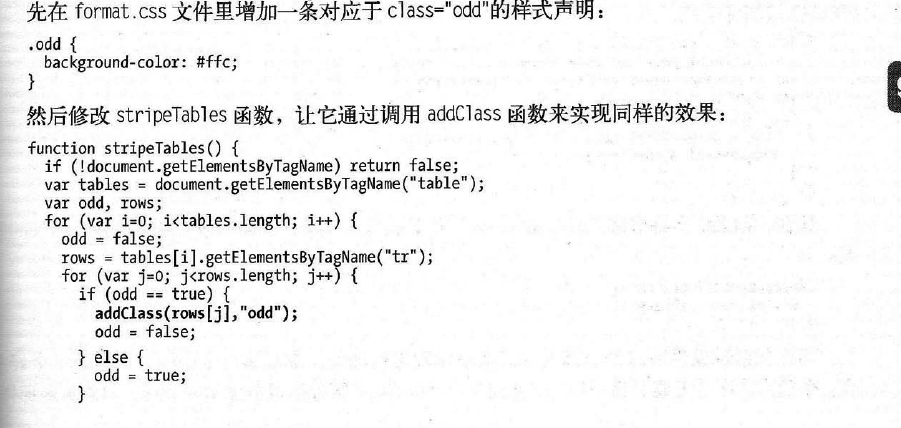
范例:表格的制作











# 用javascript 实现动画效果

位置

position:

==static,根据标记的出现顺序展示出来

fixed,

==relative,与static 相似,区别是可以从正常的显示顺序脱离出来(float应用)

==absolute,父容器要么是文档本身,要么是fixed 或absolute属性的父容器,显示顺序与编写顺序无关

left

top

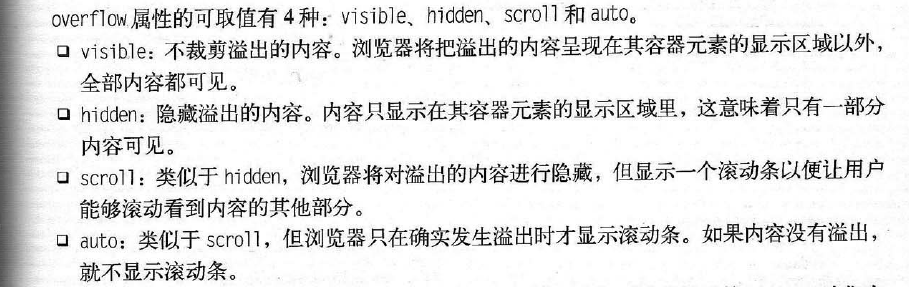
时间

setTimeout() clearsettimeout()

时间递增量

css的overflow属性

用来处理显示对象超过容器时的处理方式



范例:给图片超连接添加一个功能

在鼠标mouseover的时候

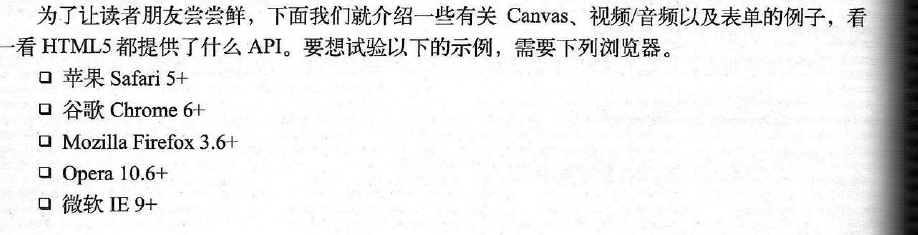
底部显示相应的图片,以动画的方式,所有超链接的动图在这一张图片里

# html5

modernizr



提供对浏览器支持html5功能的检测,检测出哪些支持,哪些不支持;



canvas 画板

画板的api

画板同样不具备可访问性

音频和视频

video 视频

audio 音频

视频格式

mp4:苹果的quicktime 的 mpeg4打包

m4v:同上

avi: 音频视频交错

flv:flash video

视频容器

音频视频的解码器,不同的格式根据不同的解码方式解码;解码经过压缩的视频或音频;

视频编码解码器

h.264 授权

theora

v8 公共 谷歌

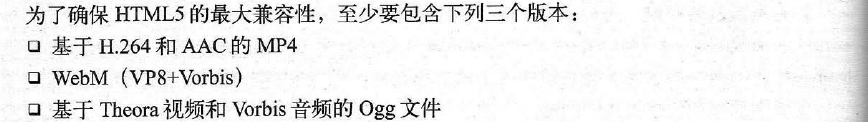
音频编码解码器

mp3

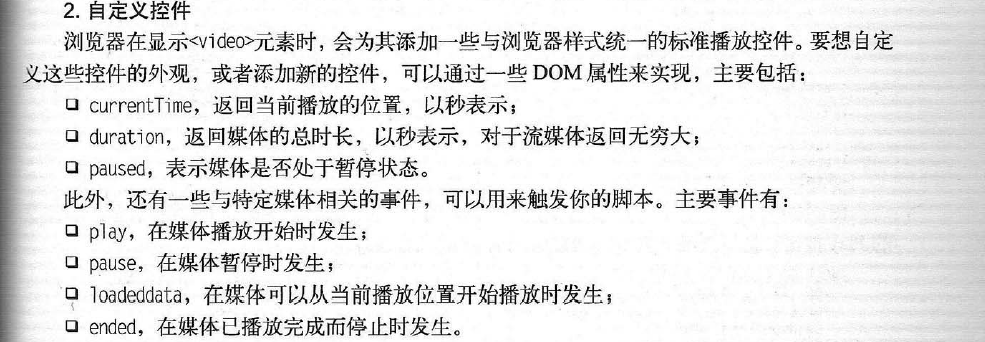
acc

ogg

浏览器视频格式兼容性问题



范例:自定义视频控件



范例:表单

# 综合示例

项目简介

简短的项目说明,一些基本,默认和特殊要求说明

原始资料

图片 视频 音频 文字资料等

站点结构

根据客户提供的资料画一张简单的站点地图;

开始创建项目目录:/img /css /javascript 等

子页面的创建

页面结构

构思每个页面的内容和结构,并添加元素;

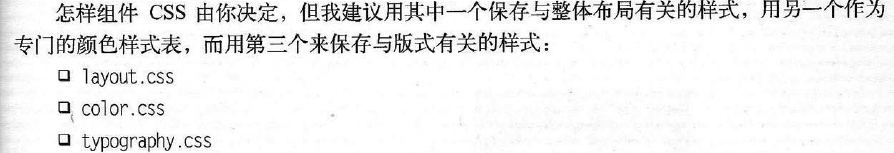
确定每个页面的核心内容和辅助内容

设计

开始设计页面并切图;

填充内容;

css



颜色(color.css):最好都给个背景色

布局(layout.css):块结构设置,边距,距离等;

板式(typograhy.css):文字字体 颜色 大小等相关属性设置;

//标记:为每个页面需要操作的元素命名;

javascript:

根据互动,编写js脚本,响应操作,展示内容等;

# javascript库



寻找合适的库

1.是否满足所有的需求

2.是否过于庞大

3.是否是功能模块化

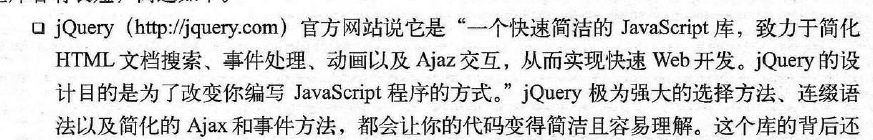
4.是否拥有良好的社区

5.是否有文档可参考;

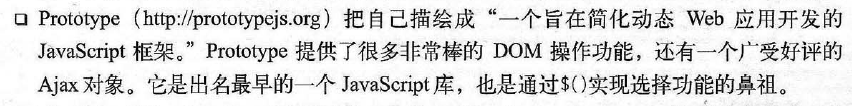
6.使用许可范围

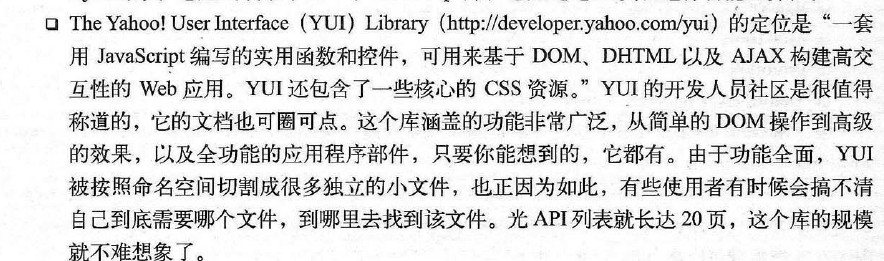
有代表性的库

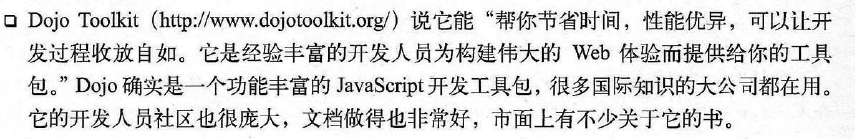
1.jquery

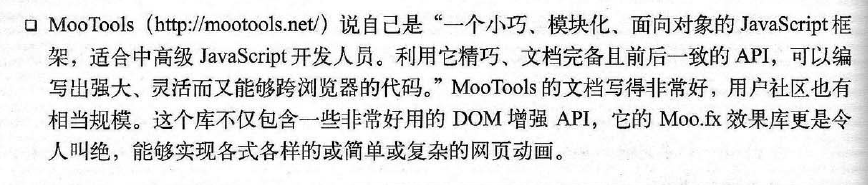


2.Prototype









**Modernizr**

内容分发网络(CDN)

1.一个由服务器构成的网络,这个网络的用途就是分散存储一些公共的内容.

2.每台服务器都包含这个公共内容

3.通过一个公共的url,访问最近的一台服务器;

google免费cdn服务

一些很多库采用的语法

1. .调用方式



2.迭代



3.选择元素

通过css选择器的方式选择元素

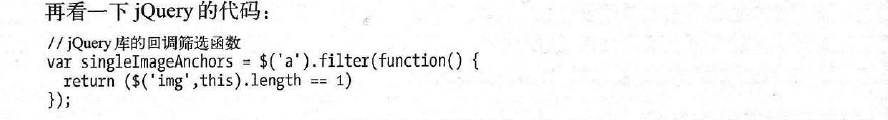
基本的几个



css高级选择器:数量众多,参考网站

库专有选择器

回调函数筛选



操作dom元素

1.生成内容



2.操作内容

处理事件

1.加载事件

2其他事件

ajax

prototype

动画和效果

moof.fx script.aculo.us 基于prototype 的动画库

基于css属性的动画

操作css属性,显示出动画效果

组合动画

打包的动画效果