**Css**

HTML+CSS  
1.对WEB标准以及W3C的理解与认识  
标签闭合、标签小写、不乱嵌套、提高搜索机器人搜索几率、使用外 链css和js脚本、结构行为表现的分离、文件下载与页面速度更快、内容能被更多的用户所访问、内容能被更广泛的设备所访问、更少的代码和组件，容易维 护、改版方便，不需要变动页面内容、提供打印版本而不需要复制内容、提高网站易用性；

2.xhtml和html有什么区别  
HTML是一种基本的WEB网页设计语言，XHTML是一个基于XML的置标语言  
最主要的不同：  
XHTML 元素必须被正确地嵌套。  
XHTML 元素必须被关闭。  
标签名必须用小写字母。  
XHTML 文档必须拥有根元素。

4.行内元素有哪些?块级元素有哪些?CSS的盒模型?  
块级元素：div p h1 h2 h3 h4 form ul  
行内元素: a b br i span input select  
Css盒模型:内容，border ,margin，padding

5.CSS引入的方式有哪些? link和@import的区别是?  
内联 内嵌 外链 导入  
区别 ：同时加载  
前者无兼容性，后者CSS2.1以下浏览器不支持  
Link 支持使用javascript改变样式，后者不可

6.CSS选择符有哪些?哪些属性可以继承?优先级算法如何计算?内联和important哪个优先级高?  
标签选择符 类选择符 id选择符  
继承不如指定 Id>class>标签选择  
后者优先级高

7.前端页面有哪三层构成，分别是什么?作用是什么?  
结构层 Html 表示层 CSS 行为层 js

8.css的基本语句构成是?  
选择器{属性1:值1;属性2:值2;……}

9.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(Ie内核) 火狐（Gecko） 谷歌（webkit） opear(Presto)

11.标签上title与alt属性的区别是什么?  
Alt 当图片不显示是 用文字代表。  
Title 为该属性提供信息

12.描述css reset的作用和用途。  
Reset重置浏览器的css默认属性 浏览器的品种不同，样式不同，然后重置，让他们统一

13.解释css sprites，如何使用。  
Css 精灵 把一堆小的图片整合到一张大的图片上，减轻服务器对图片的请求数量

14.浏览器标准模式和怪异模式之间的区别是什么?  
盒子模型 渲染模式的不同  
使用 window.top.document.compatMode 可显示为什么模式

15.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

16.什么是语义化的HTML?  
直观的认识标签 对于搜索引擎的抓取有好处

17.清除浮动的几种方式，各自的优缺点  
1.使用空标签清除浮动 clear:both（理论上能清楚任何标签，，，增加无意义的标签）  
2.使用overflow:auto（空标签元素清除浮动而不得不增加无意代码的弊端,,使用zoom:1用于兼容IE）  
3.是用afert伪元素清除浮动(用于非IE浏览器)

**18.写出几种IE6 BUG的解决方法**

1.双边距BUG float引起的  使用display

2.3像素问题 使用float引起的 使用dislpay:inline -3px

3.超链接hover 点击后失效  使用正确的书写顺序 link visited hover active

4.Ie z-index问题 给父级添加position:relative

5.Png 透明 使用js代码 改

6.Min-height 最小高度 ！Important 解决’  
7.select 在ie6下遮盖 使用iframe嵌套

8.为什么没有办法定义1px左右的宽度容器（IE6默认的行高造成的，使用over:hidden,zoom:0.08 line-height:1px）

**11.<img>标签上title与alt属性的区别是什么？**

Alt 当图片不显示是 用文字代表。

Title 为该属性提供信息

**12.描述css reset的作用和用途。**

Reset重置浏览器的css默认属性        浏览器的品种不同，样式不同，然后重置，让他们统一

**13.解释css sprites，如何使用。**

Css 精灵 把一堆小的图片整合到一张大的图片上，减轻服务器对图片的请求数量

**14.浏览器标准模式和怪异模式之间的区别是什么？**

盒子模型 渲染模式的不同

使用 window.top.document.compatMode 可显示为什么模式

**17.清除浮动的几种方式，各自的优缺点**

1.使用空标签清除浮动 clear:both（理论上能清楚任何标签，，，增加无意义的标签）

2.使用overflow:auto（空标签元素清除浮动而不得不增加无意代码的弊端,,使用zoom:1用于兼容IE）

3.是用afert伪元素清除浮动(用于非IE浏览器)

Javascript

1.javascript的typeof返回哪些数据类型  
Object number function boolean underfind  
  
2.例举3种强制类型转换和2种隐式类型转换?  
强制（parseInt,parseFloat,number）  
隐式（== – ===）  
  
3.split() join() 的区别  
前者是切割成数组的形式，后者是将数组转换成字符串  
  
4.数组方法pop() push() unshift() shift()  
Push()尾部添加 pop()尾部删除  
Unshift()头部添加 shift()头部删除

5.事件绑定和普通事件有什么区别

6.IE和DOM事件流的区别  
1.执行顺序不一样、  
2.参数不一样  
3.事件加不加on  
4.this指向问题

7.IE和标准下有哪些兼容性的写法  
Var ev = ev || window.event  
document.documentElement.clientWidth || document.body.clientWidth  
Var target = ev.srcElement||ev.target

8.ajax请求的时候get 和post方式的区别  
一个在url后面 一个放在虚拟载体里面  
有大小限制  
安全问题  
应用不同 一个是论坛等只需要请求的，一个是类似修改密码的

9.call和apply的区别  
Object.call(this,obj1,obj2,obj3)  
Object.apply(this,arguments)

10.ajax请求时，如何解释json数据  
使用eval parse 鉴于安全性考虑 使用parse更靠谱

11.b继承a的方法  
  
12.写一个获取非行间样式的函数  
  
function getStyle(obj,attr,value)  
{  
if(!value)  
{  
if(obj.currentStyle)  
{  
return obj.currentStyle(attr)  
}  
else  
{  
obj.getComputedStyle(attr,false)  
}  
}  
else  
{  
obj.style[attr]=value  
}  
}

13.事件委托是什么  
让利用事件冒泡的原理，让自己的所触发的事件，让他的父元素代替执行！  
http://www.webasily.com/?p=78 例子可见此链接

14.闭包是什么，有什么特性，对页面有什么影响  
闭包就是能够读取其他函数内部变量的函数。  
http://blog.csdn.net/gaoshanwudi/article/details/7355794 此链接可查看（问这个问题的不是一个公司）

15.如何阻止事件冒泡和默认事件  
canceBubble return false

16.添加 删除 替换 插入到某个接点的方法  
obj.appendChidl()  
obj.innersetBefore  
obj.replaceChild  
obj.removeChild

17.解释jsonp的原理，以及为什么不是真正的ajax  
动态创建script标签，回调函数  
Ajax是页面无刷新请求数据操作

18.javascript的本地对象，内置对象和宿主对象  
本地对象为array obj regexp等可以new实例化  
内置对象为gload Math 等不可以实例化的  
宿主为浏览器自带的document,window 等

19.document load 和document ready的区别  
Document.onload 是在结构和样式加载完才执行js  
Document.ready原生种没有这个方法，jquery中有 $().ready(function)

20.”==”和“===”的不同  
前者会自动转换类型  
后者不会

21.javascript的同源策略  
一段脚本只能读取来自于同一来源的窗口和文档的属性，这里的同一来源指的是主机名、协议和端口号的组合

22.编写一个数组去重的方法  
function oSort(arr)  
{  
var result ={};  
var newArr=[];  
for(var i=0;i<arr.length;i++)  
{  
if(!result[arr])  
{  
newArr.push(arr)  
result[arr]=1  
}  
}  
return newArr  
}

23.截取字符串abcdefg的efg  
//alert('abcdefg'.substring(4));

24.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');

25.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();

25.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

6，javascript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();

26.FF下面实现outerHTML  
/\*var nDiv = document.createElement('div');  
var oDiv1 = document.getElementById('div1');  
var oWarp = document.getElementById('warp');  
  
oWarp.insertBefore(nDiv, oDiv1);  
nDiv.appendChild(oDiv1);  
var sOut = nDiv.innerHTML;  
oWarp.insertBefore(oDiv1, nDiv);  
oWarp.removeChild(nDiv);  
alert(sOut);

8，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;  
}

10，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\*/

12，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;

11、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;  
}\*/

4. 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

6. 你如何对网站的文件和资源进行优化？

1、文件合并（目的是减少http请求）：Web性能优化最佳实践中最重要的一条是减少HTTP 请求

减少HTTP请求的方案主要有合并JavaScript和CSS文件、[CSS Sprites](http://alistapart.com/articles/sprites" \t "_blank)、[图像映射 （Image Map）](http://www.w3.org/TR/html401/struct/objects.html" \l "h-13.6" \t "_blank)和[使用Data URI来编码图片](http://dancewithnet.com/2009/08/15/data-uri-mhtml/" \t "_blank)

**2.　　请正确理解 Repaint 和 Reflow**

　　注：Repaint 和 Reflow 也就是重绘和重排

**3.　　请减少对DOM的操作**

修改和访问DOM元素会造成页面的Repaint和Reflow，循环对DOM操作更是罪恶的行为。所以请合理的使用JavaScript变量储存内容，考虑大量DOM元素中循环的性能开销，在循环结束时一次性写入。

　　减少对DOM元素的查询和修改，查询时可将其赋值给局部变量。

　　注：在IE中:hover会降低响应速度。

**4.　　使用JSON格式来进行数据交换**

**5.　　高效使用HTML标签和CSS样式**

**7.　　将CSS和JS放到外部文件中引用，CSS放头，JS放尾**

**8.　　精简CSS和JS文件**

**9.　　压缩图片和使用图片Sprite技术**

**10.　　注意控制Cookie大小和污染**

7. 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

8.请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\*/

9.什么是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\*

10.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\*/

11.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

4.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\*/

5.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

9\* 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  
\*/

10.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\*/

12.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\*/

**1.javascript的typeof返回哪些数据类型**

Object number function boolean underfind

**2.例举3种强制类型转换和2种隐式类型转换？**

强制（parseInt,parseFloat,number）

隐式（== - ===）

**3.split() join() 的区别**

前者是切割成数组的形式，后者是将数组转换成字符串

**4.数组方法pop() push() unshift() shift()**

Push()尾部添加 pop()尾部删除

Unshift()头部添加  shift()头部删除

**6.IE和DOM事件流的区别**

1.执行顺序不一样、

2.参数不一样

3.事件加不加on

4.this指向问题

**7.IE和标准下有哪些兼容性的写法**

Var ev = ev || window.event

document.documentElement.clientWidth || document.body.clientWidth

Var target = ev.srcElement||ev.target

**8.ajax请求的时候get 和post方式的区别**

一个在url后面 一个放在虚拟载体里面

有大小限制

安全问题

应用不同 一个是论坛等只需要请求的，一个是类似修改密码的

**9.call和apply的区别**

Object.call(this,obj1,obj2,obj3)

Object.apply(this,arguments)

**10.ajax请求时，如何解释json数据**

使用eval parse 鉴于安全性考虑 使用parse更靠谱

**12.写一个获取非行间样式的函数**

1. function getStyle(obj,attr,value)
2. {
3. if(!value)
4. {
5. if(obj.currentStyle)
6. {
7. return obj.currentStyle(attr)
8. }
9. else
10. {
11. obj.getComputedStyle(attr,false)
12. }
13. }
14. else
15. {
16. obj.style[attr]=value
17. }
18. }

复制代码

**13.事件委托是什么**

让利用事件冒泡的原理，让自己的所触发的事件，让他的父元素代替执行！

**15.如何阻止事件冒泡和默认事件**

canceBubble   return false

**16.添加 删除 替换 插入到某个接点的方法**

obj.appendChidl()

obj.innersetBefore

obj.replaceChild

obj.removeChild

**17.解释jsonp的原理，以及为什么不是真正的ajax**

动态创建script标签，回调函数

Ajax是页面无刷新请求数据操作

**18.javascript的本地对象，内置对象和宿主对象**

本地对象为array obj regexp等可以new实例化

内置对象为gload Math 等不可以实例化的

宿主为浏览器自带的document,window 等

**19.document load 和document ready的区别**

Document.onload 是在结构和样式加载完才执行js

Document.ready原生种没有这个方法，jquery中有 $().ready(function)

**20.”==”和“===”的不同**

前者会自动转换类型

后者不会