--

[理论方面 1](#_Toc22669)

理论方面

|  |  |
| --- | --- |
| 使用注意点 | 1. 尽可能的保持代码简洁  可能大家都听到过了N遍这个代码简洁问题了。作为一个开发人员你可能在你的代码开发过程中使用了很多次，但千万不要在 js 开发中忘记这点。  尽量在开发模式中添加注释和空格，这样保持代码的可读性  在发布到产品环境前请将空格和注释都删除，并且尽量缩写变量和方法名  使用第三方工具帮助你实现压缩 javascript。  4. 避免 Eval  你的 JS 在没有 eval 方法的时候也可以很好的工作。eval 允许访问 javascript 编译器。如果一个字符串作为参数传递到 eval，那么它的结果可以被执行。  这会很大的降低代码的性能。尽量避免在产品环境中使用 eval。  5. 最小化 DOM 访问  DOM 是最复杂的 API，会使得代码执行过程变慢。有时候 web 页面可能没有加载或者加载不完整。最好避免 DOM。  9. 尽量不要多次使用 var  在初始每一个变量的时候，程序员都习惯使用 var 关键字。相反，建议你使用逗号来避免多余的关键字，并且减少代码体积。 如下：  var variableOne = 'string 1',  variableTwo = 'string 2',  variableThree = 'string 3';    10. 不要忽略分号 “;”  这往往是大家花费数个小时进行 debug 的原因之一。 |
|  | 一、请选择出前端页面开发，执行javascript代码alert(1)的代码：  成功alert(1) <iframe src="javascript:alert(1);"></iframe> 成功alert(1) <img src=""/onerror=alert(1);> ie下兼容性视图成功alert(1) <s style="top:expression(alert(1))"></s> 没有alert(1) <div onclick="alert(1)"></div>    二、ECMAScript5新增的特性：  　　2.Object.create  　　3.Object.preventExtensions  　　4.Object.isExtensible  参考资料：<http://blog.csdn.net/kabukon/article/details/6913381> |
| JavaScript之 DOM + BOM 常用方法 68 | 10. 设置cookie值  function setCookie(name, value, Hours) {  var d = new Date();  var offset = 8;  var utc = d.getTime() + (d.getTimezoneOffset() \* 60000);  var nd = utc + (3600000 \* offset);  var exp = new Date(nd);  exp.setTime(exp.getTime() + Hours \* 60 \* 60 \* 1000);  document.cookie = name + "=" + escape(value) + ";path=/;expires=" + exp.toGMTString() + ";domain=360doc.com;"  }  11. 获取cookie值  function getCookie(name) {  var arr = document.cookie.match(new RegExp("(^| )" + name + "=([^;]\*)(;|$)"));  if (arr != null) return unescape(arr[2]);  return null  }  12. 加入收藏夹  function AddFavorite(sURL, sTitle) {  try {  window.external.addFavorite(sURL, sTitle)  } catch(e) {  try {  window.sidebar.addPanel(sTitle, sURL, "")  } catch(e) {  alert("加入收藏失败，请使用Ctrl+D进行添加")  }  }  }  13. 设为首页  function setHomepage() {  if (document.all) {  document.body.style.behavior = 'url(#default#homepage)';  document.body.setHomePage('http://w3cboy.com')  } else if (window.sidebar) {  if (window.netscape) {  try {  netscape.security.PrivilegeManager.enablePrivilege("UniversalXPConnect")  } catch(e) {  alert("该操作被浏览器拒绝，如果想启用该功能，请在地址栏内输入 about:config,然后将项 signed.applets.codebase\_principal\_support 值该为true")  }  }  var prefs = Components.classes['@mozilla.org/preferences-service;1'].getService(Components.interfaces.nsIPrefBranch);  prefs.setCharPref('browser.startup.homepage', 'http://w3cboy.com')  }  }  14. 加载样式文件  function LoadStyle(url) {  try {  document.createStyleSheet(url)  } catch(e) {  var cssLink = document.createElement('link');  cssLink.rel = 'stylesheet';  cssLink.type = 'text/css';  cssLink.href = url;  var head = document.getElementsByTagName('head')[0];  head.appendChild(cssLink)  }  }  15. 返回脚本内容  function evalscript(s) {  if(s.indexOf('<script') == -1) return s;  var p = /<script[^\>]\*?>([^\x00]\*?)<\/script>/ig;  var arr = [];  while(arr = p.exec(s)) {  var p1 = /<script[^\>]\*?src=\"([^\>]\*?)\"[^\>]\*?(reload=\"1\")?(?:charset=\"([\w\-]+?)\")?><\/script>/i;  var arr1 = [];  arr1 = p1.exec(arr[0]);  if(arr1) {  appendscript(arr1[1], '', arr1[2], arr1[3]);  } else {  p1 = /<script(.\*?)>([^\x00]+?)<\/script>/i;  arr1 = p1.exec(arr[0]);  appendscript('', arr1[2], arr1[1].indexOf('reload=') != -1);  }  }  return s;  }  16. 清除脚本内容  function stripscript(s) {  return s.replace(/<script.\*?>.\*?<\/script>/ig, '');  }  17. 动态加载脚本文件  function appendscript(src, text, reload, charset) {  var id = hash(src + text);  if(!reload && in\_array(id, evalscripts)) return;  if(reload && $(id)) {  $(id).parentNode.removeChild($(id));  }    evalscripts.push(id);  var scriptNode = document.createElement("script");  scriptNode.type = "text/javascript";  scriptNode.id = id;  scriptNode.charset = charset ? charset : (BROWSER.firefox ? document.characterSet : document.charset);  try {  if(src) {  scriptNode.src = src;  scriptNode.onloadDone = false;  scriptNode.onload = function () {  scriptNode.onloadDone = true;  JSLOADED[src] = 1;  };  scriptNode.onreadystatechange = function () {  if((scriptNode.readyState == 'loaded' || scriptNode.readyState == 'complete') && !scriptNode.onloadDone) {  scriptNode.onloadDone = true;  JSLOADED[src] = 1;  }  };  } else if(text){  scriptNode.text = text;  }  document.getElementsByTagName('head')[0].appendChild(scriptNode);  } catch(e) {}  }  18. 返回按ID检索的元素对象  function $(id) {  return !id ? null : document.getElementById(id);  }  19. 跨浏览器绑定事件  function addEventSamp(obj,evt,fn){  if(!oTarget){return;}  if (obj.addEventListener) {  obj.addEventListener(evt, fn, false);  }else if(obj.attachEvent){  obj.attachEvent('on'+evt,fn);  }else{  oTarget["on" + sEvtType] = fn;  }  }  20. 跨浏览器删除事件  function delEvt(obj,evt,fn){  if(!obj){return;}  if(obj.addEventListener){  obj.addEventListener(evt,fn,false);  }else if(oTarget.attachEvent){  obj.attachEvent("on" + evt,fn);  }else{  obj["on" + evt] = fn;  }  }  21. 为元素添加on方法  Element.prototype.on = Element.prototype.addEventListener;    NodeList.prototype.on = function (event, fn) {、  []['forEach'].call(this, function (el) {  el.on(event, fn);  });  return this;  };  22. 为元素添加trigger方法  Element.prototype.trigger = function (type, data) {  var event = document.createEvent('HTMLEvents');  event.initEvent(type, true, true);  event.data = data || {};  event.eventName = type;  event.target = this;  this.dispatchEvent(event);  return this;  };    NodeList.prototype.trigger = function (event) {  []['forEach'].call(this, function (el) {  el['trigger'](event);  });  return this;  };  27. 判断是否移动设备  function isMobile(){  if (typeof this.\_isMobile === 'boolean'){  return this.\_isMobile;  }  var screenWidth = this.getScreenWidth();  var fixViewPortsExperiment = rendererModel.runningExperiments.FixViewport ||rendererModel.runningExperiments.fixviewport;  var fixViewPortsExperimentRunning = fixViewPortsExperiment && (fixViewPortsExperiment.toLowerCase() === "new");  if(!fixViewPortsExperiment){  if(!this.isAppleMobileDevice()){  screenWidth = screenWidth/window.devicePixelRatio;  }  }  var isMobileScreenSize = screenWidth < 600;  var isMobileUserAgent = false;  this.\_isMobile = isMobileScreenSize && this.isTouchScreen();  return this.\_isMobile;  }  28. 判断是否移动设备访问  function isMobileUserAgent(){  return (/iphone|ipod|android.\*mobile|windows.\*phone|blackberry.\*mobile/i.test(window.navigator.userAgent.toLowerCase()));  }  29. 判断是否苹果移动设备访问  function isAppleMobileDevice(){  return (/iphone|ipod|ipad|Macintosh/i.test(navigator.userAgent.toLowerCase()));  }  30. 判断是否安卓移动设备访问  function isAndroidMobileDevice(){  return (/android/i.test(navigator.userAgent.toLowerCase()));  }  31. 判断是否Touch屏幕  function isTouchScreen(){  return (('ontouchstart' in window) || window.DocumentTouch && document instanceof DocumentTouch);  }  32. 判断是否打开视窗  function isViewportOpen() {  return !!document.getElementById('wixMobileViewport');  }  33. 获取移动设备初始化大小  function getInitZoom(){  if(!this.\_initZoom){  var screenWidth = Math.min(screen.height, screen.width);  if(this.isAndroidMobileDevice() && !this.isNewChromeOnAndroid()){  screenWidth = screenWidth/window.devicePixelRatio;  }  this.\_initZoom = screenWidth /document.body.offsetWidth;  }  return this.\_initZoom;  }  34. 获取移动设备最大化大小  function getZoom(){  var screenWidth = (Math.abs(window.orientation) === 90) ? Math.max(screen.height, screen.width) : Math.min(screen.height, screen.width);  if(this.isAndroidMobileDevice() && !this.isNewChromeOnAndroid()){  screenWidth = screenWidth/window.devicePixelRatio;  }  var FixViewPortsExperiment = rendererModel.runningExperiments.FixViewport || rendererModel.runningExperiments.fixviewport;  var FixViewPortsExperimentRunning = FixViewPortsExperiment && (FixViewPortsExperiment === "New" || FixViewPortsExperiment === "new");  if(FixViewPortsExperimentRunning){  return screenWidth / window.innerWidth;  }else{  return screenWidth / document.body.offsetWidth;  }  }  35. 获取移动设备屏幕宽度  function getScreenWidth(){  var smallerSide = Math.min(screen.width, screen.height);  var fixViewPortsExperiment = rendererModel.runningExperiments.FixViewport || rendererModel.runningExperiments.fixviewport;  var fixViewPortsExperimentRunning = fixViewPortsExperiment && (fixViewPortsExperiment.toLowerCase() === "new");  if(fixViewPortsExperiment){  if(this.isAndroidMobileDevice() && !this.isNewChromeOnAndroid()){  smallerSide = smallerSide/window.devicePixelRatio;  }  }  return smallerSide;  }  36.  37. getElementsByClassName  function getElementsByClassName(name) {  var tags = document.getElementsByTagName('\*') || document.all;  var els = [];  for (var i = 0; i < tags.length; i++) {  if (tags.className) {  var cs = tags.className.split(' ');  for (var j = 0; j < cs.length; j++) {  if (name == cs[j]) {  els.push(tags);  break  }  }  }  }  return els  }  38. 获取页面高度  function getPageHeight(){  var g = document, a = g.body, f = g.documentElement, d = g.compatMode == "BackCompat"  ? a  : g.documentElement;  return Math.max(f.scrollHeight, a.scrollHeight, d.clientHeight);  }  39. 获取页面可视宽度  function getPageViewWidth(){  var d = document, a = d.compatMode == "BackCompat"  ? d.body  : d.documentElement;  return a.clientWidth;  }  40. 获取页面宽度  function getPageWidth(){  var g = document, a = g.body, f = g.documentElement, d = g.compatMode == "BackCompat"  ? a  : g.documentElement;  return Math.max(f.scrollWidth, a.scrollWidth, d.clientWidth);  }  41. 获取页面scrollTop  function getPageScrollTop(){  var a = document;  return a.documentElement.scrollTop || a.body.scrollTop;  }  42. 获取页面可视高度  function getPageViewHeight() {  var d = document, a = d.compatMode == "BackCompat"  ? d.body  : d.documentElement;  return a.clientHeight;  }  46. 确认是否键盘有效输入值  function checkKey(iKey){  if(iKey == 32 || iKey == 229){return true;}/\*空格和异常\*/  if(iKey>47 && iKey < 58){return true;}/\*数字\*/  if(iKey>64 && iKey < 91){return true;}/\*字母\*/  if(iKey>95 && iKey < 108){return true;}/\*数字键盘1\*/  if(iKey>108 && iKey < 112){return true;}/\*数字键盘2\*/  if(iKey>185 && iKey < 193){return true;}/\*符号1\*/  if(iKey>218 && iKey < 223){return true;}/\*符号2\*/  return false;  }  47. 获取网页被卷去的位置  function getScrollXY() {  return document.body.scrollTop ? {  x: document.body.scrollLeft,  y: document.body.scrollTop  }: {  x: document.documentElement.scrollLeft,  y: document.documentElement.scrollTop  }  }  48.  49.  50. 解决offsetX兼容性问题  // 针对火狐不支持offsetX/Y  function getOffset(e){  var target = e.target, // 当前触发的目标对象  eventCoord,  pageCoord,  offsetCoord;  // 计算当前触发元素到文档的距离  pageCoord = getPageCoord(target);  // 计算光标到文档的距离  eventCoord = {  X : window.pageXOffset + e.clientX,  Y : window.pageYOffset + e.clientY  };  // 相减获取光标到第一个定位的父元素的坐标  offsetCoord = {  X : eventCoord.X - pageCoord.X,  Y : eventCoord.Y - pageCoord.Y  };  return offsetCoord;  }  function getPageCoord(element){  var coord = { X : 0, Y : 0 };  // 计算从当前触发元素到根节点为止，  // 各级 offsetParent 元素的 offsetLeft 或 offsetTop 值之和  while (element){  coord.X += element.offsetLeft;  coord.Y += element.offsetTop;  element = element.offsetParent;  }  return coord;  }  51.  52. 返回顶部的通用方法  function backTop(btnId) {  var btn = document.getElementById(btnId);  var d = document.documentElement;  var b = document.body;  window.onscroll = set;  btn.style.display = "none";  btn.onclick = function() {  btn.style.display = "none";  window.onscroll = null;  this.timer = setInterval(function() {  d.scrollTop -= Math.ceil((d.scrollTop + b.scrollTop) \* 0.1);  b.scrollTop -= Math.ceil((d.scrollTop + b.scrollTop) \* 0.1);  if ((d.scrollTop + b.scrollTop) == 0) clearInterval(btn.timer, window.onscroll = set);  }, 10);  };  function set() {  btn.style.display = (d.scrollTop + b.scrollTop > 100) ? 'block': "none"  }  };  backTop('goTop');  53.  54. 打开一个窗体通用方法  function openWindow(url,windowName,width,height){  var x = parseInt(screen.width / 2.0) - (width / 2.0);  var y = parseInt(screen.height / 2.0) - (height / 2.0);  var isMSIE= (navigator.appName == "Microsoft Internet Explorer");  if (isMSIE) {  var p = "resizable=1,location=no,scrollbars=no,width=";  p = p+width;  p = p+",height=";  p = p+height;  p = p+",left=";  p = p+x;  p = p+",top=";  p = p+y;  retval = window.open(url, windowName, p);  } else {  var win = window.open(url, "ZyiisPopup", "top=" + y + ",left=" + x + ",scrollbars=" + scrollbars + ",dialog=yes,modal=yes,width=" + width + ",height=" + height + ",resizable=no" );  eval("try { win.resizeTo(width, height); } catch(e) { }");  win.focus();  }  }  55.  56.  57.  58.  59. 清除html代码中的脚本  function clear\_script(){  K1.value=K1.value.replace(/<script.\*?>[\s\S]\*?<\/script>|\s+on[a-zA-Z]{3,16}\s?=\s?"[\s\S]\*?"|\s+on[a-zA-Z]{3,16}\s?=\s?'[\s\S]\*?'|\s+on[a-zA-Z]{3,16}\s?=[^ >]+/ig,"");  }  动态执行JavaScript脚本  function javascript(){  try{  eval(K1.value);  }catch(e){  alert(e.message);  }  }  60. 动态执行VBScript脚本  function vbscript(){  try{  var script=document.getElementById("K1").value;  if(script.trim()=="")return;  window.execScript('On Error Resume Next \n'+script+'\n If Err.Number<>0 Then \n MsgBox "请输入正确的VBScript脚本!",48,"脚本错误!" \n End If',"vbscript")  }catch(e){  alert(e.message);  }  }  61.  62. resize的操作  (function(){  var fn = function(){  var w = document.documentElement ? document.documentElement.clientWidth : document.body.clientWidth  ,r = 1255  ,b = Element.extend(document.body)  ,classname = b.className;  if(w < r){  //当窗体的宽度小于1255的时候执行相应的操作  }else{  //当窗体的宽度大于1255的时候执行相应的操作  }  }  if(window.addEventListener){  window.addEventListener('resize', function(){ fn(); });  }else if(window.attachEvent){  window.attachEvent('onresize', function(){ fn(); });  }  fn();  })();  63.  64.  65. 获取窗体可见范围的宽与高  function getViewSize(){  var de=document.documentElement;  var db=document.body;  var viewW=de.clientWidth==0 ? db.clientWidth : de.clientWidth;  var viewH=de.clientHeight==0 ? db.clientHeight : de.clientHeight;  return Array(viewW ,viewH);  }  66. 断鼠标是否移出事件  function isMouseOut(e, handler) {  if (e.type !== 'mouseout') {  return false;  }  var reltg = e.relatedTarget ? e.relatedTarget : e.type === 'mouseout' ? e.toElement : e.fromElement;  while (reltg && reltg !== handler) {  reltg = reltg.parentNode;  }  return (reltg !== handler);  }  67.  68. |
| 犀牛书中小问题总结 | 补充： 首先强调犀牛书很牛逼,到底有多牛逼呢,很多面试笔试的问题,其实犀牛书中都有涉及,部分问题有详解.俺很无聊的把知识测试部分给扣下来了(2章到14章),有需要的同学可以看看啦,手打难免出错,大家多多纠正.鞠躬! chapter2 1.在下列的变量名中,哪些是合法的变量名?哪些是非法变量名?为什么 $someVariable \_someVariable 1Variable some\_Variable function  some\*variable ----答案---- 合法: $someVariable \_someVariable some\_Variable 非法: 1Variable 首字母是非法字母 function  保留字 some\*variable  包含非法字符\*  2.将下列标示符转为骆驼拼写法 var some\_month; function theMonth  //返回当前月份的函数 current-month  //常量 var summer\_month;//夏季月份的数组 MyLibrary-afunction//javascript包中的某个函数 ----答案---- var someMonth; function getCurrentMonth  CURRENT\_MONTH  summerMonths myLibraryFunction  3.下列字符串是否合法？如果不合法，如何修改？ var someString='who once said,"only two things are infinite,the universe and human stupidity,and I'm not sure about the former."' ----答案---- 字符串非法，修改方法为在双引号中只是用单引号或者转义双引号  var someString="who once said,'only two things are infinite,the universe and human stupidity,and I'm not sure about the former.'"  var someString='who once said,"only two things are infinite,the universe and human stupidity,and I\'m not sure about the former."'  4.对于给定的数字432.54，用什么javascript函数可以返回该数字的整数部分？如何 转换为八进制和十进制？ ----答案---- var fltNumber=432.54; var intNumber=parseInt(fltNumber); var octNumber=intNumber.toString(8); var hexNumber=intNumber.toString(16);  5. ================================================= chapter3 1.请为下列的表达式添加圆括号，从而使得表达式的结果为8 var valA=37; var valB=3; var valC=18; var resultOfComp=varA-varB%3/2\*4+varC-3; ----答案---- var resultOfComp=(varA-varB)%3/2\*(4+varC)-3;  2.请使用switch语句，判断表达式的数值是否为'one','two'或者'three',当表达式为'one'或者'two' 的时候将变量的值设置为'OK';当表达式的值为'three'的时候，将变量值设置为'OK2';如果没有匹配的 值，就将变量的值设置为'NONE'。 ----答案---- switch(val){ case'one': case'two':         result='OK';         break; case'three':         result='OK2';         break; default:         result='NONE'; }  3.假设有3个变量，varOne,varTwo,varThree。那么如何对这三个变量进行判断，从而仅当 varOne为33，varTwo小于等于100,但是varThree大于0的时候执行某段代码？ ----答案---- if((varOne==33)&&(varTwo<=100)&&(varThree>0))  4.如果希望执行某段代码6次，有哪3种方式？对这3种方式如何取舍？ ----答案---- for(var i=0;i<6;i++){...}  i=0; while(i<6){... i++;}  i=0; do{i++;...}while(i<6) do...while适合不论条件是否成立都至少要执行一次代码块的情况。 for循环适合代码块需要执行特定的次数的情况。 while循适合代码块会影响条件表达式的判断的情况。  5.下列条件表达式是否有问题？如果有，那么是什么问题？ if(valTest1==valTest2)... ----答案---- 如果不确定变量的数据类型，那么请考虑使用严格相等运算符 ===========================================================  chapter4 1.以逗号分隔的字符串是常见的数据格式。如何根据逗号分隔的字符串创建数组呢？ 请以下面的字符串创建一个数组，并访问第三个元素： "cats,dogs,birds,horses" ----答案---- 使用String.split方法，传入逗号作为分隔符，如 var animalString="cats,dogs,birds,horses"; var animalArray=String.split(animalString,","); alert(animalArray[2]);  2.特殊字符\b是表示分隔单词的符号，而\B则是表示非分割单词符号。请定义一个正则表达式， 在下面的字符串中查找所有单词fun，并将其替换成power。 "The fun of function is that they are functional." ----答案---- var funPattern=/\bfun\b/; var strToSearch="The fun of function is that they are functional."; var afterMatch=strToSearch.replace(funPattern,"power");  3.编写一段代码，获取今天的日期并计算下一周同一天的日期。 ----答案---- var dtNow=new Date(); var hours=dtNow.getHours(); hours+=168;//7\*24 dtNow.setHours(hours); document.writeln(dtNow.toString());  4.编写一段代码，分别对数字34.44执行向上和向下取整操作。 ----答案---- var baseNum=34.44; var numFloor=Math.floor(baseNum);//Math.floor可以对数字执行向下取整 var numCeil=Math.ceil(baseNum);//Math.ceil可以对数字执行向上取整  5.根据下面的字符串，利用模式匹配将已有的标点符号替换成逗号，然后作为数组载入， 并输出每个数值: var str="apple.orange-strawberry,lemon-.lime"; ----答案---- var strToAlter="apple.orange-strawberry,lemon-.lime"; var puncPattern=/[\.|-]/g; var afterMatch=strToAlter.replace(puncPattern,","); var fruits=afterMatch.split(','); for(var i=0;i<fruits.length;i++){         document.writeln(fruits);         } ===========================================================  chapter5 1.对一个数字(n)执行阶乘操作，将得到1~n的数字乘积，通常写做3!(1\*2\*3或者6)。 或者编写一个javascript函数，使用递归的方法计算出指定数字的阶乘。 ----答案---- function findFactorial(n){ if(n==0)return 1; return (n\*findFactorial(n-1)); } var num=findFactorial(4);//将返回24  2.函数如何才能修改其作用域之外的变量？编写一个函数，由1~5的数字组成的数组作为 参数，调用该函数后把其中的数字项替换成相应的字符串表示形式(也就是"one","two"等) ----答案---- 如果将一个对象(如一个数组)作为函数的参数传入，那么在函数中对这个数组的修改就会 反映到函数外部。针对这个问题，其中一种解决办法是： function makeArray(){ var arr=[1,5,3]; alert(arr); alterArray(arr); alert(arr); }  function alterArray(arrofNumbers){ for(var i=0;i<arrofNumbers.length;i++){    switch(arrofNumbers){         case 1:arrofNumbers="one";break;         case 2:arrofNumbers="two";break;         case 3:arrofNumbers="three";break;         case 4:arrofNumbers="four";break;         case 5:arrofNumbers="five";break; }  }  }  3.创建一个函数，它的参数是一个数据对象和一个函数，它将对这个数据对象调用该函数。 ----答案---- 使用匿名函数满足需求 function invokeFunction(dataObject,functionToCall){ functionToCall(dataObject); } var funcCall=new Function('x','alert(x)'); invokeFunction('hello',funcCall); =========================================================== chapter6 1.编写一段代码，用来测试某个变量的值，不得使用特定于浏览器的调试器。 ----答案---- 要测试一个变量的值最快的方法是使用警告窗口 //测试一些变量 alert(firstname);//其值要么已设置，要么为null，要么为undefined  2.css属性text-shadow的历史很有趣，它曾经被添加到CSS2标准中，但没有浏览器实现它， 然后再CSS2.1标准中又被删掉了。不过，然后浏览器实现了它，现在它已经添加到了CSS3 标准中，目前4个目标浏览器中已经有3个实现了该属性。是否可以通过对象检测来测试浏 览器是否支持这个样式属性？编写一段跨浏览器兼容的代码，对一个已有的header元素设 置这个CSS属性 ----答案---- 首先想到的方法可能是测试是否在style对象中实现了textShadow，这是针对该CSS属性 启用脚本的名称，但其和opacity/alpha示例一样，这不是始终可以依赖的办法。 最简单的方法是直接设置textShadow的值，不管浏览器是否支持，如果浏览器支持它， 则会改变该元素的textShadow值；如果浏览器不支持，那么仍然会动态设置该值并且忽略它： var headerElement=document.getElementById("pageHeader"); headerElement.style.textShadow="#ff0000 2px 2px 3px"; ===========================================================  chapter7 1.编写一段代码，使用DOM Level 0方法为document的click事件指定一个事件处理函数。 ----答案---- document.onclick=clickMe;  2.现在，使用DOM Level 2事件处理机制为document添加click事件处理程序(不用考虑跨浏览器兼容问题)。 ----答案---- document.addEventListener("click",clickMe,false);  3.如何使问题2中所写的代码能够安全的运行在所有浏览器上？ ----答案---- if(document.addEventListener){         document.addEventListener("click",clickMe,false); }else if(document.attachEvent){         document.attachEvent("onclick",clickMe); }  4.对于为document对象指定的onclick事件处理程序，如何知道是在屏幕的什么位置执行了单击操作？ ----答案---- 如果使用DOM Level 0的事件处理系统，那么要么访问window对象的event对象，要么将其作为 参数传递给函数。对于DOM Level 2事件处理模型而言，event对象始终会传递给事件处理函数。 可以通过event对象访问其screenX和screenY属性。  5.使用DOM Level 2事件系统，如何组织从其他元素中冒泡上来的事件？ ----答案---- IE支持的方法和绝大多数浏览器支持的方法不太一样，因此需要分别支持IE和其他浏览器， 可以检查event对象是否支持stopPropagation方法，如果支持，则调用它，否则就将cancelBubble 属性设置为true.  6.将下面这段针对DOM Level 0的事件处理程序改为能够跨浏览器兼容的DOM Level 2方法： <body onload="functionCall();"> ----答案---- if(window.addEventListner){         window.addEventListener("load",functioncall,false); }else if(window.attachEvent){         window.attachEvent("onload",functioncall); }  7.编写一段JavaScript程序代码，捕获document对象的keydown事件，然后通过调用 document.writeln函数输出当前所按的键。 ----答案---- 虽然没有介绍过键盘捕捉事件，但在此处捕获的是keydown事件；然后从该事件的 which属性中就能够获取当前的unicode格式的键码 window.onload=function(){         if(document.addEventListener){           document.addEventListener("keydown",getKey,false);         }else if(document.attachEvent){           document.attachEvent("keydown",getKey);         } } function getKey(event){   var  theEvent=event?evnt:window.event;   document.writeln(theEvent.which); }  ===========================================================  chapter8 1.如果表单数据不完整或者无效，如何停止表单提交操作？ ----答案---- 如果使用DOM Level 0的事件，那么只要从事件处理程序中返回false值，并且事件处理脚本 取消对表单的提交操作即可。如果使用DOM Level 2的事件，那么就将event对象的cancelBubble 属性设置为true(针对IE)，并调用其preventDefault方法(针对其他浏览器)。  2.想在表单提交之前对文本框的内容进行验证，那么应该捕获它的什么事件？ ----答案---- 当一个字段失去焦点时将触发blur事件，这时是检查文本框的值并确保其为有效数据的最佳时机。  3.如果要确保一个字段的内容只包含字符和空格，应该使用什么样的代码？ ----答案---- var fieldPattern=/^[A-Za-z\s]\*$/g; var OK=fieldPattern.exec(document.form[0].text1.value);  4.编写一段JavaScript代码当选中一个单选按钮时捕获触发事件，如果单击其中一个按钮， 则禁用文本框；如果单击另一个按钮，则启用文本框。 ----答案---- 首先，代码必须为每个单选按钮的onclick事件处理程序指定一个事件处理函数： document.forms[0].radiogroup[0].onclick=handleClick; document.forms[0].radiogroup[1].onclick=handleClick; 如果有多个按钮，那么通过一个for循环实现该过程更加简单。在handleClick函数中，检查单选 按钮是否选中，并根据其结果决定是否禁用该表单元素。例如，禁用submit按钮： function  handleClick(){ if(document.forms[0].radiogroup[1].checked){         document.forms[0].submit.disabled=true; }else{         document.forms[0].submit.disabled=false; } } ===========================================================  chapter9 1.如果希望得到一个文本响应，那么应该使用哪种对话框？在程序代码中使用这个对话框，让用户 输入其姓氏。 ----答案---- 如果希望获得一个文本响应应该使用prompt对话框， var firstname=prompt("enter your first name","");  2.定义一个定时器，使其隔3000毫秒调用一个callFunction函数，并且传入两个参数：paramA和paramB。 ----答案---- setTimeout(callFunction,3000,paramA,paramB);  3.如果要检查cookie是否启用，应该使用什么样的程序代码？ ----答案---- 可以通过navigator对象来检查cookie是否启用： if(navigator.cookieEnabled)...  4.创建一个大小为200px\*200px的，不带工具栏和状态栏的新窗口，并在该窗口中打开Google的搜索页面。 ----答案---- var newWindow= window.open("http://www.google.com","","width=200,height=200,toolbar=no,status=no");  5.访问web页面的表单元素有哪些方法？ ----答案---- 如果这个表单元素有标识符，可以使用document.getElementById方法，还可以 通过document对象的forms集合来访问表单元素。 ===========================================================  chapter10 1.说出几种在客户端计算机上存储信息的方法。 ----答案---- 将日期或者其他信息存储到客户端计算机上有以下的几种办法： 使用cookie；使用第三方插件，如flash或Google Gears； 要求用户在一个链接的资源上单击鼠标右键，然后将其保存到本地计算机上； 插入一个作为链接的可下载文件； 创建一个用来保存数据的浏览器扩展； 使用HTML5.0中的localStorage。  2.脚本的cookie由几个部分组成？ ----答案---- 脚本的cookie是由cookie名称，值，日期，与cookie相关联的路径组成的。  3.如何创建在浏览器关闭时自动删除的cookie？ ----答案---- 不提供任何有效日期，并且/或者将值设置为空。  4.对于用户输入的哪类数据需要进行清理？ ----答案---- 任何可以在浏览器上调用的数据，或者能够用来监听客户端cookie的，或者能够 运行服务器端进程的，在用户输入过程中都将进行清理。特别是"javascript:" 或脚本标签都会从输入中清理。但对输入清理并不完全是直接删除。对于管理工具而言， 用户在一个特定的帖子或者页面中输入脚本是可行的。但在一个多用户环境中，某个用户 就可能通过脚本获得使用该系统的其他用户信息。  5.考虑你已经创建或未来将创建的网站。请构思脚本cookie在你网站中可能实现的5个不同用途。 那么在这些应用场景中，你需要的空间是否比cookie所能提供的更大？ ----答案---- 维护某个人的名字，URL和评论系统的电子邮件； 为数据项提供实时反馈； 启用拼写检查； 保存登陆信息； 维护购物车； 以上这些数据所需的存储空间不超过4KB。 ===========================================================  chapter11 1.所有HTML元素都支持的属性有哪些？ ----答案---- 这样的属性包括id,title,lang,dir,className  2.当给定一个已命名的元素时，如何使用HTML DOM找到其元素类型？ ----答案---- 访问该元素的tagName属性。  3.给定一个位于核心DOM中的节点，如何输出其每个子节点的元素类型？ ----答案---- var children=targetNode.childNodes; for(var i=0;i<children.length;i++){         alert(children.nodeType); }  4.已知页面中的所有div元素，如何找出这些div元素的ID（标识符）？ ----答案---- var divs=document.getElementByTagName('div'); for(var i=0;i<divs.length;i++){ alert(divs.id); }  5.对于下面这个元素，如果要访问它有哪三种不同类型的方法？ <div id="elem1" class="thediv">...</div> ----答案---- var theDiv=docuemnt.getElementById("elem1");  var theDivs=document.getElementByTagName("div")； var theDiv=theDivs[0];   var theDivs=document.getElementByClassName("thediv")； var theDiv=theDivs[0];  6.除了使用innerHTML，如何将下面这个div元素中的header元素换成一个段落： <div id="elem1"> <h1>this is a header</h1> </div> ----答案---- var targetElement=document.getElementById('elem1'); var specChild=targetElement.getElementByTagName('h1')[0]; var newPara=document.createElement('p'); var paraTxt=document.createTextNode('hello'); newPara.appendChild(paraTxt);  targetElement.replaceChild(newPara,specChild); =========================================================== chapter12 1.当在JavaScript程序中尝试使用obj.style.color方法访问一个元素的文本颜色时，却没有任何返回值。 你知道这个字体颜色是通过样式表设置的。为什么没有任何返回值？要如何修改程序才能够获取该值？ 写一段代码，访问名为“name”的元素的字体颜色，以演示你的方法。 ----答案---- 可以使用getComputedStyle方法或currentStyle方法，这取决于使用的是哪种浏览器。另外一种方法是在 页面载入时设置style属性。 var nameDiv=document.getElementById("name"); var nameColor; if(nameDiv.currentStyle){ nameColor=nameDiv.currentStyle['color'];         }else if(window.getComputedStyle){ nameColor= document.defaultView.getComputedStyle(nameDiv,null).getPropotypeVaule('color'); }  2.对于div中的文本，如何将其显示的样式改成14px的字体，红色，行高16px？ ----答案---- divElement.style.font="14px/16px"; divElement.style.color="#ff0000";  3.如果之前的修改未生效，可能是什么原因导致的？ ----答案---- 如果这个div元素中的文本实际上是属于另一个元素的，如一个段落元素，并且它有不同的样式设置， 那么段落元素的样式设置将覆盖外层的div元素的新样式设置。  4.让一个HTML元素块消失有哪三种办法？写一段代码，将名为“name”的元素彻底从页面布局中删除掉， 以演示方法。 ----答案---- 要让一个div元素消失，可以将其width或height属性设置为0，或者将整个元素完全修剪掉。还可以将 该元素的visibility属性设置为hidden，或者将其display属性设置为none。最后还可以将该元素移到 页面外，移动到最顶部或者最左边。如果要从页面布局中删除该元素，需要修改display属性： var nameDiv=document.getElementById("name"); name.style.display="none";  5.如果拖放技术不是有效的购物车技术，那么对于这种类型的服务，可以使用哪种动态web页面效果？ ----答案---- 除了使用拖放功能，还可以为显示"Buy me!"信息的链接或图像添加一个鼠标单击事件处理程序，当 触发这个单击事件处理程序时，在应用程序中添加一些代码，以便自动将所选商品添加到购物车中。 =========================================================== chapter13 1.如果要为Number对象创建一个名为triple的新办法，用来求当前Number对象的数值的3倍，而且希望 该方法能够适用于所有数字，那么要怎么实现呢？ ----答案---- 使用Number的prototype属性： Number.prototype.triple=function(){   var nmToTriple=this.valueOf()\*3;   return nmToTriple; } var newNum=new Number(3.0)； alert(newNum.triple());  2.如何隐藏新对象的数据成员？为什么要这样做？ ----答案---- 通过var关键字声明数据成员，而不是把数据成员赋予this。数据隐藏的目的是控制如何访问或者修改数据。  3.创建一个参数为数字类型的函数，并且当参数类型错误时返回错误。如果不使用return语句，应该 实现该功能？ ----答案---- 使用throw语句触发错误，然后在主调程序中通过try...catch语句捕捉异常。 if(typeof value !="number"){         throw "NotANumber"; }  4.在之前的事例中可以用于事件的对象检测代码 var theEvent=nsEvent?nsEvent:window.event; 为什么在处理透明度差异的时候不能使用相同的功能呢？ ----答案---- 与event对象不一样，处理透明度的时候不只会涉及到业务模型上的差异。不仅属性不同，而且与 赋值属性的可能值有关。  5.创建一个自定义对象，使其拥有3个公有办法，changeState,getColor和getState;以及两个私有 数据成员:background和state。将state属性设置为on，将background颜色属性设置为#fff。 changeState方法同时用来判断state是否为on，如果state为on，那么将state改为off，并同时将 颜色设置为#000。getColor方法将返回背景颜色，getState负责返回状态。 ----答案---- function Control(){   var state='on';   var background='#fff';      this.changeState=function(){         if(state=='on'){           state='off';           background='#000';         }else {           state='on';           background='#fff';         }   };   this.getState=function(){      return state;   };   this.getColor=function(){      return background;   }; } =========================================================== chapter14 1.XMLHttpRequest请求也可以设置为同步模式（也就是停下来等待响应），虽然这看起来和Ajax 的概念有些背道而驰。但应该如何打开这样的请求呢？ ----答案---- XMLHttpRequest.open函数的第三个，可选的参数是一个布尔值。将该参数设置为true(默认值) 将创建异步请求；如果将其设置为false则将创建同步请求。  2.当请求接受到响应时，就需要对响应进行处理。那么如何指定在服务器响应时调用的函数呢？ ----答案---- 当获得XMLHttpRequest对象的引用并调用其open方法之后，将赋值给onReadyStateChange属性。  3.对于Ajax请求而言，表示请求成功需要对哪两个状态进行判断？表示请求完成呢？ ----答案---- XMLHttpRequest对象的readyState属性值为4就表示请求完成，要表明服务请求是成功的， 还必须确保其status属性值为200。 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |