关于前端开发工程师的面试题

### 1、谈一谈对前端开发的理解，需要具备哪些知识和技术？你擅长哪方面？

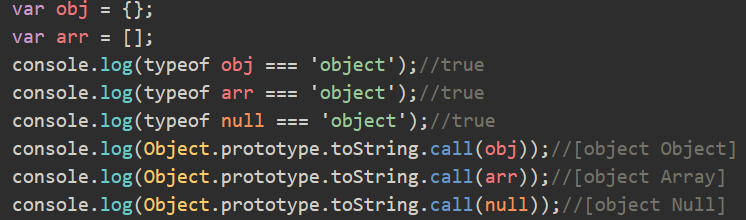
Web前端开发工程师既要与上游的交互设计师、视觉设计师和产品经理沟通，又要与下游的服务器端工程师沟通，需要掌握的技能非常多。一位好的Web前端开发工程师在知识体系上既要有广度，又要有深度，并且面对迅速发展的前端技术，保持敏感的触觉，不断学习。

前端开发需要具备的知识除了最基本的HTML5、CSS3、JavaScript，还要了解浏览器之间的异同，比如使用hack解决浏览器之间的兼容问题，还有浏览器之间的缓存策略；跨域技术解决同源策略造成的问题；页面优化的方法；常见的容易出现攻击的问题并及时进行防护，比如SQL注入，XSS攻击等；了解最新的框架，比如Vue.js，React，node.js。

本人基础知识比较扎实，比如基本的双飞翼布局和圣杯布局，还有hack技术解决浏览器之间的兼容问题，在JavaScript方面比如使用跨域技术解决同源策略造成的安全限制等，浏览器的缓存策略解决了cookie存储空间不足的问题。

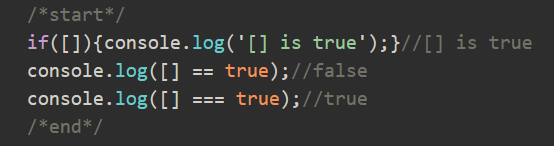
**2、使用 typeof bar === "object" 判断 bar 是不是一个对象有什么潜在的弊端？如何避免这种弊端？**

使用 typeof 的弊端是显而易见的(这种弊端同使用 instanceof)：

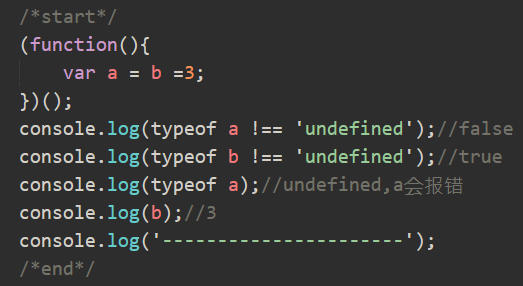


从上面的输出结果可知，typeof bar === "object" 并不能准确判断 bar 就是一个 Object。可以通过 Object.prototype.toString.call(bar) === "[object Object]" 来避免这种弊端。

另外，为了珍爱生命，请远离 ==：



3、**下面的代码会在 console 输出神马？为什么？**



跟变量作用域有关，拆解一下自执行函数中的变量赋值：

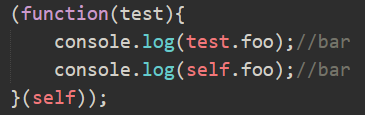
b = 3;  
var a = b;

所以 b 成了全局变量，而 a 是执行函数的一个局部变量。

**4、下面的代码会在 console 输出神马？为什么？**



第一个和第二个的输出不难判断，在ES6之前，JavaScript 只有函数作用域，所以func中的IIFE（立即调用的函数表达式）有自己的独立作用域，并且它能访问到外部作用域中的 self，所以第三个输出会报错，因为 this 在可访问到的作用域内是 undefined，第四个输出是 bar。如果你知道闭包，也很容易解决的：



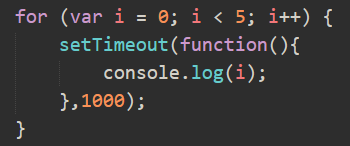
闭包是指有权访问另外一个函数作用域中的变量的函数。从概念上说，闭包有两个特点： 1、函数 2、能访问另外一个函数作用域中的变量

在ES6之前，Javascript只有函数作用域的概念，没有块级作用域（catch捕获的异常只能在catch块中访问）的概念（IIFE可以创建局部作用域）。每个函数作用域都是封闭的，即外部是访问不到函数作用域中的变量。

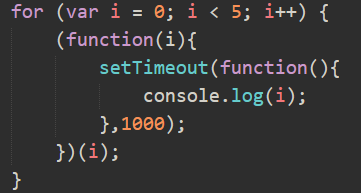
**5、将 JavaScript 代码包含在一个函数块中有什么意思呢？为什么要这么做？**

换句话说，为什么要用立即执行函数表达式（Immediately-Invoked Function Expression）。

IIFE 有两个比较经典的使用场景，一是类似于在循环中定时输出数据项，二是类似于 JQuery/Node 的插件和模块开发。



上面的输出并不是你以为的0，1，2，3，4，而输出的全部是5，这时 IIFE 就能有用了：

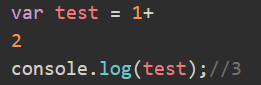


**6、在严格模式('use strict')下进行 JavaScript 开发有什么好处？**

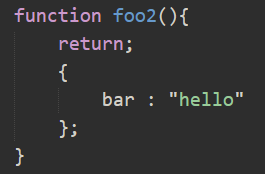
消除Javascript语法的一些不合理、不严谨之处，减少一些怪异行为;  
消除代码运行的一些不安全之处，保证代码运行的安全；  
提高编译器效率，增加运行速度；  
为未来新版本的Javascript做好铺垫。  
**7、下面两个函数的返回值是一样的吗？为什么？**



在编程语言中，基本都是使用分号（;）将语句分隔开，这可以增加代码的可读性和整洁性。而在JS中，如若语句各占独立一行，通常可以省略语句间的分号（;），JS 解析器会根据能否正常编译来决定是否自动填充分号：



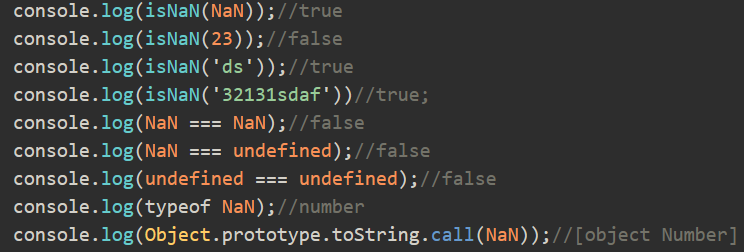
在上述情况下，为了正确解析代码，就不会自动填充分号了，但是对于 return 、break、continue 等语句，如果后面紧跟换行，解析器一定会自动在后面填充分号(;)，所以上面的第二个函数就变成了这样：



所以第二个函数是返回 undefined。

**8、什么是 NaN，它的类型是什么？怎么测试一个值是否等于 NaN?**

NaN 是 Not a Number 的缩写，JavaScript 的一种特殊数值，其类型是 Number，可以通过 isNaN(param) 来判断一个值是否是 NaN。ES6 中，isNaN() 成为了 Number 的静态方法：Number.isNaN().



**9、解释一下下面代码的输出**



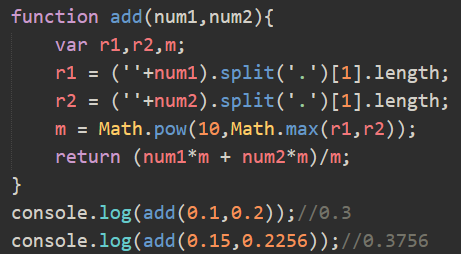
JavaScript 中的 number 类型就是浮点型，JavaScript 中的浮点数采用IEEE-754 格式的规定，这是一种二进制表示法，可以精确地表示分数，比如1/2，1/8，1/1024，每个浮点数占64位。但是，二进制浮点数表示法并不能精确的表示类似0.1这样的简单的数字，会有舍入误差。

由于采用二进制，JavaScript 也不能有限表示 1/10、1/2 等这样的分数。在二进制中，1/10(0.1)被表示为 0.00110011001100110011…… 注意 0011 是无限重复的，这是舍入误差造成的，所以对于 0.1 + 0.2 这样的运算，操作数会先被转成二进制，然后再计算：

**0.1 => 0.0001 1001 1001 1001…（无限循环）  
0.2 => 0.0011 0011 0011 0011…（无限循环）** 双精度浮点数的小数部分最多支持52位，所以两者相加之后得到这么一串 0.0100110011001100110011001100110011001100…因浮点数小数位的限制而截断的二进制数字，这时候，再把它转换为十进制，就成了 0.30000000000000004。

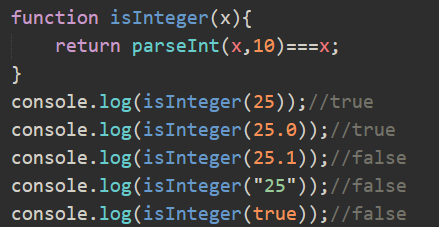
对于保证浮点数计算的正确性，有两种常见方式。

**一是先升幂再降幂：**

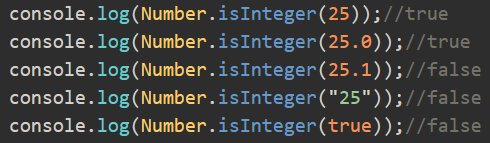


**10、实现函数 isInteger(x) 来判断 x 是否是整数**

可以将 x 转换成10进制，判断和本身是不是相等即可：



ES6 对数值进行了扩展，提供了静态方法 isInteger() 来判断参数是否是整数：

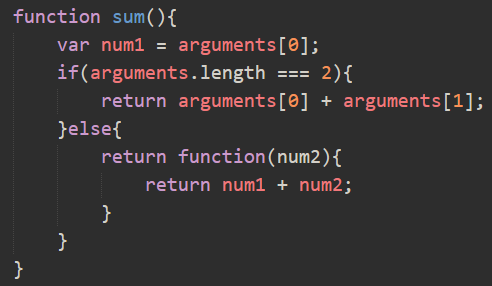


JavaScript能够准确表示的整数范围在 -2^53 到 2^53 之间（不含两个端点），超过这个范围，无法精确表示这个值。ES6 引入了Number.MAX\_SAFE\_INTEGER 和 Number.MIN\_SAFE\_INTEGER这两个常量，用来表示这个范围的上下限，并提供了 Number.isSafeInteger() 来判断整数是否是安全型整数。

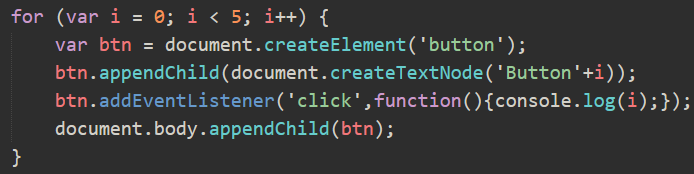
**13、写一个按照下面方式调用都能正常工作的 sum 方法**



针对这个题，可以判断参数个数来实现：



**14、根据下面的代码片段回答后面的问题**



1、点击 Button 4，会在控制台输出什么？

2、给出一种符合预期的实现方式

1、点击5个按钮中的任意一个，都是输出5

2、参考 IIFE。

**19、解释下列代码的输出**

[?](http://www.jb51.net/article/77140.htm)

|  |  |
| --- | --- |
| 1  2  3  4 | console.log("0 || 1 = "+(0 || 1));  console.log("1 || 2 = "+(1 || 2));  console.log("0 && 1 = "+(0 && 1));  console.log("1 && 2 = "+(1 && 2)); |

逻辑与和逻辑或运算符会返回一个值，并且二者都是短路运算符：

逻辑与返回第一个是 false 的操作数 或者 最后一个是 true的操作数

[?](http://www.jb51.net/article/77140.htm)

|  |  |
| --- | --- |
| 1  2  3 | console.log(1 && 2 && 0); //0  console.log(1 && 0 && 1); //0  console.log(1 && 2 && 3); //3 |

如果某个操作数为 false，则该操作数之后的操作数都不会被计算

逻辑或返回第一个是 true 的操作数 或者 最后一个是 false的操作数

[?](http://www.jb51.net/article/77140.htm)

|  |  |
| --- | --- |
| 1  2  3 | console.log(1 || 2 || 0); //1  console.log(0 || 2 || 1); //2  console.log(0 || 0 || false); //false |

如果某个操作数为 true，则该操作数之后的操作数都不会被计算

如果逻辑与和逻辑或作混合运算，则逻辑与的优先级高：

[?](http://www.jb51.net/article/77140.htm)

|  |  |
| --- | --- |
| 1  2  3 | console.log(1 && 2 || 0); //2  console.log(0 || 2 && 1); //1  console.log(0 && 2 || 1); //1 |

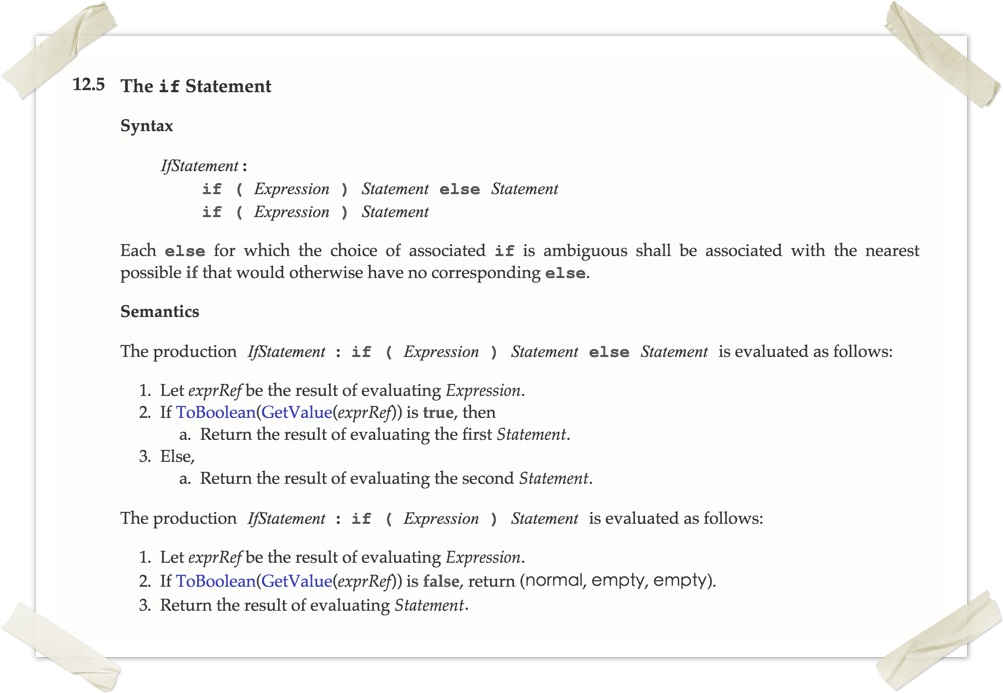
在 JavaScript，常见的 false 值：

0, '0', +0, -0, false, '',null,undefined,null,NaN  
要注意空数组([])和空对象({}):

[?](http://www.jb51.net/article/77140.htm)

|  |  |
| --- | --- |
| 1  2  3  4 | console.log([] == false) //true  console.log({} == false) //false  console.log(Boolean([])) //true  console.log(Boolean({})) //true |

所以在 if 中，[] 和 {} 都表现为 true：

[](http://files.jb51.net/file_images/article/201512/2015122992646146.png?2015112992655)

**21、解释下面代码的输出**

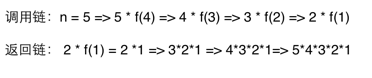
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | var a={},   b={key:'b'},   c={key:'c'};    a[b]=123;  a[c]=456;    console.log(a[b]); |

在指定a[b]时，数组中的b被转化成了"[object Object]"其实是指定了a["[object Object]"]，在给a[c]赋值时重写了a["[object Object]"]，所以输出是456。

**22、解释下面代码的输出**

|  |  |
| --- | --- |
| 1 | console.log((function f(n){return ((n > 1) ? n \* f(n-1) : n)})(10)); |

结果是10的阶乘。这是一个递归调用，为了简化，我初始化 n=5，则调用链和返回链如下：



**23、解释下面代码的输出**

|  |  |
| --- | --- |
| 1  2  3  4  5 | (function(x) {   return (function(y) {   console.log(x);   })(2)  })(1); |

输出1，闭包能够访问外部作用域的变量或参数。

**24、解释下面代码的输出，并修复存在的问题**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | var hero = {   \_name: 'John Doe',   getSecretIdentity: function (){   return this.\_name;   }  };    var stoleSecretIdentity = hero.getSecretIdentity;    console.log(stoleSecretIdentity());  console.log(hero.getSecretIdentity()); |

将 getSecretIdentity 赋给 stoleSecretIdentity，等价于定义了 stoleSecretIdentity 函数：

[?](http://www.jb51.net/article/77140.htm)

|  |  |
| --- | --- |
| 1  2  3  4 | var stoleSecretIdentity = function (){   return this.\_name;  }  stoleSecretIdentity |

的上下文是全局环境，所以第一个输出 undefined。若要输出 John Doe，则要通过 call 、apply 和 bind 等方式改变 stoleSecretIdentity 的this 指向(hero)。

第二个是调用对象的方法，输出 John Doe。

**25、给你一个 DOM 元素，创建一个能访问该元素所有子元素的函数，并且要将每个子元素传递给指定的回调函数。**

函数接受两个参数：

1. **DOM**
2. **指定的回调函数**

原文利用 深度优先搜索(Depth-First-Search) 给了一个实现：

[?](http://www.jb51.net/article/77140.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | function Traverse(p\_element,p\_callback) {   p\_callback(p\_element);   var list = p\_element.children;   for (var i = 0; i < list.length; i++) {   Traverse(list[i],p\_callback); // recursive call   }  } |

26、题目1：回答以下代码，alert的值分别是多少？

1. <span style="font-family:Microsoft YaHei;font-size:18px;"><script>
2. **var** a = 10;
3. **function** test() {
4. a = 100;
5. alert(a);
6. alert(**this**.a);
7. **var** a;
8. alert(a);
9. }
10. test();
11. </script>
12. </span>

正确答案是：100， 10， 100

题目2：回答以下代码，alert的值分别是多少？

1. <span style="font-family:Microsoft YaHei;font-size:18px;"><script>
2. **var** a = 100;
3. **function** test(){
4. alert(a);
5. **var** a = 10;
6. alert(a);
7. }
8. test();
9. </script>
10. </span>

正确答案是： undefined， 10

题目3：回答以下代码，alert的值分别是多少？

1. <span style="font-family:Microsoft YaHei;font-size:18px;"><script>
2. **var** a = 100;
3. **function** test(){
4. alert(a);
5. a = 10;  //去掉了var 就变成定义了全局变量了
6. alert(a);
7. }
8. test();
9. alert(a);
10. </script></span>

正确答案是： 100， 10， 10

27、截取字符串abcdefg的efg  
**alert('abcdefg'.substring(4));**  
  
28、判断一个字符串中出现次数最多的字符，统计这个次数

var str = 'asdfssaaasasasasaa';  
var json = {};  
for (var i = 0; i < str.length; i++) {  
        if(!json[str.charAt(i)]){  
                json[str.charAt(i)] = 1;  
        }else{  
                json[str.charAt(i)]++;  
        }  
};  
var iMax = 0;  
var iIndex = '';  
for(var i in json){  
        if(json[i]>iMax){  
                iMax = json[i];  
                iIndex = i;  
        }  
}  
alert('出现次数最多的是:'+iIndex+'出现'+iMax+'次');  
  
29、IE与FF脚本兼容性问题  
obj.addEventListener(sEv, fn, false);  
obj.attachEvent('on'+sEv,fn);  
detachevet  
removeEventListener  
DOMContentLoaded  
onreadystatechange  complete  
DOMMouseScroll FF  
onmousewheel   非FF  
event.wheelDelta 上滚120 下-120  
event.detail     上滚-3   下3    
obj.getCurrentStyle[attr]  
getComputedStyle(obj,false)[attr]  
XMLHttpRequest  
ActiveXObject('Mircorsoft.XMLHttp')  
FF本地能设置读取cookie 其他不行  
event  ev  
事件源  
srcElement||target  
toElement||relatedTarget  
obj.setCapture();只有ie认  
obj.releaseCapture();  
  
30、规避javascript多人开发函数重名问题  
命名空间  
封闭空间  
js模块化mvc（数据层、表现层、控制层）  
seajs  
变量转换成对象的属性  
对象化  
  
31、jvascript面向对象中继承实现  
function Person(name){  
        this.name = name;  
}  
Person.prototype.showName = function(){  
        alert(this.name);  
}  
function Worker(name, job){  
        Person.apply(this,arguments)  
        this.job = job;  
}  
for(var i in Person.prototype){  
        Worker.prototype = Person.prototype;  
}  
new Worker('sl', 'coders').showName();  
  
32、FF下面实现outerHTML  
/\*var oDiv = document.createElement('div');  
var oDiv1 = document.getElementById('div1');  
var oWarp = document.getElementById('warp');  
  
oWarp.insertBefore(oDiv, oDiv1);  
oDiv.appendChild(oDiv1);  
var sOut = oDiv.innerHTML;  
oWarp.insertBefore(oDiv1, oDiv);  
oWarp.removeChild(oDiv);  
alert(sOut);\*/  
  
  
33、编写一个方法，求一个字符串的字节长度

/假设一个中文占两个字节  
var str = '22两是';  
alert(getStrlen(str))  
function getStrlen(str){  
        var json = {len:0};  
        var re = /[\u4e00-\u9fa5]/;  
        for (var i = 0; i < str.length; i++) {  
                if(re.test(str.charAt(i))){  
                        json['len']++;  
                }  
        };  
        return json['len']+str.length;  
}  
34、编写一个方法 去掉一个数组的重复元素  
/\*var arr = [1,2,3,1,43,12,12,1];  
var json = {};  
var arr2 = [];  
for (var i = 0; i < arr.length; i++) {  
        if(!json[arr[i]]){  
                json[arr[i]] = true;  
        }else{  
                json[arr[i]] = false;  
        }  
        if(json[arr[i]]){  
                arr2.push(arr[i]);  
        }  
};  
for (var i = 0; i < arr.length; i++) {  
        if(!aa(arr[i], arr2)){  
                arr2.push(arr[i])  
        }  
};  
function aa(obj, arr){  
        for (var i = 0; i < arr.length; i++) {  
                if(arr[i] == obj) return true;  
                else return false;  
        };  
}  
alert(arr2)\*/  
35、写出3个使用this的典型应用  
事件： 如onclick  this->发生事件的对象  
构造函数          this->new 出来的object  
call/apply        改变this  
36、如何深度克隆  
var arr = [1,2,43];  
var json = {a:6,b:4,c:[1,2,3]};  
var str = 'sdfsdf';  
var json2 = clone(json);  
alert(json['c'])  
function clone(obj){  
        var oNew = new obj.constructor(obj.valueOf());  
        if(obj.constructor == Object){  
                for(var i in obj){  
                        oNew[i] = obj[i];  
                        if(typeof(oNew[i]) == 'object'){  
                                clone(oNew[i]);  
                        }  
                }  
        }  
        return oNew;  
}  
37、JavaScript中如何检测一个变量是一个String类型？请写出函数实现  
// typeof(obj) == 'string'  
// obj.constructor == String;  
38、网页中实现一个计算当年还剩多少时间的倒数计时程序，要求网页上实时动态显示

“××年还剩××天××时××分××秒”  
var oDate = new Date();  
var oYear = oDate.getFullYear();  
var oNewDate = new Date();  
oNewDate.setFullYear(oYear, 11, 31, 23, 59, 59);  
var iTime = oNewDate.getTime()-oDate.getTime();  
var iS = iTime/1000;  
var iM = oNewDate.getMonth()-oDate.getMonth();  
var iDate =iS\*/  
39、 请解释一下什么是语义化的HTML。  
//内容使用特定标签，通过标签就能大概了解整体页面的布局分布  
40、 为什么利用多个域名来存储网站资源会更有效？  
//确保用户在不同地区能用最快的速度打开网站，其中某个域名崩溃用户也能通过其他郁闷访问网站  
  
41、说出三种减低页面加载时间的方法  
1、压缩css、js文件  
2、合并js、css文件，减少http请求  
3、外部js、css文件放在最底下  
4、减少dom操作，尽可能用变量替代不必要的dom操作  
41、什是FOUC？你如何来避免FOUC？  
/\*由于css引入使用了@import 或者存在多个style标签以及css文件在页面底部引入使得css文件加载在html之后导致页面闪烁、花屏  
用link加载css文件，放在head标签里面\*/  
42、文档类型的作用是什么？你知道多少种文档类型？  
/\*影响浏览器对html代码的编译渲染  
html2.0  
xHtml  
html5\*/  
  
43、浏览器标准模式和怪异模式之间的区别是什么？  
//盒模型解释不同  
  
44、使用过那些Javascript库？  
//jquery seajs yui  
45、哈希表  
//具有散列（映射）特性的数据模型  
  
46、闭包  
//子函数能被外部调用到，则该作用连上的所有变量都会被保存下来。  
  
47、请解释什么是Javascript的模块模式，并举出实用实例。  
/\*js模块化mvc（数据层、表现层、控制层）  
seajs  
命名空间\*/  
  
48、你如何组织自己的代码？是使用模块模式，还是使用经典继承的方法？  
/\*对内：模块模式  
对外：继承\*/  
  
49、 你如何优化自己的代码？  
/\*代码重用  
避免全局变量（命名空间，封闭空间，模块化mvc..）  
拆分函数避免函数过于臃肿  
注释  
\*/  
50.你能解释一下JavaScript中的继承是如何工作的吗？  
/\*子构造函数中执行父构造函数，并用call\apply改变this  
克隆父构造函数原型上的方法\*/  
51、请尽可能详尽的解释AJAX的工作原理。  
/\*创建ajax对象（XMLHttpRequest/ActiveXObject(Microsoft.XMLHttp)）  
判断数据传输方式(GET/POST)  
打开链接 open()  
发送 send()  
当ajax对象完成第四步（onreadystatechange）数据接收完成，判断http响应状态（status）200-300之间或者304（缓存）执行回调函数\*/

52、JS闭包   
f = function() {return true;};     
g = function() {return false;};     
(function() {     
   if (g() && [] == ![]) {     
      f = function f() {return false;};     
      function g() {return true;}     
   }     
})();     
alert(f()); // true or false ?    
－－－－－－－－－－－－－－－－－－－－－－－－   
答案：   
(function() {       
   if (g() && [] == ![]) {       
//应该看成if((g() && [] )== ![])     
//因为g()是false后面那个&&[]就没起作用 整个都是false      
//![]也是false 所以if成立 进入if块内     
      f = function f() {return false;};       
//重新定义f      
      function g() {return true;}       
//这句没用     
   }       
})();      
alert(f());     
//false    
  
53、截取字符串abcdefg的efg   
abcdefg   
    <script type="text/javascript">   
  var mytext=document.getElementByIdx\_x\_x("text");   
  var myvalue=mytext.innerHTML;   
  var jiequ=myvalue.substring(myvalue.length-3,myvalue.length);   
  alert(jiequ)   
</script>   
  
54、写出一下运算结果   
  
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) // string   
55、写出函数DateDemo的返回结果，系统时间假定为今天   
function DateDemo(){   
var d, s="今天日期是：";   
d = new Date();   
s += d.getMonth() + "/";   
s += d.getDate() + "/";   
s += d.getYear();   
return s;   
}   
结果：今天日期是：7/17/2010   
  
56、写出程序运行的结果？    
for(i=0, j=0; i<10, j<6; i++, j++){   
k = i + j;   
57、以下哪个在JAVASCRIPT中会报错:   
var a=();   
var a=[];   
var a={};   
var a=//;

答案：var a=();var a=//； 报错     两个不行！  
58、JAVA中有哪些保留字,越多越好,10个以上

int  char  while  for  do  switch  void  double  float  unsigned  long  try  abstract   super  extent  bool  break  case  catch  class  delegate foreach in   static  void  public private protected internal  等等  太多了    
59、CSS中有几种声明方式.

1.导入一个已经存在的CSS文件    
<link rel="stylesheet" type="text/css" href="">      
2.直接在HARD里声明一个CSS代码段：   
<style type="text/css">      
   .....  </style>     
3.直接在网页元素里声明使用CSS：  style="color:red;"    如果是声明CSS类 ， 则有三种：  直接标签名作为CSS类名称：  TagName{ ... }  以元素ID作为CSS类声明：    #ID{...}  自定义CSS类名称 ：         .ClassName{}     
  
  
60、写出你所知道的开源项目或软件,并说明他们的好处以及作用

开源的    客户端脚本的话就说说 ExtJS    她可以用来开发RIA也即富客户端的AJAX应用，是一个用javascript写的，主要用于创建前端用户界面，是一个与后台技术无关的前端ajax框架。因此，可以把ExtJS用在.Net、Java、Php等各种开发语言开发的应 用中。  而且，使用这个框架可以直接用一行代码实现非常漂亮的而且标准的网页控件界面，可以使用代码调用的控件来替代目前的流行网页布局，和WinForm程序一样，使得前端开发完全控件化。  Jquery也是优秀的前台框架，优点：  代码简练、语义易懂、学习快速、文档丰富。    jQuery是一个轻量级的脚本，其代码非常小巧，最新版的JavaScript包只有20K左右。    jQuery支持CSS1-CSS3,以及基本的xPath。    jQuery是跨浏览器的，它支持的浏览器包括IE 6.0+, FF 1.5+, Safari 2.0+, Opera 9.0+。    可以很容易的为jQuery扩展其他功能。    能将JS代码和HTML代码完全分离，便于代码和维护和修改。    插件丰富，除了jQuery本身带有的一些特效外，可以通过插件实现更多功能，如表单验证、tab导航、拖放效果、表格排序、DataGrid，树形菜单、图像特效以及ajax上传等。     除此之外，还有ProtoType,Bindows(JS框架)   服务，ZedGragh(.Net做统计图的),Spring,Hibernate,Structs,EJB，Jbmp等比较有名的

61、form中的input有哪些类

 text   radio   checkbox   file   button   image   submit   reset   hidden   
62、table标签中border,cellpadding     td标签中colspan,rowspan分别起什么作用？

border边界   
cellpadding边距   
colspan跨列数   
rowspan跨行数   
  
63、form中的input可以设置readonly和disable，请问这两项属性有什么区别？

readonly不可编辑,但可以选择和复制   
disable不能编辑复制选择   
64、JS中的三种弹出式消息提醒(警告窗口、确认窗口、信息输入窗口)的命令是什么？  
alert   
confirm   
prompt   
**作用域**

65、考虑如下代码：

JavaScript

|  |  |
| --- | --- |
|  | function() {  var a = b = 5;  })();  console.log(b);  请问控制台上会输出什么？  答案：输出 5 |

这一题的陷阱是，在函数表达式中有两个赋值，但a是用关键字var 来声明的，这意味着a是局部变量，而b则被赋予为全局变量。

另一个陷阱是，它并没有使用严格模式（use strict）。在函数里面，如果启用了严格模式，代码就会报错：“Uncaught ReferenceError: b is not defined”。请记住，严格模式需要你显式地引用全局作用域，代码应该写成：

|  |
| --- |
| (function() {  'use strict';  var a = window.b = 5;  })();  console.log(b); |

**66、创建“内置”方法**

给String对象定义一个repeatify方法。该方法接收一个整数参数，作为字符串重复的次数，最后返回重复指定次数的字符串。例如：

|  |  |
| --- | --- |
| 1 | console.log('hello'.repeatify(3));  输出应该是：hellohellohello |

**答案**

一个可行的做法如下

|  |
| --- |
| String.prototype.repeatify = String.prototype.repeatify || function(times) {  var str = '';  for (var i = 0; i < times; i++) {  str += this;  }  return str;  }; |

这题测试开发者对Javascript的继承及原型属性的知识，它同时也检验了开发者是否能扩展内置数据类型的方法。

这里的另一个关键点是，看你怎样避免重写可能已经定义了的方法。这可以通过在定义自己的方法之前，检测方法是否已经存在。

|  |  |
| --- | --- |
| 1 | String.prototype.repeatify = String.prototype.repeatify || function(times) {/\* code here \*/}; |

当你被问起去扩展一个Javascript方法时，这个技术非常有用。

**67、声明提前**

下面这段代码的结果是什么？为什么？

|  |
| --- |
| function test() {  console.log(a);  console.log(foo());  var a = 1;  function foo() {  return 2;  }}  test(); |

**答案**

代码的运行结果：undefined和 2

理由是，变量和函数的声明都被提前至函数体的顶部，而同时变量并没有被赋值。因此，当打印变量a时，它虽存在于函数体（因为a已经被声明），但仍然是undefined。换句话说，上面的代码等同于下面的代码：

|  |
| --- |
| function test() {  var a;  function foo() {  return 2;  }  console.log(a);  console.log(foo());  a = 1;  }  test(); |

**68、JavaScript中的this**

下面代码的运行结果是什么并做解释

|  |  |
| --- | --- |
|  | var fullname = 'John Doe';  var obj = {  fullname: 'Colin Ihrig',  prop: {  fullname: 'Aurelio De Rosa',  getFullname: function() {  return this.fullname;  }}};  console.log(obj.prop.getFullname());  var test = obj.prop.getFullname;  console.log(test()); |

**答案**

代码输出：Aurelio De Rosa 和 John Doe

理由是，Javascript中关键字this所指代的函数上下文，取决于函数是怎样被调用的，而不是怎样被定义的。

在第一个console.log()，getFullname()被作为obj.prop对象被调用。因此，当前的上下文指代后者，函数返回这个对象的fullname属性。相反，当getFullname()被赋予test变量，当前的上下文指代全局对象window，这是因为test被隐式地作为全局对象的属性。基于这一点，函数返回window的fullname，在本例中即为代码的第一行。

**69、jQuery的美元符号$有什么作用？**

回答：其实美元符号$只是”jQuery”的别名，它是jQuery的选择器，如下代码：

$(document).ready(function(){

});

当然你也可以用jQuery来代替$，如下代码：

jQuery(document).ready(function(){

});

jQuery中就是通过这个美元符号来实现各种灵活的DOM元素选择的，例如$(“#main”)即选中id为main的元素。

**70、body中的onload()函数和jQuery中的document.ready()有什么区别？**

回答：onload()和document.ready()的区别有以下两点：

1、我们可以在页面中使用多个document.ready()，但只能使用一次onload()。

2、document.ready()函数在页面DOM元素加载完以后就会被调用，而onload()函数则要在所有的关联资源（包括图像、音频）加载完毕后才会调用。

**71、jQuery中有哪几种类型的选择器？**

回答：从我自己的角度来讲，可以有3种类型的选择器，如下：

1、基本选择器：直接根据id、css类名、元素名返回匹配的dom元素。

2、层次选择器：也叫做路径选择器，可以根据路径层次来选择相应的DOM元素。

3、过滤选择器：在前面的基础上过滤相关条件，得到匹配的dom元素。

**72、请使用jQuery将页面上的所有元素边框设置为2px宽的虚线？**

回答：这正是jQuery选择器上场的时候了，代码如下：

<script language="javascript" type="text/javascript">

**$("\*").css("border", "2px dotted red");**

</script>

**73、当CDN上的jQuery文件不可用时，该怎么办？**

回答：为了节省带宽和脚本引用的稳定性，我们会使用CDN上的jQuery文件，例如google的jquery cdn服务。但是如果这些CDN上的jQuery服务不可用，我们还可以通过以下代码来切换到本地服务器的jQuery版本：

<script type="text/javascript" language="Javascript" src="http://ajax.aspnetcdn.com/ajax/jquery/jquery-1.4.1.min.js "></script>

<script type='text/javascript'>//<![CDATA[

if (typeof jQuery == 'undefined') {

document.write(unescape("%3Cscript src='/Script/jquery-1.4.1.min.js' type='text/javascript' %3E%3C/script%3E"));

}//]]>

</script>

**74、如何使用jQuery实现点击按钮弹出一个对话框？**

回答：代码如下：

HTML：

**<input id="inputField" type="text" size="12"/>**

jQuery：

**<script type="text/javascript"> $(document).ready(function () { $('#Button1').click(function () { alert($('#inputField').attr("value")); }); }); </script>**

**75、jQuery中的Delegate()函数有什么作用？**

回答：delegate()会在以下两个情况下使用到：

1、如果你有一个父元素，需要给其下的子元素添加事件，这时你可以使用delegate()了，代码如下：

$("ul").delegate("li", "click", function(){

$(this).hide();

});

2、当元素在当前页面中不可用时，可以使用delegate()

**76、怎样用jQuery编码和解码URL？**

回答：在jQuery中，我们可以使用以下方法实现URL的编码和解码。

encodeURIComponent(url) and decodeURIComponent(url)

**77、如何用jQuery禁用浏览器的前进后退按钮？**

回答：实现代码如下：

<script type="text/javascript" language="javascript">

$(document).ready(function() {

window.history.forward(1);

//OR

window.history.forward(-1);

});

</script>

78、**你做的页面在哪些流览器测试过？这些浏览器的内核分别是什么?**

• IE: trident内核   
• Firefox：gecko内核   
• Safari:webkit内核  
• Opera:以前是presto内核，Opera现已改用Google Chrome的Blink内核

• Chrome:Blink([基于webkit，Google与Opera Software共同开发](http://baike.baidu.com/view/1369399.htm" \l "1_5" \t "_blank))

**79、每个HTML文件里开头都有个很重要的东西，Doctype，知道这是干什么的吗？**

　　答案：<!DOCTYPE> 声明位于文档中的最前面的位置，处于 <html> 标签之前。此标签可告知浏览器文档使用哪种 HTML 或 XHTML 规范。  
（重点：告诉浏览器按照何种规范解析页面）

**80、Quirks模式是什么？它和Standards模式有什么区别**

　　答案：

　　从IE6开始，引入了Standards模式，标准模式中，浏览器尝试给符合标准的文档在规范上的正确处理达到在指定浏览器中的程度。

　　在IE6之前CSS还不够成熟，所以IE5等之前的浏览器对CSS的支持很差， IE6将对CSS提供更好的支持，然而这时的问题就来了，因为有很多页面是基于旧的布局方式写的，而如果IE6 支持CSS则将令这些页面显示不正常，如何在即保证不破坏现有页面，又提供新的渲染机制呢？

　　在写程序时我们也会经常遇到这样的问题，如何保证原来的接口不变，又提供更强大的功能，尤其是新功能不兼容旧功能时。遇到这种问题时的一个常见做法是增加参数和分支，即当某个参数为真时，我们就使用新功能，而如果这个参数 不为真时，就使用旧功能，这样就能不破坏原有的程序，又提供新功能。IE6也是类似这样做的，它将DTD当成了这个“参数”，因为以前的页面大家都不会去写DTD，所以IE6就假定 如果写了DTD，就意味着这个页面将采用对CSS支持更好的布局，而如果没有，则采用兼容之前的布局方式。这就是Quirks模式（怪癖模式，诡异模式，怪异模式）。

　　区别：

　　总体会有布局、样式解析和脚本执行三个方面的区别。

　　盒模型：在W3C标准中，如果设置一个元素的宽度和高度，指的是元素内容的宽度和高度，而在Quirks 模式下，IE的宽度和高度还包含了padding和border。



设置行内元素的高宽：在Standards模式下，给<span>等行内元素设置wdith和height都不会生效，而在quirks模式下，则会生效。  
　　设置百分比的高度：在standards模式下，一个元素的高度是由其包含的内容来决定的，如果父元素没有设置百分比的高度，子元素设置一个百分比的高度是无效的用margin:0 auto设置水平居中：使用margin:0 auto在standards模式下可以使元素水平居中，但在quirks模式下却会失效。  
　　（还有很多，答出什么不重要，关键是看他答出的这些是不是自己经验遇到的，还是说都是看文章看的，甚至完全不知道。）  
   
**81、div+css的布局较table布局有什么优点？**

•改版的时候更方便 只要改css文件。  
•页面加载速度更快、结构化清晰、页面显示简洁。  
•表现与结构相分离。  
•易于优化（seo）搜索引擎更友好，排名更容易靠前。  
   
**82、a：img的alt与title有何异同？b：strong与em的异同？**

　　答案：  
　　a:  
•alt(alt text):为不能显示图像、窗体或applets的用户代理（UA），alt属性用来指定替换文字。替换文字的语言由lang属性指定。(在IE浏览器下会在没有title时把alt当成 tool tip显示)  
•title(tool tip):该属性为设置该属性的元素提供建议性的信息。  
　　b:  
•strong:粗体强调标签，强调，表示内容的重要性  
•em:斜体强调标签，更强烈强调，表示内容的强调点  
　　  
**83、你能描述一下渐进增强和优雅降级之间的不同吗?**

•渐进增强 progressive enhancement：针对低版本浏览器进行构建页面，保证最基本的功能，然后再针对高级浏览器进行效果、交互等改进和追加功能达到更好的用户体验。  
•优雅降级 graceful degradation：一开始就构建完整的功能，然后再针对低版本浏览器进行兼容。  
   
　　区别：优雅降级是从复杂的现状开始，并试图减少用户体验的供给，而渐进增强则是从一个非常基础的，能够起作用的版本开始，并不断扩充，以适应未来环境的需要。降级（功能衰减）意味着往回看；而渐进增强则意味着朝前看，同时保证其根基处于安全地带。　  
　  
　　“优雅降级”观点  
　　“优雅降级”观点认为应该针对那些最高级、最完善的浏览器来设计网站。而将那些被认为“过时”或有功能缺失的浏览器下的测试工作安排在开发周期的最后阶段，并把测试对象限定为主流浏览器（如 IE、Mozilla 等）的前一个版本。  
　　在这种设计范例下，旧版的浏览器被认为仅能提供“简陋却无妨 (poor, but passable)” 的浏览体验。你可以做一些小的调整来适应某个特定的浏览器。但由于它们并非我们所关注的焦点，因此除了修复较大的错误之外，其它的差异将被直接忽略。  
　　“渐进增强”观点  
　　“渐进增强”观点则认为应关注于内容本身。  
　　内容是我们建立网站的诱因。有的网站展示它，有的则收集它，有的寻求，有的操作，还有的网站甚至会包含以上的种种，但相同点是它们全都涉及到内容。这使得“渐进增强”成为一种更为合理的设计范例。这也是它立即被 Yahoo! 所采纳并用以构建其“分级式浏览器支持 (Graded Browser Support)”策略的原因所在。  
   
　　那么问题了。现在产品经理看到IE6,7,8网页效果相对高版本现代浏览器少了很多圆角，阴影（CSS3），要求兼容（使用图片背景，放弃CSS3），你会如何说服他？  
　　（自由发挥）  
　　  
**84、为什么利用多个域名来存储网站资源会更有效？**

•CDN缓存更方便   
•突破浏览器并发限制   
•节约cookie带宽   
•节约主域名的连接数，优化页面响应速度   
•防止不必要的安全问题  
   
**85、请谈一下你对网页标准和标准制定机构重要性的理解。**

　　（无标准答案）网页标准和标准制定机构都是为了能让web发展的更‘健康’，开发者遵循统一的标准，降低开发难度，开发成本，SEO也会更好做，也不会因为滥用代码导致各种BUG、安全问题，最终提高网站易用性。  
   
**86、请描述一下cookies，sessionStorage和localStorage的区别？**  
　　sessionStorage用于本地存储一个会话（session）中的数据，这些数据只有在同一个会话中的页面才能访问并且当会话结束后数据也随之销毁。因此sessionStorage不是一种持久化的本地存储，仅仅是会话级别的存储。而localStorage用于持久化的本地存储，除非主动删除数据，否则数据是永远不会过期的。  
　　web storage和cookie的区别  
•Web Storage的概念和cookie相似，区别是它是为了更大容量存储设计的。Cookie的大小是受限的，并且每次你请求一个新的页面的时候Cookie都会被发送过去，这样无形中浪费了带宽，另外cookie还需要指定作用域，不可以跨域调用。  
•除此之外，Web Storage拥有setItem,getItem,removeItem,clear等方法，不像cookie需要前端开发者自己封装setCookie，getCookie。但是Cookie也是不可以或缺的：Cookie的作用是与服务器进行交互，作为HTTP规范的一部分而存在 ，而Web Storage仅仅是为了在本地“存储”数据而生。  
   
**87、简述一下src与href的区别。**

　　答案：  
　　src用于替换当前元素，href用于在当前文档和引用资源之间确立联系。  
　　src是source的缩写，指向外部资源的位置，指向的内容将会嵌入到文档中当前标签所在位置；在请求src资源时会将其指向的资源下载并应用到文档内，例如js脚本，img图片和frame等元素。  
　　<script src ="js.js"></script>  
　　当浏览器解析到该元素时，会暂停其他资源的下载和处理，直到将该资源加载、编译、执行完毕，图片和框架等元素也如此，类似于将所指向资源嵌入当前标签内。这也是为什么将js脚本放在底部而不是头部。  
   
　　href是Hypertext Reference的缩写，指向网络资源所在位置，建立和当前元素（锚点）或当前文档（链接）之间的链接，如果我们在文档中添加  
　　<link href="common.css" rel="stylesheet"/>  
　　那么浏览器会识别该文档为css文件，就会并行下载资源并且不会停止对当前文档的处理。这也是为什么建议使用link方式来加载css，而不是使用@import方式。  
   
**88、知道的网页制作会用到的图片格式有哪些？**

　　答案：  
　　png-8，png-24，jpeg，gif，svg。  
　　但是上面的那些都不是面试官想要的最后答案。面试官希望听到是Webp,Apng。（是否有关注新技术，新鲜事物）  
　　科普一下Webp：WebP格式，谷歌（google）开发的一种旨在加快图片加载速度的图片格式。图片压缩体积大约只有JPEG的2/3，并能节省大量的服务器带宽资源和数据空间。Facebook Ebay等知名网站已经开始测试并使用WebP格式。  
　　在质量相同的情况下，[WebP](http://baike.baidu.com/view/4447461.htm" \t "_blank)格式图像的体积要比JPEG格式图像小40%。  
  
**Apng**：全称是**“Animated Portable Network Graphics”**, 是PNG的位图动画扩展，可以实现png格式的动态图片效果。04年诞生，但一直得不到各大浏览器厂商的支持，直到日前得到 iOS safari 8的支持，有望代替GIF成为下一代动态图标准。

89、知道什么是微格式吗？谈谈理解。在前端构建中应该考虑微格式吗？

　　答案：

　　微格式（Microformats）是一种让机器可读的语义化XHTML词汇的集合，是结构化数据的开放标准。是为特殊应用而制定的特殊格式。

　　优点：将智能数据添加到网页上，让网站内容在搜索引擎结果界面可以显示额外的提示。（应用范例：豆瓣，有兴趣自行google）

90、在css/js代码上线之后开发人员经常会优化性能，从用户刷新网页开始，一次js请求一般情况下有哪些地方会有缓存处理？

　　答案：dns缓存，cdn缓存，浏览器缓存，服务器缓存。

91、一个页面上有大量的图片（大型电商网站），加载很慢，你有哪些方法优化这些图片的加载，给用户更好的体验。

1. 图片懒加载，在页面上的未可视区域可以添加一个滚动条事件，判断图片位置与浏览器顶端的距离与页面的距离，如果前者小于后者，优先加载。
2. 如果为幻灯片、相册等，可以使用图片预加载技术，将当前展示图片的前一张和后一张优先下载。
3. 如果图片为css图片，可以使用CSSsprite，SVGsprite，Iconfont、Base64等技术。
4. 如果图片过大，可以使用特殊编码的图片，加载时会先加载一张压缩的特别厉害的缩略图，以提高用户体验。
5. 如果图片展示区域小于图片的真实大小，则因在服务器端根据业务需要先行进行图片压缩，图片压缩后大小与展示一致。

92、你如何理解HTML结构的语义化？

1. 去掉或样式丢失的时候能让页面呈现清晰的结构：

html本身是没有表现的，我们看到例如<h1>是粗体，字体大小2em，加粗；<strong>是加粗的，不要认为这是html的表现，这些其实html默认的css样式在起作用，所以去掉或样式丢失的时候能让页面呈现清晰的结构不是语义化的HTML结构的优点，但是浏览器都有有默认样式，默认样式的目的也是为了更好的表达html的语义，可以说浏览器的默认样式和语义化的HTML结构是不可分割的。

1. 屏幕阅读器（如果访客有视障）会完全根据你的标记来“读”你的网页.

　　例如,如果你使用的含语义的标记,屏幕阅读器就会“逐个拼出”你的单词,而不是试着去对它完整发音.

1. PDA、手机等设备可能无法像普通电脑的浏览器一样来渲染网页（通常是因为这些设备对CSS的支持较弱）

　　使用语义标记可以确保这些设备以一种有意义的方式来渲染网页.理想情况下,观看设备的任务是符合设备本身的条件来渲染网页.

　　语义标记为设备提供了所需的相关信息,就省去了你自己去考虑所有可能的显示情况（包括现有的或者将来新的设备）.例如,一部手机可以选择使一段标记了标题的文字以粗体显示.而掌上电脑可能会以比较大的字体来显示.无论哪种方式一旦你对文本标记为标题,您就可以确信读取设备将根据其自身的条件来合适地显示页面.

1. 搜索引擎的爬虫也依赖于标记来确定上下文和各个关键字的权重

　　过去你可能还没有考虑搜索引擎的爬虫也是网站的“访客”,但现在它们他们实际上是极其宝贵的用户.没有他们的话,搜索引擎将无法索引你的网站,然后一般用户将很难过来访问.

1. 你的页面是否对爬虫容易理解非常重要,因为爬虫很大程度上会忽略用于表现的标记,而只注重语义标记.

　　因此,如果页面文件的标题被标记,而不是,那么这个页面在搜索结果的位置可能会比较靠后.除了提升易用性外,语义标记有利于正确使用CSS和JavaScript,因为其本身提供了许多“钩钩”来应用页面的样式与行为.  
SEO主要还是靠你网站的内容和外部链接的。

1. 便于团队开发和维护

　　W3C给我们定了一个很好的标准，在团队中大家都遵循这个标准，可以减少很多差异化的东西，方便开发和维护，提高开发效率，甚至[实现模块化开发](http://www.cssforest.org/blog/index.php?id=134)。

93、谈谈以前端角度出发做好SEO需要考虑什么？

1. 了解搜索引擎如何抓取网页和如何索引网页

　　你需要知道一些搜索引擎的基本工作原理，各个搜索引擎之间的区别，搜索机器人（SE robot 或叫 web crawler）如何进行工作，搜索引擎如何对搜索结果进行排序等等。

1. Meta标签优化

　　主要包括主题（Title)，网站描述(Description)，和关键词（Keywords）。还有一些其它的隐藏文字比如Author（作者），Category（目录），Language（编码语种）等。

1. 如何选取关键词并在网页中放置关键词

　　搜索就得用关键词。关键词分析和选择是SEO最重要的工作之一。首先要给网站确定主关键词（一般在5个上下），然后针对这些关键词进行优化，包括关键词密度（Density），相关度（Relavancy），突出性（Prominency）等等。

1. 了解主要的搜索引擎

　　虽然搜索引擎有很多，但是对网站流量起决定作用的就那么几个。比如英文的主要有Google，Yahoo，Bing等；中文的有百度，搜狗，有道等。不同的搜索引擎对页面的抓取和索引、排序的规则都不一样。还要了解各搜索门户和搜索引擎之间的关系，比如AOL网页搜索用的是Google的搜索技术，MSN用的是Bing的技术。

1. 主要的互联网目录

　　Open Directory自身不是搜索引擎，而是一个大型的网站目录，他和搜索引擎的主要区别是网站内容的收集方式不同。目录是人工编辑的，主要收录网站主页；搜索引擎是自动收集的，除了主页外还抓取大量的内容页面。

1. 按点击付费的搜索引擎

　　搜索引擎也需要生存，随着互联网商务的越来越成熟，收费的搜索引擎也开始大行其道。最典型的有Overture和百度，当然也包括Google的广告项目Google Adwords。越来越多的人通过搜索引擎的点击广告来定位商业网站，这里面也大有优化和排名的学问，你得学会用最少的广告投入获得最多的点击。

1. 搜索引擎登录

　　网站做完了以后，别躺在那里等着客人从天而降。要让别人找到你，最简单的办法就是将网站提交（submit）到搜索引擎。如果你的是商业网站，主要的搜索引擎和目录都会要求你付费来获得收录（比如Yahoo要299美元），但是好消息是（至少到目前为止）最大的搜索引擎Google目前还是免费，而且它主宰着60％以上的搜索市场。

1. 链接交换和链接广泛度（Link Popularity）

　　网页内容都是以超文本（Hypertext）的方式来互相链接的，网站之间也是如此。除了搜索引擎以外，人们也每天通过不同网站之间的链接来Surfing（“冲浪”）。其它网站到你的网站的链接越多，你也就会获得更多的访问量。更重要的是，你的网站的外部链接数越多，会被搜索引擎认为它的重要性越大，从而给你更高的排名。

1. 合理的标签使用

**Css篇：**

94、有哪项方式可以对一个DOM设置它的CSS样式？

•外部样式表，引入一个外部css文件  
•内部样式表，将css代码放在 <head> 标签内部  
•内联样式，将css样式直接定义在 HTML 元素内部

95、CSS都有哪些选择器？

•派生选择器（用HTML标签申明）  
•id选择器（用DOM的ID申明）  
•类选择器（用一个样式类名申明）  
•属性选择器（用DOM的属性申明，属于CSS2，IE6不支持，不常用，不知道就算了）  
除了前3种基本选择器，还有一些扩展选择器，包括

•后代选择器（利用空格间隔，比如div .a{  }）  
•群组选择器（利用逗号间隔，比如p,div,#a{  }）　  
那么问题来了，CSS选择器的优先级是怎么样定义的？

基本原则：

一般而言，选择器越特殊，它的优先级越高。也就是选择器指向的越准确，它的优先级就越高。

复杂的计算方法：

•用1表示派生选择器的优先级  
•用10表示类选择器的优先级  
•用100标示ID选择器的优先级   
•div.test1 .span var 优先级 1+10 +10 +1    
•span#xxx .songs li 优先级1+100 + 10 + 1    
•#xxx li 优先级 100 +1

那么问题来了，看下列代码，<p>标签内的文字是什么颜色的？。

**CSS Code**复制内容到剪贴板

1. <style>
2. .classA{ **color**:blue;}
3. .classB{ **color**:red;}
4. </style>
5. <body>
6. <p class='classB classA'> 123 </p>
7. </body>

答案：red。与样式定义在文件中的先后顺序有关，即是后面的覆盖前面的，与在<p class='classB classA'>中的先后关系无关。

96、CSS中可以通过哪些属性定义，使得一个DOM元素不显示在浏览器可视范围内？

最基本的：

设置display属性为none，或者设置visibility属性为hidden

技巧性：

设置宽高为0，设置透明度为0，设置z-index位置在-1000

97、超链接访问过后hover样式就不出现的问题是什么？如何解决？

答案：被点击访问过的超链接样式不在具有hover和active了,解决方法是改变CSS属性的排列顺序: L-V-H-A（link,visited,hover,active）

98、什么是Css Hack？ie6,7,8的hack分别是什么？

答案：针对不同的浏览器写不同的CSS code的过程，就是CSS hack。

示例如下：

**CSS Code**复制内容到剪贴板

1. #test       {
2. **width**:300px;
3. **height**:300px;
5. **background-color**:blue;      /\*firefox\*/
6. **background-color**:red\9;      /\*all ie\*/
7. **background-color**:yellow\0;    /\*ie8\*/
8. +**background-color**:pink;        /\*ie7\*/
9. \_background-**color**:orange;       /\*ie6\*/    }
10. :root #test { **background-color**:purple\9; }  /\*ie9\*/
11. @media all and (**min-width**:0px){ #test {**background-color**:black\0;} }  /\*opera\*/
12. @media screen and (-webkit-min-device-pixel-ratio:0){ #test {**background-color**:gray;} }       /\*chrome and safari\*/

99、请用Css写一个简单的幻灯片效果页面   
  
　　答案：知道是要用css3。使用animation动画实现一个简单的幻灯片效果。

**CSS Code**复制内容到剪贴板

1. /\*\*HTML\*\*/
2. div.ani
4. /\*\*css\*\*/
5. .ani{
6. **width**:480px;
7. **height**:320px;
8. **margin**:50px auto;
9. **overflow**: hidden;
10. box-shadow:0 0 5px rgba(0,0,0,1);
11. **background**-**size**: cover;
12. **background-position**: center;
13. -webkit-animation-name: "loops";
14. -webkit-animation-duration: 20s;
15. -webkit-animation-iteration-count: infinite;
16. }
17. @-webkit-keyframes "loops" {
18. 0% {
19. **background**:url(http://d.hiphotos.baidu.com/image/w%3D400/sign=c01e6adca964034f0fcdc3069fc27980/e824b899a9014c08e5e38ca4087b02087af4f4d3.jpg) no-repeat;
20. }
21. 25% {
22. **background**:url(http://b.hiphotos.baidu.com/image/w%3D400/sign=edee1572e9f81a4c2632edc9e72b6029/30adcbef76094b364d72bceba1cc7cd98c109dd0.jpg) no-repeat;
23. }
24. 50% {
25. **background**:url(http://b.hiphotos.baidu.com/image/w%3D400/sign=937dace2552c11dfded1be2353266255/d8f9d72a6059252d258e7605369b033b5bb5b912.jpg) no-repeat;
26. }
27. 75% {
28. **background**:url(http://g.hiphotos.baidu.com/image/w%3D400/sign=7d37500b8544ebf86d71653fe9f9d736/0df431adcbef76095d61f0972cdda3cc7cd99e4b.jpg) no-repeat;
29. }
30. 100% {
31. **background**:url(http://c.hiphotos.baidu.com/image/w%3D400/sign=cfb239ceb0fb43161a1f7b7a10a54642/3b87e950352ac65ce2e73f76f9f2b21192138ad1.jpg) no-repeat;
32. }
33. }

100、行内元素和块级元素的具体区别是什么？行内元素的padding和margin可设置吗？

块级元素(block)特性：

•总是独占一行，表现为另起一行开始，而且其后的元素也必须另起一行显示;  
•宽度(width)、高度(height)、内边距(padding)和外边距(margin)都可控制;  
内联元素(inline)特性：  
•和相邻的内联元素在同一行;  
•宽度(width)、高度(height)、内边距的top/bottom(padding-top/padding-bottom)和外边距的top/bottom(margin-top/margin-bottom)都不可改变（也就是padding和margin的left和right是可以设置的），就是里面文字或图片的大小。  
那么问题来了，浏览器还有默认的天生inline-block元素（拥有内在尺寸，可设置高宽，但不会自动换行），有哪些？

答案：<input> 、<img> 、<button> 、<textarea> 、<label>。

101、什么是外边距重叠？重叠的结果是什么？

答案：

外边距重叠就是margin-collapse。

在CSS当中，相邻的两个盒子（可能是兄弟关系也可能是祖先关系）的外边距可以结合成一个单独的外边距。这种合并外边距的方式被称为折叠，并且因而所结合成的外边距称为折叠外边距。

折叠结果遵循下列计算规则：

1.两个相邻的外边距都是正数时，折叠结果是它们两者之间较大的值。  
2.两个相邻的外边距都是负数时，折叠结果是两者绝对值的较大值。  
3.两个外边距一正一负时，折叠结果是两者的相加的和。

102、rgba()和opacity的透明效果有什么不同？

答案：

rgba()和opacity都能实现透明效果，但最大的不同是opacity作用于元素，以及元素内的所有内容的透明度，  
而rgba()只作用于元素的颜色或其背景色。（设置rgba透明的元素的子元素不会继承透明效果！）

103、css中可以让文字在垂直和水平方向上重叠的两个属性是什么？

答案：

垂直方向：line-height  
水平方向：letter-spacing

那么问题来了，关于letter-spacing的妙用知道有哪些么？  
答案:可以用于消除inline-block元素间的换行符空格间隙问题。

104、居中一个浮动元素？

**CSS Code**复制内容到剪贴板

1. // 方法一：已知元素的高宽
3. #div1{
4. **background-color**:#6699FF;
5. **width**:200px;
6. **height**:200px;
8. **position**: absolute;        //父元素需要相对定位
9. top: 50%;
10. left: 50%;
11. **margin-top**:-100px ;   //二分之一的**height**，**width**
12. **margin-left**: -100px;
13. }
15. //方法二:未知元素的高宽
17. #div1{
18. **width**: 200px;
19. **height**: 200px;
20. **background-color**: #6699FF;
22. **margin**:auto;
23. **position**: absolute;        //父元素需要相对定位
24. left: 0;
25. top: 0;
26. **right**right: 0;
27. **bottom**bottom: 0;
28. }

那么问题来了，如何垂直居中一个<img>?（用更简便的方法。）

**CSS Code**复制内容到剪贴板

1. #container     //<img>的容器设置如下
2. {
3. **display**:table-cell;
4. **text-align**:center;
5. **vertical-align**:middle;
6. }

105、px和em的区别  
　　px和em都是长度单位，区别是，px的值是固定的，指定是多少就是多少，计算比较容易。em得值不是固定的，并且em会继承父级元素的字体大小。   
  
　　浏览器的默认字体高都是16px。所以未经调整的浏览器都符合: 1em=16px。那么12px=0.75em, 10px=0.625em。   
  
106、描述一个"reset"的CSS文件并如何使用它。知道normalize.css吗？你了解他们的不同之处？　   
　重置样式非常多，凡是一个前端开发人员肯定有一个常用的重置CSS文件并知道如何使用它们。他们是盲目的在做还是知道为什么这么做呢？原因是不同的浏览器对一些元素有不同的默认样式，如果你不处理，在不同的浏览器下会存在必要的风险，或者更有戏剧性的性发生。   
  
　　你可能会用[Normalize](http://necolas.github.io/normalize.css/" \t "_blank)来代替你的重置样式文件。它没有重置所有的样式风格，但仅提供了一套合理的默认样式值。既能让众多浏览器达到一致和合理，但又不扰乱其他的东西（如粗体的标题）。   
  
在这一方面，无法做每一个复位重置。它也确实有些超过一个重置，它处理了你永远都不用考虑的怪癖，像HTML的audio元素不一致或line-height不一致。   
  
107、.Sass、LESS是什么？大家为什么要使用他们？   
　　他们是CSS预处理器。他是CSS上的一种抽象层。他们是一种特殊的语法/语言编译成CSS。   
  
　　例如[Less](http://www.lesscss.org/" \t "_blank)是一种动态样式语言. 将CSS赋予了动态语言的特性，如变量，继承，运算， 函数. LESS 既可以在客户端上运行 (支持IE 6+, Webkit, Firefox)，也可一在服务端运行 (借助 Node.js)。   
  
为什么要使用它们？   
  
•结构清晰，便于扩展。   
•可以方便地屏蔽浏览器私有语法差异。这个不用多说，封装对浏览器语法差异的重复处理，减少无意义的机械劳动。   
•可以轻松实现多重继承。   
•完全兼容 CSS 代码，可以方便地应用到老项目中。LESS 只是在 CSS 语法上做了扩展，所以老的 CSS 代码也可以与 LESS 代码一同编译。   
　　   
108、display:none与visibility:hidden的区别是什么？   
  
•display : 隐藏对应的元素但不挤占该元素原来的空间。   
•visibility: 隐藏对应的元素并且挤占该元素原来的空间。   
　　即是，使用CSS display:none属性后，HTML元素（对象）的宽度、高度等各种属性值都将“丢失”;而使用visibility:hidden属性后，HTML元素（对象）仅仅是在视觉上看不见（完全透明），而它所占据的空间位置仍然存在。　　   
  
109、知道css有个content属性吗？有什么作用？有什么应用？   
　　答案：   
　　知道。css的content属性专门应用在 before/after 伪元素上，用于来插入生成内容。   
  
　　最常见的应用是利用伪类清除浮动。

**CSS Code**复制内容到剪贴板

1. //一种常见利用伪类清除浮动的代码
2. .clearfix:after {
3. **content**:".";       //这里利用到了**content**属性
4. **display**:block;
5. **height**:0;
6. **visibility**:hidden;
7. **clear**:both; }
9. .clearfix {
10. \*zoom:1;
11. }

after伪元素通过 content 在元素的后面生成了内容为一个点的块级元素，再利用clear:both清除浮动。   
  
　　那么问题继续还有，知道css计数器（序列数字字符自动递增）吗？如何通过css content属性实现css计数器？   
  
　　答案：css计数器是通过设置counter-reset 、counter-increment 两个属性 、及 counter()/counters()一个方法配合after / before 伪类实现。

110**、线程与进城的区别**

一个程序至少有一个进程,一个进程至少有一个线程.

线程的划分尺度小于进程，使得多线程程序的并发性高。

另外，进程在执行过程中拥有独立的内存单元，而多个线程共享内存，从而极大地提高了程序的运行效率。

线程在执行过程中与进程还是有区别的。每个独立的线程有一个程序运行的入口、顺序执行序列和程序的出口。但是线程不能够独立执行，必须依存在应用程序中，由应用程序提供多个线程执行控制。

从逻辑角度来看，多线程的意义在于一个应用程序中，有多个执行部分可以同时执行。但操作系统并没有将多个线程看做多个独立的应用，来实现进程的调度和管理以及资源分配。这就是进程和线程的重要区别。

**111请解释一下什么是“语义化的 HTML”**

语义化的好处：1：去掉或样式丢失的时候能让页面呈现清晰的结构：

 html本身是没有表现的，我们看到例如<h1>是粗体，字体大小2em，加粗；<strong>是加粗的，不要认为这是html的表现，这些其实html默认的css样式在起作用，所以去掉或样式丢失的时候能让页面呈现清晰的结构不是语义化的HTML结构的优点，但是浏览器都有有默认样式，默认样式的目的也是为了更好的表达html的语义，可以说浏览器的默认样式和语义化的HTML结构是不可分割的。

 2.屏幕阅读器（如果访客有视障）会完全根据你的标记来“读”你的网页.

 3.PDA、手机等设备可能无法像普通电脑的浏览器一样来渲染网页（通常是因为这些设备对CSS的支持较弱）.

 4.搜索引擎的爬虫也依赖于标记来确定上下文和各个关键字的权重.

 5.你的页面是否对爬虫容易理解非常重要,因为爬虫很大程度上会忽略用于表现的标记,而只注重语义标记.

 6.便于团队开发和维护

 语义化的HTML就是：标题用h1-h6，文字段落用p，列表用ul li，大致如此

**112你如何对网站的文件和资源进行优化？**

期待的解决方案包括：

 文件合并

 文件最小化/文件压缩

 使用 CDN 托管

 缓存的使用（多个域名来提供缓存）

 其他

**113、请说出三种减少页面加载时间的方法。（加载时间指感知的时间或者实际加载时间）**

 1.优化图片 2.图像格式的选择（GIF：提供的颜色较少，可用在一些对颜色要求不高的地方）

 3.优化CSS（压缩合并css，如margin-top,margin-left...) 4.网址后加斜杠（如www.campr.com/目录，会判断这个“目录是什么文件类型，或者是目录。） 5.标明高度和宽度（如果浏览器没有找到这两个参数，它需要一边下载图片一边计算大小，如果图片很多，浏览器需要不断地调整页面。这不但影响速度，也影响浏览体验。

当浏览器知道了高度和宽度参数后，即使图片暂时无法显示，页面上也会腾出图片的空位，然后继续加载后面的内容。从而加载时间快了，浏览体验也更好了。） 6.减少http请求（合并文件，合并图片）。

**114、doctype（文档类型）的作用是什么？你知道多少种文档类型？**

此标签可告知浏览器文档使用哪种 HTML 或 XHTML 规范。

 该标签可声明三种 DTD 类型，分别表示严格版本、过渡版本以及基于框架的 HTML 文档。

 HTML 4.01 规定了三种文档类型：Strict、Transitional 以及 Frameset。

 XHTML 1.0 规定了三种 XML 文档类型：Strict、Transitional 以及 Frameset。

Standards （标准）模式（也就是严格呈现模式）用于呈现遵循最新标准的网页，而 Quirks

 （包容）模式（也就是松散呈现模式或者兼容模式）用于呈现为传统浏览器而设计的网页。

**115、讨论CSS hacks，条件引用或者其他。**

background-color:blue; 各个浏览器都认识，这里给firefox用；background-color:red\9;\9所有的ie浏览器可识别；background-color:yellow\0; \0 是留给ie8的+background-color:pink; + ie7定了；\_background-color:orange; \_专门留给神奇的ie6；:root #test { background-color:purple\9; } :root是给ie9的，@media all and (min-width:0px){ #test {background-color:black\0;} } 这个是老是跟ie抢着认\0的神奇的opera，必须加个\0,不然firefox，chrome，safari也都认识。。。@media screen and (-webkit-min-device-pixel-ratio:0){ #test {background-color:gray;} }最后这个是浏览器新贵chrome和safari的。

**116、你用过媒体查询，或针对移动端的布局/CSS 吗？**

@media screen and (min-width: 400px) and (max-width: 700px) { … }

@media handheld and (min-width: 20em), screen and (min-width: 20em) { … }

媒体查询，就是响应式布局。 响应式布局框架：Bootstrap、PureCSS等

**117、解释下原型继承的原理。**

当查找一个对象的属性时，JavaScript 会向上遍历原型链，直到找到给定名称的属性为止。(*大多数JavaScript的实现用 \_\_proto\_\_ 属性来表示一个对象的原型链。*)

以下代码展示了JS引擎如何查找属性：

function getProperty(obj, prop) {

    if (obj.hasOwnProperty(prop))

        return obj[prop]

    else if (obj.\_\_proto\_\_ !== null)

        return getProperty(obj.\_\_proto\_\_, prop)

    else

        return undefined

}

**注：**\_\_proto\_\_ 是一个不应在你代码中出现的非正规的用法，这里仅仅用它来解释JavaScript原型继承的工作原理。

**118、js数组主要有哪些方法？主要参数你了解吗？**

具体你可以查一下，常用的一般有如下：

1 、shift():删除数组的第一个元素,返回删除的值。这里是0

2 、unshift(3,4):把参数加载数组的前面，返回数组的长度。现在list:中是3,4,0,1,2

3、pop():删除数组的最后一个元素，返回删除的值。这里是2.

4、push(3):将参数加载到数组的最后，返回数组的长度，现在List中时：0,1,2,3

5、concat(3,4):把两个数组拼接起来。

6、splice(start,deleteCount,val1,val2,...)：从start位置开始删除deleteCount项，并从该位置起插入val1,val2,...

7、reverse：将数组反序

var a = [1,2,3,4,5];

var b = a.reverse(); //a：[5,4,3,2,1] b：[5,4,3,2,1]

8、sort(orderfunction)：按指定的参数对数组进行排序 var a = [1,2,3,4,5]; var b = a.sort(); //a：[1,2,3,4,5] b：[1,2,3,4,5]

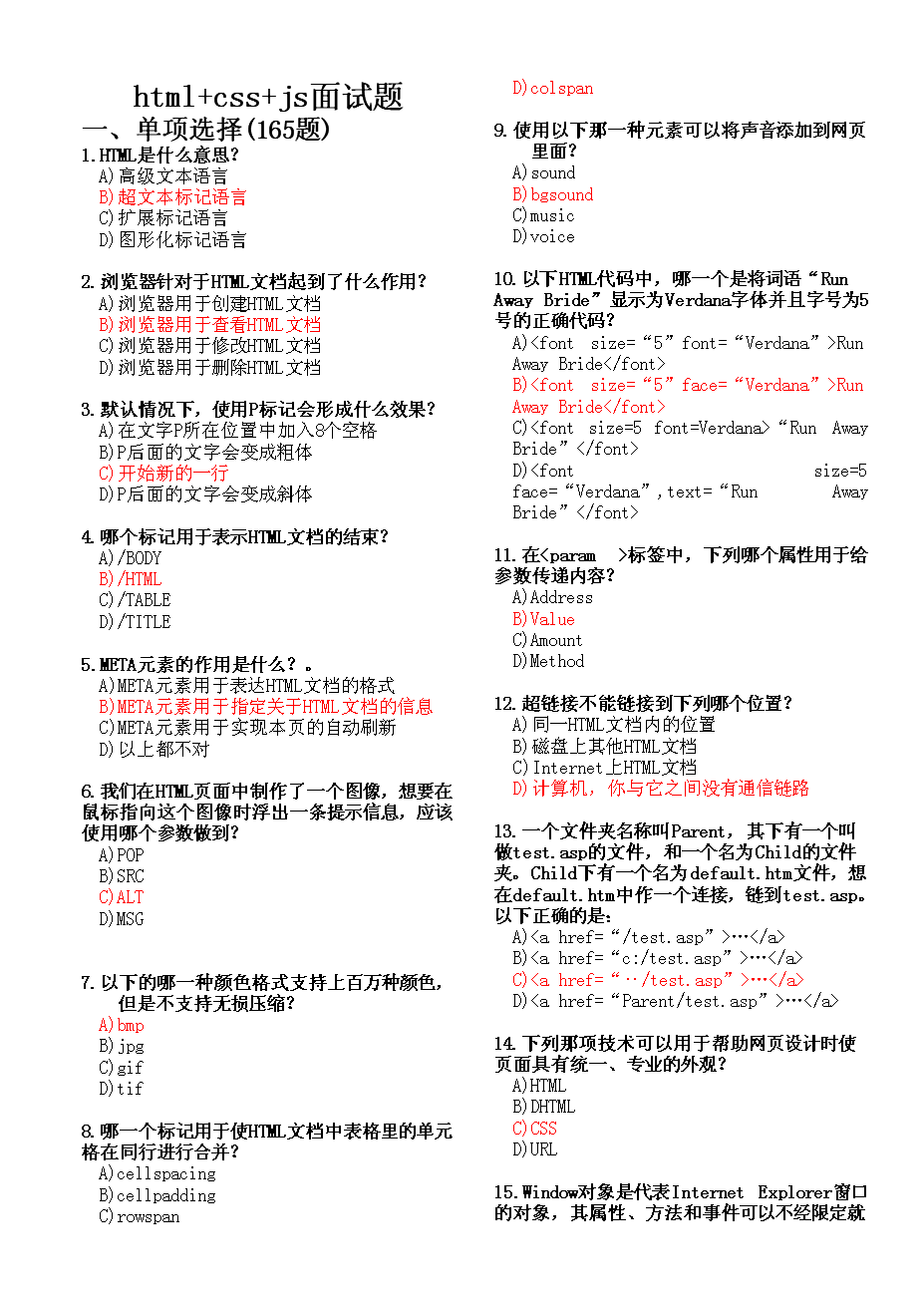
9、slice(start,end)：返回从原数组中指定开始下标到结束下标之间的项组成的新数组

var a = [1,2,3,4,5];

var b = a.slice(2,5); //a：[1,2,3,4,5] b：[3,4,5]

**119、js迭代的方法**

every() 、fliter()、forEach()、map()、some()具体大家可以查一下！



 [](http://images.wenku.it168.com/doc/2013-04-13/895690/F02E51F1DE134F51404C768AA2005DA0.png)