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。  
  
3.Doctype? 严格模式与混杂模式-如何触发这两种模式，区分它们有何意义?   
用于声明文档使用那种规范（html/Xhtml）一般为 严格 过度 基于框架的html文档  
加入XMl声明可触发，解析方式更改为IE5.5 拥有IE5.5的bug  
  
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)  
  
10.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.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)  
  
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  
}

来自

HTML+CSS  
  
1.对WEB标准以及W3C的理解与认识？  
（1）web标准规范要求，书写标签必须闭合、标签小写、不乱嵌套，可提高搜索机器人对网页内容的搜索几率；  
（2）建议使用外链css和js脚本，从而达到结构与行为、结构与表现的分离，提高页面的渲染速度，能更快地显示页面的内容；  
（3）样式与标签的分离，更合理的语义化标签，使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， 从而降低维护成本、改版更方便；  
（4）不需要变动页面内容，便可提供打印版本而不需要复制内容，提高网站易用性；  
遵循w3c制定的web标准，能够使用户浏览者更方便的阅读，使网页开发者之间更好的交流。  
  
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。  
  
3.Doctype? 严格模式与混杂模式-如何触发这两种模式，区分它们有何意义?   
用于声明文档使用那种规范（html/Xhtml）一般为 严格 过度 基于框架的html文档  
加入XMl声明可触发，解析方式更改为IE5.5 拥有IE5.5的bug  
  
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)  
  
10.写出几种IE6 BUG的解决方法  
1.双边距BUG float引起的 使用display：inline；  
2.3像素问题 使用float引起的 使用dislpay:inline -3px 或者 margin-right:-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.标签上title与alt属性的区别是什么?  
alt属性是在你的图片因为某种原因不能加载时在页面显示的提示信息，它会直接输出在原本加载图片的地方；  
title属性是在你鼠标悬停在该图片上时显示一个小提示，鼠标离开就没有了，有点类似jQuery的hover，你可以自己试试，另外，HTML的绝大多数标签都支持title属性，title属性就是专门做提示信息的；  
  
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。例如（有最简单的\*{margin:0 ;  padding:0}）；  
  
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?  
所谓的标准模式是指，浏览器按W3C标准解析执行代码；  
怪异模式则是使用浏览器自己的方式解析执行代码，因为不同浏览器解析执行的方式不一样，所以我们称之为怪异模式。  
浏览器解析时到底使用标准模式还是怪异模式，与你网页中的DTD声明直接相关，DTD声明定义了标准文档的类型（标准模式解析）文档类型，会使浏览器使用相应的方式加载网页并显示，忽略DTD声明,将使网页进入怪异模式(quirks mode)。  
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?  
语义化的HTML就是写出的HTML代码，符合内容的结构化（内容语义化），选择合适的标签（代码语义化），能够便于开发者阅读和写出更优雅的代码的同时让浏览器的爬虫和机器很好地解析。  
　　1.语义化有利于SEO，有利于搜索引擎爬虫更好的理解我们的网页，从而获取更多的有效信息，提升网页的权重。  
　　2.在没有CSS的时候能够清晰的看出网页的结构，增强可读性。  
　　3.便于团队开发和维护，语义化的HTML可以让开发者更容易的看明白，从而提高团队的效率和协调能力。  
　　4.支持多终端设备的浏览器渲染。  
（总结：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)  
  
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  
split()方法：用于把一个字符串分割成字符串数组.   
join() 方法用于把数组中的所有元素放入一个字符串。  
（总结：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？  
事件绑定就是针对dom元素的事件，绑定在dom元素上  
普通事件即为非针对dom元素的事件；  
例如：  
普通事件  
            var btn = document.getElementById("hello");  
            btn.onclick = function(){  
                alert(1);  
            };  
            btn.onclick = function(){  
                alert(2);  
            };                                                //这个事件只会弹出2；  
  
事件绑定  
            var btn = document.getElementById("hello");  
            btn.addEventListener("click",function(){  
                alert(1);  
            },false);  
            btn.addEventListener("click",function(){  
                alert(2);  
            },false);                                    //这个事件首先会弹出1，然后在弹出2；  
  
  
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？  
1、get是把参数数据队列加到提交表单的ACTION属性所指的URL中，值和表单内各个字段一一对应，在URL中可以看到；  
      post是通过HTTP post机制，将表单内各个字段与其内容放置在HTML HEADER内一起传送到ACTION属性所指的URL地址。用户看不到这个过程；  
2、Get请求有如下特性：它会将数据添加到URL中，通过这种方式传递到服务器，通常利用一个问号？代表URL地址的结尾与数据参数的开端，后面的参数每一个数据参数以“名称=值”的形式出现，参数与参数之间利用一个连接符&来区分。  
      Post请求有如下特性：数据是放在HTTP主体中的，其组织方式不只一种，有&连接方式，也有分割符方式，可隐藏参数，传递大批数据，比较方便。  
3、get传送的数据量较小，不能大于2KB；  
      post传送的数据量较大，一般被默认为不受限制。但理论上，因服务器的不同而异.  
4、get安全性非常低，post安全性较高；  
（总结：  
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？  
相同点:两个方法产生的作用是完全一样的  
不同点:方法传递的参数不同  
Object.call(this,obj1,obj2,obj3)调用一个对象的一个方法，以另一个对象替换当前对象  
Object.apply(this,arguments)应用某一对象的一个方法，用另一个对象替换当前对象。   
  
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？  
b.prototype=new a;  
  
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！  
事件委托就是事件目标自身不处理事件，而是把处理任务委托给其父元素或者祖先元素，甚至根元素（document）；  
jQuery为绑定和委托事件提供了.bind()、.live()和.delegate()方法；  
http://www.tuicool.com/articles/zQVvau  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  
查看  http://www.2cto.com/kf/201412/359961.html  案例  
  
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  
}

<http://www.cnblogs.com/bigboyLin/p/5272902.html>

<http://blog.csdn.net/qq_34753236/article/details/51306259>

http://web.jobbole.com/85340/