

- ★ 1. 请把以下用于连接字符串的JavaScript代码修改为更高效的方式：

```
var htmlString = '<div class="container">' + '<ul id="news-list">';
for(var i = 0; i < NEWS.length; i++) {
    htmlString += '<li><a href="' + NEWS[i].LINK + '">' + NEWS[i].TITLE + '</a></li>';
}
htmlString += '</ul></div>';
```

参考答案：

```
var arr = [];
arr.push('<div class="container"><ul id="news-list">');
for(var i = 0; i < NEWS.length; i++) {
    arr.push('<li><a href="' + NEWS[i].LINK + '">' + NEWS[i].TITLE + '</a></li>');
}
arr.push('</ul></div>');
var str = arr.join("");
```

- ★ 2. 尝试实现注释部分的Javascript代码，可在其他任何地方添加更多代码（如不能实现，说明一下不能实现的原因）：

```
var Obj = function(msg) {
    this.msg = msg;
    this.shout = function() {
        alert(this.msg);
    }
    this.waitAndShout = function() {
        //隔五秒钟后执行上面的shout方法
    }
}
```

参考答案：

```
var Obj = function(msg) {
    this.msg = msg;
    this.shout = function() {
        alert(this.msg);
    }
    this.waitAndShout = function() {
        //隔五秒钟后执行上面的shout方法
        var o = this;
        setTimeout(function() {
            o.shout();
        }, 5000);
    }
}
```

注：这个问题考的是闭包！

- ★ 3. 请编写一个JavaScript 函数toRGB，它的作用是转换CSS中常用的颜色编码。要求：

```
alert(toRGB("#0000FF")); // 输出 rgb(0, 0, 255)
alert(toRGB("invalid")); // 输出 invalid
alert(toRGB("#G00")); // 输出 #G00
```

参考答案：

注：该方法网上搜集，个人觉得考虑不全面

```
function toRGB(color) {
    var hh = "";
    var kk = [];
    var reg = /^#([0-9a-zA-Z]{3})$/;
    if(color == "invalid")
        return "invalid";
    if(reg.test(color))
        return color;
    for(var i = 1; i < color.length; i += 2) {
        hh = color.substr(i, 2);
        var cc = parseInt(hh, 16);
        kk.push(cc);
    }
    return ("RGB(" + kk + ")");
}
```

注：该方法网上搜集，测试通过

```
function toRGB(val) {
    var reg1 = /^#([0-9A-F]{2})([0-9A-F]{2})([0-9A-F]{2})$/i,
        reg2 = /^#([0-9A-F])([0-9A-F])([0-9A-F])$/i,
        reg3 = /^[0-9A-F]{2}/g, m;
    if(reg2.test(val)) {
        val = val.replace(reg2, "#$1$1$2$2$3$3");
    }
    if(reg1.test(val)) {
        m = val.match(reg3);
        val = "rgb(" + [parseInt(m[0], 16), parseInt(m[1], 16), parseInt(m[2], 16)].join(", ") + ")";
    }
    return val;
}
```

```
function toRGB(sColor) {
    var re = /^#([0-9a-f]{3}|[0-9a-f]{6})$/ig;
    if(re.test(sColor)) {
        var arr = [];
        if(sColor.length == 7) {
```

```

    arr = sColor.replace("#", "").match(/[0-9a-f]{2}/ig);
  } else if(sColor.length == 4) {
    arr = sColor.replace("#", "").match(/[0-9a-f]{1}/ig);
  }
  return "RGB(" + parseInt(arr[0], 16) + "," + parseInt(arr[1], 16) + "," + parseInt(arr[2], 16) + ")";
} else {
  return "invalid";
}
}
}

```

考点：

1. 对于函数的理解
2. 正则表达式

★ 4. 请说明下面各种情况的执行结果，并注明产生对应结果的理由。

```

function doSomething() {
  alert(this);
}

```

- ① element.onclick = doSomething，点击element元素后。
- ② element.onclick = function() {doSomething()}，点击element元素后。
- ③ 直接执行doSomething()。

参考答案：

- 1、element，调用执行时直接指向事件元素。
- 2、window，函数调用中的 this 绑定到全局对象。
- 3、window，函数调用中的this绑定到全局对象。

注：考点是函数中this指向的对象，这个题目在实际工作中经常会遇到，非常重要！

★ 5. 阅读以下JavaScript代码：

```

if(window.addEventListener) {
  var addListener = function(el, type, listener, useCapture) {
    el.addEventListener(type, listener, useCapture);
  };
} else if(document.all) {
  addListener = function(el, type, listener) {
    el.attachEvent("on" + type, function() {
      listener.apply(el);
    });
  };
}

```

请阐述

- a) 代码的功能；
- b) 代码的优点和缺点；
- c) listener.apply(el) 在此处的作用；
- d) 如果有可改进之处，请给出改进后的代码，并说明理由。

参考答案：

- a) 功能：事件注册
- b) 优点：跨浏览器，特性探测，性能优化。缺点：document.all
- c) 作用：使得IE中listener的this为el，与其它浏览器一致
- d) 改进：document.all改成window.attachEvent; useCapture的默认值

注：考点是

- a. 动态绑定事件
- b. 能力检测
- c. apply用法

★ 6. 请编写一个JavaScript函数，它的作用是校验输入的字符串是否是一个有效的电子邮件地址。要求：

- a) 使用正则表达式。
- b) 如果有效返回true，反之为false。

参考答案：

```

var checkEmail = function(email) {
  var preg = "([a-zA-Z]|^[\w- \.]*[a-zA-Z0-9])@(\w+|\.)+\w+$";
  pregObj = new RegExp(preg);
  return pregObj.test(email);
}

```

★ 7. 请分别列出HTML、JavaScript、CSS、Java、php、python的注释代码形式。

参考答案：

```

Js：
单行 // 注释内容
多行
/*
  注释内容
*/

Css：/* 注释内容 */

Html：<!--注释的内容-->

PHP：//单行注释内容
/*
  多行注释内容
*/

```

```
Python :
#单行注释内容
```

- ★ 8. 请编写一段JavaScript脚本生成下面这段DOM结构。要求：使用标准的DOM方法或属性。（淘宝）

```
<div id=" example" >
    <p class=" slogan" >淘！你喜欢</p>
</div>
```

参考答案：

```
var oDiv = document.createElement('div');
oDiv.setAttribute('id', 'example');
var oP = document.createElement('p');
oDiv.appendChild(oP);
var oText = document.createTextNode('淘！你喜欢');
oP.appendChild(oText);
document.body.appendChild(oDiv);
```

评分要点：

1. 创建元素的DOM方法 - createElement;
 2. 设置元素属性的DOM方法 - setAttribute;
 3. 添加到元素上的DOM方法 - appendChild;
 4. 创建文字的DOM方法 - createTextNode;
 5. 设置样式的DOM属性 - className (HTML DOM属性)
- 或者：
- ```
if (isIE) {
 oDiv.setAttribute("className", "box");
} else {
 oDiv.setAttribute("class", "box");
}
```

- ★ 9. 对于下面这段代码：

```

```

问题一：请列举其优缺点，说说你的看法。

问题二：请编写一个通用的事件注册函数（请看下面的代码）。

```
function addEvent(element, type, handler) {
 // 在此输入你的代码，实现预定功能
}
```

参考答案：

问题一：

优点：简单易懂

缺点：按照表现与业务分享的原则，不应该将onclick事件写在html代码中，应该提取出来。

问题二：

```
function addEvent(element, type, handler) {
 if (element.addEventListener) {
 element.addEventListener(type, handler, false);
 } else if (element.attachEvent) { //for IE
 element.attachEvent('on' + type, handler);
 } else {
 element['on' + type] = handler;
 }
}
```

- ★ 10. 下面是个输入框，当没有获取焦点时，显示灰色的提示信息：

请输入内容

当用户输入时，隐藏提示文字，且恢复为默认色：

|

当输入框失去焦点，如果输入为空，需还原提示信息：

请输入内容

要求：

- a) 写出HTML和CSS代码
- b) 用JavaScript实现功能

参考答案：

```
<style type="text/css">
 .tip { color: gray; }
</style>

<input type="text" id="test" class="tip" value="请输入内容" />

<script type="text/javascript">
 (function() {
 var test = document.getElementById("test"), REG = /^(^|\s+)tip(\s+|$)/, TIP_CLS =
 "tip", TIP_STR = "请输入内容";
 test.addEventListener("focus", function() {
 var className = this.className;
 if (className.match(REG)) {
 this.value = "";
 this.className = className.replace(TIP_CLS, "");
 }
 }, false);
 test.addEventListener("blur", function() {
 if (this.value.length == 0) {
 this.value = TIP_STR;
 this.className += " " + TIP_CLS;
 }
 }, false);
 })();
</script>
```

```
})();
</script>
```

阅卷说明:

合格: HTML和CSS代码正确, JS功能正确

良好: 事件注册采用addEventListener或attachEvent; 样式和行为分离, JS代码中通过class改变样式

优秀: 事件注册兼容主流浏览器; 根据className而不是value值来判断切换; 封装成通用util

- ★11. 请给JavaScript的String 原生对象添加一个名为trim 的原型方法, 用于截取空白字符。要求: ( 淘宝 )

```
alert(" taobao".trim()); // 输出 "taobao"
alert(" taobao ".trim()); // 输出 "taobao"
```

参考答案:

```
String.prototype.trim = function() {
 return this.replace(/^\s+|\s+$/g, "");
};
```

- ★12. 请编写一个JavaScript函数 parseQueryString, 它的用途是把URL参数解析为一个对象, 如:

```
var url = "http://www.taobao.com/index.php?key0=0&key1=1&key2=2....."
var obj = parseQueryString(url);
alert(obj.key0) // 输出0
```

参考答案:

注: 该代码网上搜集, 未测试

```
function parseQueryString(name) {
 name = name.replace(/[\[\]]/, "\\[\\]").replace(/[\]\]/, "\\]");
 var regexS = "[\\?&]" + name + "=[^&#]*";
 var regex = new RegExp(regexS);
 var results = regex.exec(window.Location.href);
 if(results == null)
 return "";
 else
 return results[1];
}
```

另一段参考代码, 未测试

```
function parseQueryString(url) {
 var params = {};
 var arr = url.split("?");
 if(arr.Length <= 1)
 return params;
 arr = arr[1].split("&");
 for(var i = 0, L = arr.Length; i < L; i++) {
 var a = arr[i].split("=");
 params[a[0]] = a[1];
 }
 return params;
}

function parseUrl(url) { //http://www.w3school.com.cn:8080/html/a/b/Lastpage.html?key1=1&key2=2
 var re = /^((http|https|ftp|mailto|telnet|irc):\/\/([a-zA-Z0-9.-]*)(:([0-9]{1,5}))?)?(\/([a-zA-Z0-9.-]*)?)?$/ig;
 //alert(re.exec(url));
 var result = re.exec(url);
 var strArr = parseQueryString(result[6]);
 var obj = {
 protocol : result[1],
 host : result[2],
 port : result[3],
 path : result[4],
 filename : result[5]
 };
 for(var i = 0; i < strArr.Length; i++) {
 obj[strArr[i][0]] = strArr[i][1];
 }
 return obj;
}

function parseQueryString(str) {
 var strArr = [];
 var arr = str.split("&");
 for(var i = 0; i < arr.Length; i++) {
 var arr1 = arr[i].split("=");
 strArr.push(arr1);
 }
 return strArr;
}

//测试代码
var url = parseUrl("http://www.w3school.com.cn:8080/html/a/b/c/Lastpage.html?key1=1&key2=2");
for(key in url) {
 alert(key + ":" + url[key]);
}
```

- ★13. 请给Array本地对象增加一个原型方法, 它用于删除数组条目中重复的条目(可能有多条), 返回值是一个包含被删除的重复条目的新数组。

参考答案:

```
Array.prototype.distinct = function() {
 var ret = [];
 for(var i = 0; i < this.Length; i++) {
 for(var j = i + 1; j < this.Length;) {
 if(this[i] === this[j]) {
```

```

 ret.push(this.splice(j, 1)[0]);
 } else {
 j++;
 }
}
}
return ret;
}
}
}
}
return this;
};
//for test
alert(['a', 'b', 'c', 'd', 'b', 'a', 'e'].distinct());

```

另一段参考代码：

```

Array.prototype.distinct = function() {
 for(var i = 0; i < this.length; i++) {
 for(var j = i + 1; j < this.length;) {
 if(this[i] == this[j]) {
 //result.push(this[i]);
 this.splice(j, 1);
 } else {
 j++;
 }
 }
 }
 return this;
};
//for test
alert(['a', 'b', 'c', 'd', 'b', 'a', 'e'].distinct());

```

★14. 请根据下面的描述，用JSON语法编写一个对象:

“小明今年22岁，来自杭州。兴趣是看电影和旅游。他有两个姐姐，一个叫小芬，今年25岁，职业是护士。还有一个叫小芳，今年23岁，是一名小学老师。”

var person = ?

参考答案：

```

var person = {
 "name" : "小明",
 "age" : 22,
 "city" : "杭州",
 "interest" : ["电影", "旅游"],
 "sisters" : [{
 "name" : "小芬",
 "age" : 25,
 "job" : "护士"
 }, {
 "name" : "小芳",
 "age" : 23,
 "job" : "小学老师"
 }]
};

```

★15. 请改善以下HTML代码，使其符合xHTML 1.0规范：

```



```

参考答案：

```



```

1. 标签小写（包括onclick）
2. 属性名要使用引号(单双不限)
3. 必须闭合标签

★16. 看看下面alert的结果是什么

```

function b(x, y, a) {
 arguments[2] = 10;
 alert(a);
}
b(1, 2, 3);
如果函数体改成下面，结果又会是什么？
a = 10;
alert(arguments[2]);

```

参考答案：

10
10

★17. 什么是闭包？下面这个ul，如何点击每一列的时候alert其index？

```

<ul id='test'>
 这是第一条
 这是第二条
 这是第三条


```

参考答案：

闭包的概念：比较学术的解释，闭包（Closure）是词法闭包（Lexical Closure）的简称，是引用了自由变量的函数。这个被引用的自由变量将和这个函数一同存在，即使已经离开了创造它的环境也不例外。所以，有另一种说法认为闭包是由函数和与其相关的引用环境组合而成的实体。**简单的解释就是**：闭包的作用就是在函数a执行完并返回后，闭包使得Javascript的垃圾回收机制GC不会收回函数a所占用的资源，因为函数a的内部函数b的执行需要依赖函数a中的变量。

```

window.onload = function() {
 var oUl = document.getElementById('ul');
 var ali = oUl.getElementsByTagName('li');
 for(var i = 0; i < ali.length; i++) {(function(index) {
 ali[i].onclick = function() {
 alert(index);
 };
 })(i);
 }
};

```

注：闭包很重要！

★18. Ajax是什么？Ajax的交互模型？同步和异步的区别？如何解决跨域问题？

参考答案：

Ajax是多种技术组合起来的一种浏览器和服务端交互技术，基本思想是允许一个互联网浏览器向一个远程页面/服务做异步的http调用，并且用收到的数据来更新一个当前web页面而不必刷新整个页面。该技术能够改进客户端的体验。

包含的技术：

XHTML：对应W3C的XHTML规范，目前是XHTML1.0。

CSS：对应W3C的CSS规范，目前是CSS2.0

DOM：这里的DOM主要是指HTML DOM，XML DOM包括在下面的XML中

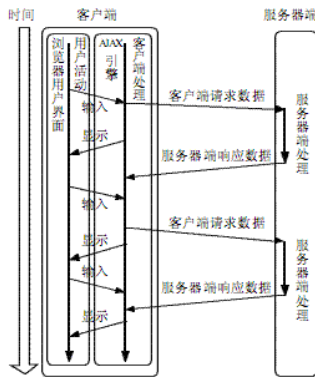
JavaScript：对应于ECMA的ECMAScript规范

XML：对应W3C的XML DOM、XSLT、XPath等等规范

JSON：JavaScript Object Notation

★XMLHttpRequest：对应WhatWG的Web Applications1.0规范（<http://whatwg.org/specs/web-apps/current-work/>）

**AJAX交互模型：**



(b) 基于 AJAX 的异步传输原理

**同步：**脚本会停留并等待服务器发送回复然后再继续

**异步：**脚本允许页面继续其进程并处理可能的回复

前端对于跨域的解决办法：

(1) 如果是主域与子域之间使用：document.domain

(2) 如果是不同主域之间利用script标签的src属性没有跨域限制

★19. 请给出异步加载js方案，不少于两种

参考答案：

默认情况javascript是同步加载的，也就是javascript的加载时阻塞的，后面的元素要等待javascript加载完毕后才能进行再加载，对于一些意义不是很大的javascript，如果放在页头会导致加载很慢的话，是会严重影响用户体验的。

异步加载方式：

(1) defer，早期的时候只有IE支持，但是对于最新的其它浏览器，也支持了

(2) async：

(3) 创建script，插入到DOM中，加载完毕后callBack，见代码：

```

function LoadScript(url, callback) {
 var script = document.createElement("script");
 script.type = "text/javascript";
 if(script.readyState) { //IE
 script.onreadystatechange = function() {
 if(script.readyState == "Loaded" || script.readyState == "complete") {
 script.onreadystatechange = null;
 callback();
 }
 };
 } else { //Others: Firefox, Safari, Chrome, and Opera
 script.onload = function() {
 callback();
 };
 }
 script.src = url;
 document.body.appendChild(script);
}

```

★20. 请设计一套方案，用于确保页面中JS加载完全。

参考答案：

```

var n = document.createElement("script");
n.type = "text/javascript";
//以上省略部分代码
//ie支持script的readystatechange属性(IE support the readystatechange event for script and css nodes)

```

```

if(ua.ie) {
 n.onreadystatechange = function() {
 var rs = this.readyState;
 if('loaded' === rs || 'complete' === rs) {
 n.onreadystatechange = null;
 f(id, url);
 //回调函数
 }
 };
 //省略部分代码
 //safari 3.x supports the Load event for script nodes(DOM2)
 n.addEventListener('Load', function() {
 f(id, url);
 });
 //firefox and opera support onLoad(but not dom2 in ff) handlers for
 //script nodes. opera, but no ff, support the onLoad event for link
 //nodes.
} else {
 n.onload = function() {
 f(id, url);
 };
}
}

```

★ 21. js中如何定义class,如何扩展prototype?

参考答案：

```

Ele.className = "****" ; //****在css中定义，形式如下：.*** {...}
A.prototype.B = C;
A是某个构造函数的名字
B是这个构造函数的属性
C是想要定义的属性的值
如何添加html元素的事件,有几种方法.
(1) 为HTML元素的事件属性赋值
(2) 在JS中使用ele.on*** = function() {...}
(3) 使用DOM2的添加事件的方法 addEventListener或attachEvent

```

★ 22. document.write和 innerHTML的区别

参考答案：

```

document.write只能重绘整个页面
innerHTML可以重绘页面的一部分

```

★ 23. js的基础对象有那些, window和document的常用的方法和属性列出来

参考答案：

```

String,Number,Boolean,Date,Math,Array,RegExp
Window:
方法：setInterval,setTimeout,clearInterval,clearTimeout,alert,confirm,open
属性：name,parent,screenLeft,screenTop,self,top,status
Document：
方法：createElement,execCommand,getElementById,getElementsByTagName,write,writeln
属性：cookie,doctype,domain,documentElement,readyState,URL

```

★ 24. 写一段js，判断是否是这样组成的：第一个必须是字母，后面可以是字母、数字、下划线，总长度为5-20

参考答案：

```

var reg = /^[a-zA-Z][a-zA-Z_0-9]{4,19}$/;
reg.test("a1a_a1a_a1a_a1a_");

```

★ 25. 写一段js，截取字符串abcdefg中的efg

参考答案：

```

var str = "abcdefg";
if(/efg/.test(str)) {
 var efg = str.substr(str.indexOf("efg"), 3);
 alert(efg);
}

```

★ 26. 写一段js，判断一个字符串中出现次数最多的字符，统计这个次数

参考答案：

```

//将字符串的字符保存在一个hash table中，key是字符，value是这个字符出现的次数
var str = "abcdefgadda";
var obj = {};
for(var i = 0; i < str.length; i++) {
 var key = str[i];
 if(!obj[key]) {
 obj[key] = 1;
 } else {
 obj[key]++;
 }
}
/*遍历这个hash table，获取value最大的key和value*/
var max = -1;
var max_key = "";

```

```

var key;
for(key in obj) {
 if(max < obj[key]) {
 max = obj[key];
 max_key = key;
 }
}

alert("max:" + max + " max_key:" + max_key);

```

★27. 如何规避javascript多人开发函数重名问题

参考答案：

- (1) 可以开发前规定命名规范，根据不同开发人员开发的功能在函数前加前缀
- (2) 将每个开发人员的函数封装到类中，调用的时候就调用类的函数，即使函数重名只要类名不重复就ok

28. javascript面向对象中继承实现

参考答案：

javascript面向对象中的继承实现一般都使用到了构造函数和Prototype原型链，简单的代码如下：

```

function Animal(name) {
 this.name = name;
}

Animal.prototype.getName = function() {
 alert(this.name)
}

function Dog() {};

Dog.prototype = new Animal("Buddy");
Dog.prototype.constructor = Dog;
var dog = new Dog();

```

29. 编写一个方法 求一个字符串的字节长度

参考答案：

假设：  
一个英文字符占用一个字节，一个中文字符占用两个字节

```

function GetBytes(str) {
 var len = str.length;
 var bytes = len;
 for(var i = 0; i < len; i++) {
 if(str.charCodeAt(i) > 255)
 bytes++;
 }
 return bytes;
}

alert(GetBytes("你好,as"));

```

30. 编写一个方法 去掉一个数组的重复元素

参考答案：

```

var arr = [1, 1, 2, 3, 3, 2, 1];
Array.prototype.unique = function() {
 var ret = [];
 var o = {};
 var len = this.length;
 for(var i = 0; i < len; i++) {
 var v = this[i];
 if(!o[v]) {
 o[v] = 1;
 ret.push(v);
 }
 }
 return ret;
};
alert(arr.unique());

```

31. 写出3个使用this的典型应用

参考答案：

- (1) 在html元素事件属性中使用，如  
- (2) 构造函数  

```

function Animal(name, color) {
 this.name = name;
 this.color = color;
}

```
- (3)  

```

<input type="button" id="text" value="点击一下" />
<script type="text/javascript">
 var btn = document.getElementById("text");
 btn.onclick = function() {
 alert(this.value); //此处的this是按钮元素
 }

```



```

 }
</script>
(4) CSS expression表达式中使用this关键字
<table width="100px" height="100px">
<tr>
<td>
<div style="width:expression(this.parentNode.width);">div element</div>
</td>
</tr>
</table>

```

32. JavaScript中如何检测一个变量是一个String类型？请写出函数实现

参考答案：

```

String类型有两种生成方式：
(1)var str = 'hello world';
(2)var str2 = new String('hello world');
function isString(str) {
 return (typeof str == "string" || str.constructor == String);
}

var str = "";
alert(IsString(1));
alert(IsString(str));
alert(IsString(new String(str)));

```

33. 网页中实现一个计算当年还剩多少时间的倒计时程序，要求网页上实时动态显示“××年还剩××天××时××分××秒”

参考答案：

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>倒计时</title>
</head>
<body>
<input type="text" value="" id="input" size="1000"/>
<script type="text/javascript">
 function counter() {
 var date = new Date();
 var year = date.getFullYear();
 var date2 = new Date(year, 12, 31, 23, 59, 59);
 var time = (date2 - date) / 1000;
 var day = Math.floor(time / (24 * 60 * 60));
 var hour = Math.floor(time % (24 * 60 * 60) / (60 * 60));
 var minute = Math.floor(time % (24 * 60 * 60) % (60 * 60) / 60);
 var second = Math.floor(time % (24 * 60 * 60) % (60 * 60) % 60);
 var str = year + "年还剩" + day + "天" + hour + "时" + minute + "分" + second + "秒";
 document.getElementById("input").value = str;
 }
 window.setInterval("counter()", 1000);
</script>
</body>
</html>

```

34. JavaScript有哪几种数据类型

参考答案：

简单：Number，Boolean，String，Null，Undefined  
引用：Object，Array，Function

35. 请实现，鼠标点击页面中的任意标签，alert该标签的名称。（注意兼容性）

参考答案：

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>鼠标点击页面中的任意标签，alert该标签的名称</title>
<style>
div {
 background: #0000FF;
 width: 100px;
 height: 100px;
}
span {
 background: #00FF00;
 width: 100px;
 height: 100px;
}
p {
 background: #FF0000;
 width: 100px;
 height: 100px;
}

```

```

</style>
<script type="text/javascript">
 document.onclick = function(evt) {
 var e = window.event || evt;
 var tag = e["target"] || e["srcElement"];
 alert(tag.tagName);
 };
</script>
</head>
<body>
 <div id="div">
 SPANDIV
 </div>
 SPAN
 <p>
 P
 </p>
</body>
</html>

```

36. <img>元素的alt和title有什么异同？(百度)

参考答案：

这个我前几天正好看到了一篇这样的文章。这两个属性是有些重复了。在不同浏览器里面表现有些不同。在alt和title同时设置的时候，alt作为图片的替代文字出现，title是图片的解释文字。

37. 编写一个JavaScript函数，实时显示当前时间，格式“年-月-日 时:分:秒”。(百度)

参考答案：

```

window.onload = function() {
 /* 显示当前时间 */
 function print_date() {
 var now = new Date();
 document.getElementById('time').innerHTML = now.getFullYear() + "年" + (now.getMonth() + 1)
 + "月" + now.getDate() + "日" + " 星期" + ['日', '一', '二', '三', '四', '五', '六',]
 [now.getDay()] + " " + now.getHours() + ":" + now.getMinutes() + ":" + now.getSeconds();
 }

 window.setInterval(print_date, 1000);
}

<div id="time"></div>

```

38. 使用JavaScript深度克隆一个对象。(百度)

参考答案：

Javascript中的对象赋值与Java中是一样的，都为引用传递。就是说，在把一个对象赋值给一个变量时，那么这个变量所指向的仍就是原来对象的地址。那怎么做呢？答案是“克隆”。

克隆有两种方法：一种是“浅克隆”，一种是“深克隆”（深度克隆）。

浅克隆：基本类型为值传递，对象仍为引用传递。

深克隆（深度克隆）：所有元素均完全复制，并于原对象完全独立（原对象的修改不影响新对象）。

深度克隆的代码如下（深克隆去掉递归即为浅克隆）：

```

Object.prototype.cloneObj = function() { // 该方法不能处理数组
 function NewObject() {};
 NewObject.prototype = this;
 var newObj = new NewObject();
 for(var ele in newObj) {
 if(typeof newObj[ele] == 'object') {
 newObj[ele] = newObj[ele].cloneObj();
 }
 }
 return newObj;
}

obj1 = {
 name : "johnny",
 age : 34,
 address : {
 province : 'heilongjiang',
 city : 'harbin'
 }
};
obj2 = obj1.cloneObj();
obj2.address.city = 'beijing';
alert(obj1.address.city);

```

另外一个参考答案：

```

Object.prototype.clone = function() {
 var o = (this.constructor === Array ? [] : {});
 for(var e in this) {
 o[e] = typeof this[e] === "object" ? this[e].clone() : this[e];
 }
 return o;
}

使用示例：
obj1 = {
 name : "johnny",

```

```

 age : 34,
 address : {
 province : 'heilongjiang',
 city : 'harbin'
 }
 };
obj2 = obj1.clone();
obj2.address.city = 'beijing';
alert(obj1.address.city);

```

39. A.prototype.name的含义是什么，用来解决什么问题？

参考答案：

利用原型prototype对“类A”进行属性和方法的扩展，详细信息参考w3school JavaScript高级教程：  
[http://www.w3school.com.cn/js/pro\\_js\\_object\\_defining.asp](http://www.w3school.com.cn/js/pro_js_object_defining.asp)

40. 运行test()和new test()的结果是什么？

```

var a = 5;
function test() {
 a = 0;
 alert(a);
 alert(this.a);
 var a;
 alert(a);
}

```

参考答案：

```

test(): 0 5 0
new test(): 0 undefined 0

```

41. 设计一个下拉框，既可以选中已有的列表项，又可以自己手动输入。

参考答案：

44. 写出运算结果

```

alert(typeof(null))
alert(typeof(undefined))
alert(typeof(NaN))
alert(NaN==undefined)
alert(NaN==NaN)
var str="123abc";alert(typeof(str++))
alert(str)

```

45. 参考答案：

```

alert(typeof(null)) -> object
alert(typeof(undefined)) -> undefined
alert(typeof(NaN)) -> number
alert(NaN==undefined) -> false
alert(NaN==NaN) -> false
var str="123abc";alert(typeof(str++)) -> number
alert(str) -> NaN

```

46. div宽position:absolute;top:0;left:0;从左到右匀速运动到300px位置；

参考答案：

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
 <head>
 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
 <title>匀速运动</title>
 <style>
 #div1{width:100px;height:100px;background:red;position:absolute;Left:0;}
 span{width:1px;height:300px;background:black;position:absolute;Left:300px;}
 </style>
 <script type="text/javascript">
 var timer = null;
 function doMove() {
 var oDiv1 = document.getElementById('div1');
 clearInterval(timer);
 timer = setInterval(function() {
 var iSpeed = 9;
 //匀速运动终止条件，当物体的Left值与目标点距离足够近时
 if(Math.abs((300 - oDiv1.offsetLeft)) < Math.abs(iSpeed)) {
 clearInterval(timer);
 oDiv1.style.Left = 300 + 'px';
 } else {
 oDiv1.style.Left = oDiv1.offsetLeft + iSpeed + 'px';
 }
 }, 30);
 }
 </script>
 </head>
 <body>
 <input type="button" value="开始移动" onClick="doMove()"/>
 <div id="div1"></div>

 </body>
</html>

```

## 47. 省市二级联动

参考答案：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
 <head>
 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
 <title>省市二级联动</title>
 <script type="text/javascript">
 function Province(name) { // 省类
 this.name = name;
 this.cities = [];
 }

 function City(name, province) { // 市类
 this.name = name;
 this.province = province;
 }

 var beijing = new Province("北京");
 var shandong = new Province("山东");
 var shanghai = new Province("上海");

 var c1 = new City("朝阳区", beijing);
 var c2 = new City("海淀区", beijing);
 var c3 = new City("东城区", beijing);
 var c4 = new City("西城区", beijing);

 var c5 = new City("济南区", shandong);
 var c6 = new City("青岛区", shandong);
 var c7 = new City("潍坊区", shandong);

 var c8 = new City("浦东区", shanghai);
 var c9 = new City("徐汇区", shanghai);
 var c10 = new City("虹桥区", shanghai);

 beijing.cities.push(c1);
 beijing.cities.push(c2);
 beijing.cities.push(c3);
 beijing.cities.push(c4);

 shandong.cities.push(c5);
 shandong.cities.push(c6);
 shandong.cities.push(c7);

 shanghai.cities.push(c8);
 shanghai.cities.push(c9);
 shanghai.cities.push(c10);

 var provinces = [beijing, shandong, shanghai];

 window.onload = function() {
 var oProvince = document.getElementById('province');
 var oCity = document.getElementById('city');
 for(var i = 0; i < provinces.length; i++) {
 var oOpt = new Option(provinces[i].name);
 oProvince.add(oOpt);
 }

 oProvince.onchange = function() {
 for(var i = 0; i < provinces.length; i++) {
 if(this.options[this.selectedIndex].text == provinces[i].name) {
 for(var k = 0; k < oCity.options.length; k++) {
 oCity.remove(k);
 k--;
 }
 for(var j = 0; j < provinces[i].cities.length; j++) {
 var city = provinces[i].cities[j];
 var oOpt = new Option(city.name);
 oCity.add(oOpt);
 }
 }
 }
 }
 };
 }
 </script>
</head>
<body>
 省: <select id="province"></select>

 市: <select id="city"></select>
</body>
```

```
</html>
```

48. 选择器的优先级问题 id class 标签 如果同为id 优先级怎么判断 比如

```
<div id="d1">
<style>
 #d1{ background:red}
 #d1{background:green}
</style>
```

参考答案：

id > class > 标签
后定义的 > 先定义的

49. 下面表达式的结果是多少？（腾讯）

参考答案：

①. js中 "5" +4=? 答案：54  
②. js中 void(0)=? 答案：undefined  
③. js中 NaN\*4=? 答案：NaN  
④. js中 null\*4.5=? 答案：0  
⑤. js中 alert(5\*015===5.075)结果?原因? 答案：false，原因是015是八进制数，换算成十进制是13，5\*13=65，当然不与5.075相等  
⑥. js中 13>>2=? -13>>2=? 答案：13>>2=3，-13>>2=-4  
⑦. js中 13|5=? 13&5=? 答案：13|5=13，13&5=5

50. js中怎么判断chrome浏览器？（腾讯）

参考答案：

```
var isChrome = window.navigator.userAgent.indexOf("Chrome") !== -1
```

51. 根据以下xml请写出对应的json（腾讯）

```
<xml>
 <list>
 <item>
 <id>12</id>
 <name>张三</name>
 </item>
 <item>
 <id>13</id>
 <name>李四</name>
 </item>
 </list>
</xml>
```

参考答案：

```
var items = [{id:12, name:'张三'}, {id:13, name:'李四'}]
```

52. js中怎么把十进制数123转化成二进制数？

参考答案：

```
var number = new Number(123);
alert(number.toString(2));
```

53. js中怎么才能按下回车键可以提交？

参考答案：

```
document.onkeydown = function(e) {
 e = e || window.event;
 if(e.keyCode == 13) {
 document.form1.submit();
 }
};
```

54. var s=" tencent is sb" ,编写js使其变成tencent1 is2 sb3

参考答案：

```
var s = "tencent is sb";
var array = s.split(" ");
s = "";
for(var i = 0; i < array.length; i++) {
 s += array[i] + (i + 1) + " ";
}
document.write(s);
```

55. 编写js的类，使其拥有public和private类型的属性和方法

参考答案：

```
function Person(_name, _age, _sex, _salary) {
 //public
 this.name = _name;
 this.age = _age;

 //private
 var sex = _sex;
 var salary = _salary;
```

```
//public method
this.getName = function() {
 return this.name;
}

this.getAge = function() {
 return this.age;
}
//private method
function getSex() {
 return sex;
}

function getSalary() {
 return salary;
}

this.display = function() {
 document.write(this.getName() + "--" + this.getAge() + "--" + getSex() + "--" +
 getSalary());
}
}

var smirk = new Person("zy", "21", "f", "5000");
smirk.display();
```

56. 给定 <http://jd.qq.com/125125>，请说出请求头，相关的报文信息(想想httpwatch工具抓包的内容)

参考答案：

57. 说出一些常用的网络优化工具

参考答案：