2015/11/16

1. DHTML不是一项新技术，二十描述HTML、CSS和JavaScript技术组合的术语。
2. 标准化DOM可以让任何一种你程序审核季节语言对使用任何一种编辑语言编写出来的任何一份文档进行操控。
3. HTML文档中特有的注释，以<!—开头，以-->结尾。但是在JavaScript中以<!—开头，只能注释一行，没有结束符。
4. 变量名允许包含字母、数字、美元符号和下划线。第一个字符不能是字母。
5. 通常驼峰格式是函数名、方法名和对象属性名命名的首选格式。
6. 数组对象 var lennon = Object()
7. Var a = false; var b = “”; if(a == b){alert(“正确”);}会执行if语句。。然而，var a = false; var b= “”; if(a === b){alert(“错误”);}不会执行if语句. 全等符号（===）严格执行比较，不仅比较值，还比较变量的类型。同理！=可以使用！==比较严格不相等。
8. 可以养成一个习惯，将变量名用下划线来表示，函数名用驼峰命名法命名
9. 在函数内，使用var为局部变量，不使用为全局变量。
10. Function aquare(num){ total = num\*num; return total; } var tatal=50; var number = square(20); alert(tatal);返回结果是400.改变了全局变量total=50的值。 因此，在定义一个函数时，一定要把内部的变量全都明确地声明为局部变量。
11. 为对象创建一个新的实例，用new关键字。var Jeremy = new Person;
12. 内建对象。比如数组Array（），Math（），Data（）
13. 宿主对象。不是由JavaScript语言本身而是由它的运行环境提供，具体到web应用，这个环境就是浏览器。由浏览器提供的预定义的对象被称为宿主对象。包括Form，Image和Element等，我们可以通过这些对象获得关于网页上表单、图像和各种表单元素等信息。另一种宿主对象也能用来获得网页上的任何一个元素的信息，它就是document对象
14. 用户定义对象。
15. 5个常用的DOM方法：getElementById、getElementByTagName、getElementByClassName、getAttribute、setAttribute 注意：支持DOM的任何一种程序设计语言都可以使用它们。它们用途并非仅限于处理网页，它们可以用来处理用任何一种标记语言（比如XML）编写出来的文档。
16. 三种节点：元素节点、文本节点和属性节点
17. 字符串、数值、布尔类型、函数、对象
18. typeof用来查看某个变量的类型
19. getElementByTagName返回一个数组，即使只有一个元素。
20. 查看某份文档里总共有多少个元素节点，像下面这样使用通配符u即可：alert( document.getElementByTagName(“\*”).lenght );
21. Function getElementsByClassName( node, classname ){

If( node. getElementsByClassName ){

Return node. GetElementsByClassName(classname);

}else{

var results = new Array();

var elems = node. GetElementsByTagName(“\*”);

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

if( elems[i].className.indexOf(classname) !=-1 ){

results[results.length] = elems[i];

}

}

return results;

}

}

其中，node表示DOM树种的搜索起点，第二个classnamme就是要搜索的类名了。最新的浏览器已经支持该函数，为了避免一些老的浏览器不支持的问题。

1. DOM的工作模式：先加载文档的静态内容，再动态刷新，动态刷新不影响文档的静态内容。也就是对页面内容等进行刷新却不需要在浏览器里刷新页面。
2. 第一级DOM。如setAttribute。DOM是一种使用与多种环境和多种程序设计语言的通用型API
3. 事件处理函数.onclick = “showpic( tihs );“注意可以同时添加任意数量的JavaScript语句放到引号间，只要把各条语句用分号隔开即可。
4. <a href=”http://www……” onclick = “return false”>超链接 点击“超链接”三个字后，这个链接的默认行为没有被触发。用这样的方式，可以避免<a>中的默认行径引起的目标链接窗口的打开行为。
5. 在页面上，通过点击链接文字，而在本页的固定位置显示对应的图片。可以利用列表设置超链接列表，然后设置一个空的占位符，利用setAttribute和getAttribute将空的占位符的src设置成超链接的herf属性。注意，完成上述步骤后，并未实现想要达到的目的，原因是链接被点击的默认行为也会被调用，这时候需要将onclick事件处理函数中的调用函数里以调用函数的形似添加return false；
6. nodeType查看节点类型。元素节点的nodeType属性值为1，属性节点的为2，文本节点的为3。
7. <p id=”descript”>这是一个段落 </p> 如果想访问p标签里面的内容，可以采用nodeValue属性。正确的访问方式是：descript.childNode[0].nodeValue 而不是descript.nodeValue。原因是，p标签本身的nodeValue属性是一个空值，而真正需要的是p元素所包含的文本的值。
8. node.childNode[0]可以写成node.firstChild;;;;;;;node.lastChild可以写成node.childNodes[node.childNodes.length-1](不常用)。
9. 平稳退化。如果正确地使用了JavaScript脚本，就可以让访问者在他们的浏览器不支持JavaScript的情况下仍能顺利地浏览你的网站，这就是所谓的平稳退化。

2015/11/17

1. (1)内嵌的处理事件处理函数 <a href=”#” onclick=”popup(‘http://www.example.com/’); return false;”> Example </a>其中#，在某些浏览器中，链接指向当前文档的开头。Href属性值设置为#，只是为了创建一个空链接。

(2)“JavaScript：”为协议 <a href=”javascript:popup(‘http://www.example.com/’);”> Example </a>

以上这两种方式均不能平稳退化。

1. 搜索机器人：是一种自动化的程序，他们浏览web的目的是为了把各种网页添加到搜索引擎的数据库里。各大搜索引擎都有类似的程序。
2. 平稳退化的经典例子。即使JavaScript已被禁用或者遇到了搜索机，该链接均可用。

<a href = <http://www.example.com/> onclick=”pipUp(this.href); return false;”> Example </a>

1. 为了分离内容和样式，<table>和<font>基本已被淘汰。文档结构和文档样式的分离可以确保网页都能平稳退化。
2. 分离JavaScript的例子。

window.onload = prepareLinks;

function prepareLinks(){

if( ! document.getElementsByTagName ) return false;

var links = document.getElementsByTagName(“a”);

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

if( links[ i ].getAttribute(“class”)==”popup”){

links[i].onclick = function(){

popUp( this.getAttribute(“href”));

return false;

}

}

}

}

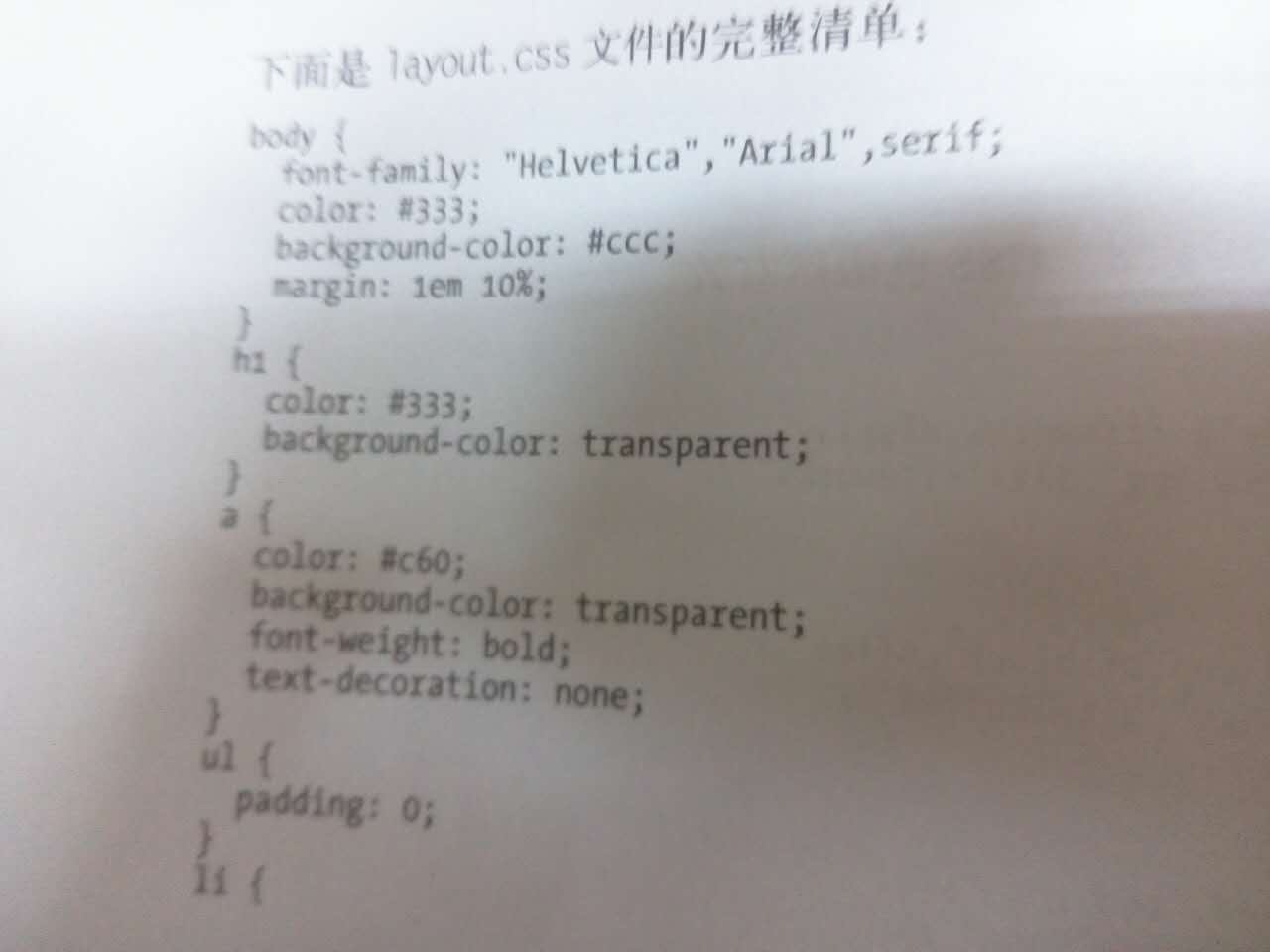
function popup(winURL){

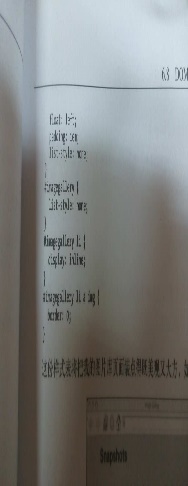
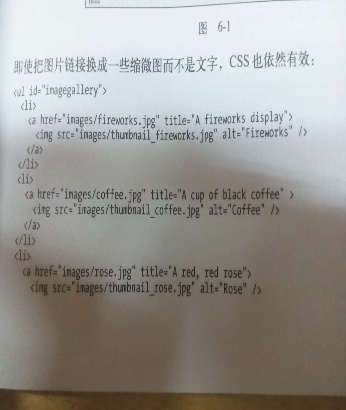
window.open(winURL, “popup”, “width=320, height=480”);

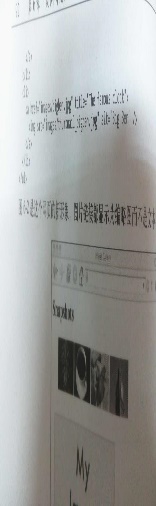
}

如果想把一个事件添加到某个带有特定id属性的元素上，用getElementById就可以解决问题。

如果涉及多个元素，可以用getElementByTagName和getAttribute把事件添加到有着特定属性的一组元素上，具体步骤是：（1）把文档里的所有链接全放入一个数组里（2）遍历数组（3）如果某个链接的class属性为popup，就表示这个链接在被点击时应该调用popUp（）函数

1. 对象检测，是检测浏览器对JavaScript的支持程度，比如支持一部分JavaScript而不支持DOM的浏览器。对象检测时，一定要删掉方法名后面的圆括号，如果不删掉，测试的将是方法的结果，无论方法是否存在。可以添加if( ! document.getElementsById || ! document.getElementByTagName ) return false; 或者 var supported = document. getElementsById || ! document.getElementByTagName ; if(! supported) return;但是就可读性而言，可以分别写，并将return语句加上括号
2. 浏览器嗅探技术，是指通过提取浏览器供应商提供的信息来解决向后兼容问题。获取浏览器的供应商和版本号，并组合匹配，工作量大，且更新快。故逐渐被对象检测技术所取代。
3. 性能考虑
4. 尽量少访问DOM和尽量少标记。
5. 不管什么时候，只要是查询DOM中的某些元素，浏览器都会搜索整个DOM树，从中查找可能匹配的元素。可以把DOM方法查询或者搜索等的结果保持到变量中，以便在循环里重用该结果。
6. 在多个函数都会取得一组类似元素的情况下，可以考虑重构代码，把搜索结果保存在一个全局变量里，或者把一组元素直接以参数形式传递给函数。比如，有一个函数要检查每个链接中的popUp类，另一个函数检查每个链接中的hover类，那么重复做会造成搜索浪费。
7. 尽量减少文档中的标记数量。否则会增加DOM树的规模，进而增加遍历DOM树以查找特定元素的时间。
8. 合并和放置脚本
9. 使用外部文件，不仅能与标记分离，浏览器也能对站点中的多个页面重用缓存过的相同脚本。
10. 将多个外部文件（.js文件）合并到一个脚本文件中，也就是写到一个script标签中，可以减少加载页面时发送的请求数量。
11. Script若放在head块中，会导致癌浏览器无法并行加载其他文件（如图像或其他脚本）。一般来说，根据HTTP规范，浏览器每次从同一个域名中最多中只能同时下载两个文件，而在下载脚本期间，浏览器不会下载其他任何文件，即使是来自不同域名的文件也不会下载，所有其他资源都要等脚本加载完毕后才能下载。
12. 放到</body>前最合适。
13. 压缩脚本
14. 删除脚本中不必要的字节，如空格和注释等。使用更短的变量名、使用综合语句等。
15. 一般要有两个版本，一个是工作副本，可以修改代码并添加注释；另一个是精简副本，用于放在站点上。通常，为了区分，最好在精简副本的文件名中加上min字样。如<script src = “scripts/scriptName.min.js”></script>
16. 一名优秀的网页设计人员总是会在每个细节上问自己这样一个问题：是否还有更好的解决办法。
17. 编写一个函数时需要完成的功能一般有：（1）检查当前浏览器是否理解getElementByTagName、getElementById（2）检查好当前网页是否存在一个id等于imagegallery的元素document.getElementById（“imagegallery”）（3）遍历imagegallery元素中的所有链接（4）设置onclick属性，注意要设置取消链接被点击时的默认行为
18. 结构化程序设计要求函数只有一个入口和一个出口，但是如果一个函数有多个出口，只要这些出口集中出现在函数的开头部分，就是可以接受的。
19. 如果想让页面在加载时得到执行，如果采用window.onload = firstFunction;wiondow.onload = secondFunction;则第二个函数将会取代第一个。因此有了下面两种方法（1）当需要加载的函数比较少时 window.onload = function(){ firstFunction(); secondFunction(); } （2）当需要绑定的函数比较多时，可以用addLoadEvent函数来完成操作。AddLoadEvent函数的代码清单如下：function addLoadEvent( func ) { var oldonload = window.onload; if( typeof window.onload != ‘function’ ) { window.onload = func; } else { window.onload = function() { oldonload();func(); } } }思路是：把现有的window.onload事件处理函数的值存入变量oldonload，如果在这个处理函数上还没有绑定任何函数，就像平时那样把新函数添加给它，如果在这个处理函数上已经绑定了一些函数，就把新函数追加到现有指令的末尾。这样只需要一下代码就行了： addLoadEvent(firstFunction); addLoadEvent(secondFunction);
20. 常用return false来阻拦提交表单或者持续履行下面的代码. return false 只在当前函数有效，不会影响其他外部函数的履行。总的来说在js中对于return用法的三种景象的总结如下：(1)retrun true； 返回正确的处理惩罚成果。(2)return false；返回错误的处理惩罚成果；终止处理惩罚；阻拦提交表单；阻拦履行默认的行动。(3)return；把把握权返回给页面。
21. 如果placeholder图片从标记文档里删除了，则点击imagegellery清单里的哪一个链接，都没有反应，此时的处理方式是，如果showpic中的图片切换成功，则返回true，并阻止默认，如果切换不成功，则返回false，并履行默认操作。这时候采用links[i].onclick = function(){ return !showPic(this) }，并在showPic函数里的末尾添加return true；
22. 如果title属性不存在，那么可以用if语句，在不存在时，将title赋值为空，于是description值为空。
23. 检查placeholder元素是否存在，但是需要假设那是一张图片。为了验证，利用nodeName属性。If(placeholder.nodeName != “IMG”)return false；注意：nodeName属性总是返回一个大小字母的值，即使元素在HTML文档里是小写字母。
24. 在几乎所有的浏览器里，用Tab键移动到某个链接然后按下回车键的动作也会触发onclick事件。
25. DOM Core和HTML-DOM. 之前用到的这些事DOM Core，HTML-DOM的举例。比如说HTML-DOM提供了一个forms对象，或者各种HTML元素的属性。代码如下：Document.getElementByTagName(“form”);简化成document.forms; 或者 element.getAttribute(“scr”);简化成document.scr.
26. 





1. MIME类型application/xhtml+xml与document.write不兼容，浏览器在呈现这种XHTML文档时根本不会执行document.write方法。
2. innerHTML属性不是W3C DOM标准的组成部分。已包含在H5中。
3. innerHTML属性会完全改变html文档中的内容。而setAttribute并未改变文档的物理内容。
4. 文档碎片：那些新创建未被添加到节点树的节点元素。
5. parentElement.insertBefore( newElement, targetElement ).

function insertAfter( newElement, targetElement ){

var parent = targetElement.parentNode;

if( parent.lastChild == targetElement ){

parent.appendChild( newElement );

}else{

Parent.insertBefore( newElement, targertElement.nextSibling);

}

}

1. Ajax依赖JavaScript，搜索引擎的蜘蛛程序也不会主渠道有关内容。浏览器自己发送请求，同时自己相应请求。
2. Function getHTTPObject(){

if(typeof XMLHttpRequest == “undefined”)

XMLHttpRequest = function(){

try{ return new ActiveXObject(“Msxml2.XMLHTTP.6.0”); }

catch(e) { }

try{ return new ActiveXObject(“Msxml2.XMLHTTP.3.0”); }

catch(e) { }

try{ return new ActiveXObject(“Msxml2.XMLHTTP”); }

catch(e) { }

return false;

}

return new XMLHttpRequest();

}

或者，

var xmlhttp;

if (window.XMLHttpRequest)

{// code for IE7+, Firefox, Chrome, Opera, Safari

xmlhttp=new XMLHttpRequest();

}

else

{// code for IE6, IE5

xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");

}

2015/11/18

1. Request.onreadystatechange = doSomething 引用的函数后面不加括号，因为加括号表示立即调用，而我们只是想把函数自身的引用，而不是函数结果，赋值给onreadystatechhange属性
2. 访问服务器发送回来的数据要通过两个属性完成。ResponseText用于保存文本字符串形式的数据。ResponseXML用于保存Content-Type头部中指定为“text/xml”的数据，其实是一个DocumentFragment对象。可以使用DOM方法处理这个对象，这也是XMLHtttpRequest这个名称中有XML的原因
3. 在使用Ajax时，千万要注意同源策略。使用XMLHttpRequest对象发送的请求只能访问其所在的HTML处于同一个域中的数据。此外，有些浏览器还会限制Ajax请求使用的协议。比如Chrome中，如果使用<file://协议从自己的硬盘中加载example.txt>文件，就会看到提示，跨域请求只支持HTTP协议的错误消息。
4. Ajax应用的一个特色就是减少重复加载页面的次数。但这种缺少状态记录的技术会与浏览器的一些使用惯例产生冲突，导致用户无法使用后退按钮或者无法为特定状态下的页面添加书签。
5. Hijax，即渐进增强的Ajax，也被称为“简化的Ajax”，也是hijack（劫持）的意思，也可直接理解成保存和操纵历史的Ajax，Hijax是由Jeremy Keith（杰里米基斯）提出来的。著作权归作者所有。Hijax的原理是“劫持”表单提交和URL链接，然后通过Ajax获取数据后局部更新DOM及内容，从而减少带宽消耗和服务器压力，避免页面刷新带来的闪烁感和重复感，改善用户浏览体验。如果客户端不支持JS或XMLHttpRequest，则表单和链接会按传统方式提交和跳转。
6. Ajax主要依赖于服务器端的处理
7. <abbr>缩略语，如D-O-M；<acronym>首字母缩写词，如dom
8. for( key in defs )其中 definition=defs[key]；如 key= “W3C”，definition=“WorldWideWeb Consortium”
9. dt和dd均属于dl的子节点。注意：dd不是dt的子节点。
10. **137页 新增加的if语句，为了避免低版本ie不支持abbr标签的情况。可是在第5行不是已经可以很好的避免了吗？？还有新增加的第二条if语句，不识别abbr的话，就不会有创建表格，自然没有返回内容，无论返回true还是false，那为什么还要返回一个false呢？**
11. blockquote标签的属性cite表明了块引用的来源。
12. <blockquote>

<p>

</p>

</ blockquote >其中blockquote的lastChild不一定是p节点，有可能是文本节点（换行符）。

1. DocumentFragment 节点不属于文档树，继承的 parentNode 属性总是 null。不过它有一种特殊的行为，该行为使得它非常有用，即当请求把一个 DocumentFragment 节点插入文档树时，插入的不是 DocumentFragment 自身，而是它的所有子孙节点。这使得 DocumentFragment 成了有用的占位符，暂时存放那些一次插入文档的节点。它还有利于实现文档的剪切、复制和粘贴操作，尤其是与 Range 接口一起使用时更是如此。
2. <sup>显示为上标，比如10的平方是 10<sup>2</sup> ，浏览器显示为10的平方是 102。
3. style属性是一个对象
4. 在浏览器里，color属性以RGB格式的颜色值返回
5. Style对象的属性值永远是一个字符串，故而属性值必须加引号，单引号或者双引号均可。
6. 何时应用DOM脚本设置样式（1）根据元素在节点树里的位置来设置样式。可以使用h1~\*选择器为所有的h1元素的下一个同辈元素声明样式（2）根据某种条件反复设置某种样式（3）相应事件。伪类hover一般只支持改变a标签链接的样式，其他的不支持，需要改变更多，采用dom方法
7. NextSibling是得到下一个节点，而不是得到下一个元素节点。

Function getNextElement( node ){ if( node.nodeType == 1 ){ return node; } if(node.nextSibling){ return getNextElement(node,nextSibling); } return null; }

1. **CMS内容管理系统**
2. 在考虑采用css还是dom解决问题的时候，一般考试如下两点：第一，这个问题最简单的解决方案是什么；第二，那种解决方案会得到更多浏览器的支持。
3. 如果只采用elem.className = “intro”则会覆盖掉以前已有的elem元素的class值，如果想要叠加使用，则可以写成elem.className += “intro”；故而，可以写一个添加class的函数

Function addClass( element, value ){ if( !element.className ) { element.className = value } else { newClassName = element.className; newClassName += “ ”;newClassName += value; element.className = newClassName; } }

1. 抽象：把一个非常具体的东西改进为一个较为通用的东西的过程叫做抽象。
2. 动画就是让元素的位置随着时间而不断地发生变化
3. parseInt parseFloat
4. 将onclick改为onmouseover后，图片显示不够流畅，而我们希望能够立刻响应。
5. 如果把position属性值是absolute的元素A放入一个position属性值是relative的元素B，B就成为A的容器元素。而A将在B的显示区域里按absolute方式进行摆放。

2015/11/19

1. 不用双引号包着的是捕捉局部作用域的。双引号中的作用域不捕捉局部变量

var a = function()

{

alert(1111)

}

function abc()

{

var a= function ()

{

alert(2222)

}

setTimeout("a()",3000)//111

setTimeout(a,3000)//222

}

abc()

1. Web设计，最准确的理解是把网页分成三层，结构层，样式层，行为层。或者还可以再加一层，也就是JavaScriptAPI，包括cookie和window等
2. 为了更好地兼容，需要把acronym元素换成abbr标签。
3. 提供了软件Modernizr。一定要放在head元素部分。能提供一些不同的css类名以及特性检测属性。（1）首先，修改html的class属性，给html元素添加一个no-js类。（2）然后检测浏览器可能支持的各种特性。如果浏览器只支持部分特性，则会根据类名，在css中定义相应的增强和退化版本。如： .multiplebgs article p { //为支持多背景浏览器编写的样式}; .no-mutiplebgs article p{ //为不支持多背景浏览器编写的后备样式}
4. 在IE中添加未知元素，必须先使用类似下面的js代码来创建该元素document.creatElement(‘article’)；
5. **Mordernizr怎么用**
6. <video src = “movie.mp4”>

//不支持原声视频时替代的内容

<a href = “movie.mp4”>下载movie.mp4</a>

</video>

1. 为了确保H5最大兼容视频音频，至少要包含以下三个版本（1）基于H.264和AAC的MP4 （2）WebM（VP8+Vorbis） （3）基于Theora视频和Vorbis音频的ogg文件。举例如下：

<video id=”movie” preload controsl>

<source src=”movie.mp4” />

<source src=”movie.webm”type=’video/webm; codecs=”vp8, vorbis”’ />

<source src=”movie.ogv”type=’video/ogg; codecs=”theora, vorbis”’ />

<p>下载movie：

<a href=”movie.mp4”>MP4</a> ,

<a href=”movie.webm”>WebM</a> ,

Or <a href=”movie.ogv”>Ogg</a>

</p>

</video>

1. 对HTML5视频，可以活着将来可以应用css属性以修改视频的外观、大小及形状，可以添加字幕和歌词等信息，还可以组合视频和画布来覆盖内容。甚至可以把视频插入到canvas对象中，像前面处理灰度图片一样，通过分析图像来检测视频运动。
2. 控制video的属性主要由currentTime、duration、paused。主要事件有play、pause、loadeddata、ended。
3. addEventListener是为对象添加事件处理函数的规范方法。IE8以及更早的浏览器使用的是attachEven方法。由于不一致，故而，通常使用onclick之类的事件。但是趋势显示，现如今主流浏览器都支持addEventListener了。
4. 新的属性包括：（1）autocomplete用于为文本（text）输入框添加一组建议的输入项（2）autofocus用于让表单元素自动获取焦点（3）form用于对<form>标签外部的表单元素分组（4）min、max和step用于范围（range）和数值（number）输入框中。（5）pattern用于定义一个正则表达式，一遍验证输入的值（6）placeholder用于在文本输入框中显示临时性的提示信息（7）required表示必填。
5. 为了应对不兼容的浏览器，必须使用特性检测来准备另一个方案。
6. 使用Modernizr库

检查浏览器是否支持某个输入类型的控件。if ( !Modernizr.inputtypes.date ) { //生成日期选择器的脚本 }

检查某个属性。if ( !Modernizr.input.placeholder ) { //生成占位符提示信息的脚本 }

1. 没有Modernizr时，采用inputSupportsType函数

function inputSupportsType( type ) { if ( !document.createElement ) return false; var input = document.createElement(‘input’); input.setAttribute( ‘type’,type ); if( input.type ==’type’&& type != ‘text’ ) { return false; } else { return true; } }

if ( !inputSupportsType( ‘date’ ) ){ //生成日期选择器的脚本 }

function elementSupportAttribute( elementName, attribute ) { if ( !document.createElement ) return false; var temp = document.createElement(elementName); return ( attribute in temp ); }

if( !elementSupportsAttributer( ‘input’, ‘placeholder’ ) ){ //生成占位符提示信息的脚本 }

1. 举例验证浏览器不支持placeholder属性的dom脚本。

if ( !Modernizr.input.placeholder ){ \

var input = document.getElementById(‘first-name’);

input.onfocus = function( ) {

var text = this.placeholder || this.getAttribute( ‘placeholder’);

if（ this.value== text ）{

//重置输入框的值，以隐藏临时的占位符文本

this.value = ‘’;

}

}

Input.onblur = function (){

If(this.value == ‘’){

//把输入框的值设置为占位符文本

This.value = this.placeholder || this.getAttribute(‘placeholder’);

}

}

//在onblur处理函数运行时中添加占位符文本

Input.onblur();

}

2015/11/23

第十二章 综合实例

1. 首先确定所要设计的网站需要几个网页，本例需要五个Home、About、Photos、Live和Contact。
2. 为各个页面创建模板。页面结构：头部head，包含logo和导航nav；内容区article。注意：因为使用H5，故而head元素中包**含Modernizr库。**写成模板代码。
3. 开始插入内容
4. 页面设计切分成多个图片。北京图片保存为background.gif，品牌图像保存为logo.gif，带有一点渐变的导航表要命名为navbar.gif，最后任务剪影保存为guitarist.gif。将所有图片放到images文件夹中。
5. Html中用link加入css文件时，加入了属性media=“screen”
6. 颜色，记住：不管哪个元素应用什么什么颜色，都要同时给它一个背景色，否则，就有可能导致意外，看不到某些文本.编辑color.css
7. 布局,设置layout.css 使用通配符选择器吧所有元素的内外边距设置为零.内边距信息放在layout.css里面
8. 版式,编辑typography.css 外边距,字体和大小放在这里面.注意内外边距的分类不是很严格,也即是说外边距也可以放到layout.css中

2015/11/26

1. 样式表color.css是最直观的，不管哪个元素用什么颜色，都要同时给它一个背景色，否则，就有可能导致意外，看不到某些文本。
2. 首先为H5元素定义默认的样式，针对浏览器的兼容问题。然后，使用通配选择器将所有元素的内外边距设置为零。
3. Background-attachment：fixed；意识是背景图片不跟随滚动条滚动，是固定不动的。
4. 实际网站开发时，不太可能一页页地编辑导航链接，更常见的做法是通过服务器端包含技术，把包含导航标记的片段插入到每个页面中。
5. 服务器端包含可以使用Apache Server Side Includes（SSIs）、PHP、ASP，或者其他服务器端语言。服务器端包含的优点是可以把重用标记块集中保存。
6. string.indexOf（substring）
7. 当由于作用域的原因，无法在函数外传值时，可以为每个链接创建一个自定义的属性，比如把这个属性明明为destination，把局部变量sectionId的值赋给它，links[i].destination = sectionId; 这个属性的作用域是持久的。
8. label通过for属性把一小段文本关联到表单的一个字段。很多浏览器中label元素的默认行为是，如果label中的文本被点击，关联的表单字段就会获得焦点。注意，并不是所有的浏览器都实现了该行为。如果不支持，我们如何自己实现该行为呢？如下：

function focusLabels(){

if(!document.getElementsByTagName) return false;

var labels = document.getElementsByTagName(“label”);

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

if(!labels[i].getAttribute(“for”)) continue;

labels[i].onclick = function(){

var id = this.getAttribute(“for”);

if(!document.getElementById(id))return false;

var element = document.getElementById(id);

element.focus();

}

}

}

1. 文档中的每个表单元素都是一个form对象，每个form对象都有一个elements.length属性。Form.elementss.length返回表单中的包含的表单元素的个数。
2. 初始化一个placeholder属性，可以取得这些占位符的值，并临时将他们作为相应表单字段的value。在该字段获得焦点时，再删除字段的value值。类似的，如果用户没有在字段中输入文本且离开了当前字段，那么再重新应用占位符值即可。例如：

function resetFields(whichform){

if(Modernizr.input.placeholder) return;

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

var element =whichform.elements[i];

if(element.type == “submit”) continue;

var check = element.placeholder || element.getAttribute(‘placeholder’);

if(!check) continue;

element.onfocus = function(){

var text = this.placeholder || this.getAttribute(‘placeholder’);

if(this.value == ‘text’){

this.className = ‘’;

this.value = ‘’;

}

}

element.onblur = function(){

if(this.value == “”){

this.className = “placeholder”;

this.value = this.placeholder || this.getAttribute(‘placeholder’);

}

}

Element.onblur();

}

}

1. Request For Comments (RFC），是一系列以编号排定的文件。文件收集了有关因特网相关资讯，以及UNIX和因特网社群的软件文件。目前RFC文件是由Internet Society（ISOC）所赞助发行。因此几乎所有的互联网标准都有收录在RFC文件之中。
2. 验证表单是否填写完成

function ifFilled(field){

if(fiele.value.replace(‘ ’,‘’).length == 0) return false;

var placeholder = field.placeholder || field.getAttribute(‘placeholder’);

return(field.value != placeholder);

}

function isEmail(field){

return(field.value.indexOf(“@”) != -1 && field.value.indexOf(“.”) != -1);

}

1. 可以使用JavaScript的encodeURIComponent函数把这些值编码成URL安全的字符串，它会把有歧义的字符串转换成ASCII编码。
2. var dataParts = [];
3. var element;
4. for(var I = 0;i<whichform.elements.length; i++){

element = whichform.elements[i];

dataParts[i] = element.name + ‘=’+encodeURIComponent(element.value);

var data = dataParts.join(‘&’);用&号连接起来

}

request.open(‘POST’,whichform.getAttribute(“action”), true);

request.setRequestHeader(“Content-type”,”application/x-www-form-urlencoded”);这个头部信息对于POST请求是必需的，它表示请求中包含URL编码的表单。

1. 对于库来说，如果有多个站点要使用同一个库，那么最好是把这个库托管到一个公共服务器上，以便所有站点共享和访问。这样，当用户从一个站点跳到另一个站点时，他们就不用再冲农夫下载相同的文件了。内容分发网络CDN解决分布共享库的问题。
2. 为了防止CDN不可用，可以添加本地文件的script标签。

<script src = “https://ajax.googleapis.com/ajax/libs/jquery/1.4.3/jquery.min.js”></script>

<script>!window.jQuery && document.write(unescape(‘%3Cscript src=”scripts/jquery-1.4.3.min.js”%3E%3C/script%3E’)) </script>

1. $(‘li’).each(function(i){ })其中的each是每一个li元素
2. 274页 为什么？$（’img’,this）
3. $(‘ul#list1 li’).appendTo(“ul#list2”);转移元素

$(‘ul#list1 li’).clone(),appendTo(“ul#list2”);复制元素

1. $.post(url, params, callback);$.get();$getJSON();$getScript(); 他们的回调方法总会被作为$.ajax()的成功回调方法调用。每个回调方法都接受两个参数，分别是请求对象的相应文本responseText和状态status。状态是以下几个值之一：success、error、notmodified。在使用getJSON和getScript方法时相应会被求值，因此getJSON方法中传给回调的参数是一个JavaScript对象。