1：Html文件顶部DOCTYPE的作用是什么？加它和不加它有什么区别：

<!DOCTYPE>声明位于HTML文档的第一行，处于《html》之前，目的是为了告诉浏览器用什么文旦标准解析这个文档（避免浏览器的怪异行为），DOCTYPE不存在或者格式不正确会导致文档以兼容模式呈现

标准模式的排版 和JS运作模式都是以该浏览器支持的最高标准运行。在兼容模式中，页面以宽松的向后兼容的方式显示,模拟老式浏览器的行为以防止站点无法工作。

2：介绍一下你对浏览器内核的理解？

主要分成两部分：渲染引擎(layout engineer或Rendering Engine)和JS引擎。  
渲染引擎：负责取得网页的内容（HTML、XML、图像等等）、整理讯息（例如加入CSS等），以及计算网页的显示方式，然后会输出至显示器或打印机。浏览器的内核的不同对于网页的语法解释会有不同，所以渲染的效果也不相同。所有网页浏览器、电子邮件客户端以及其它需要编辑、显示网络内容的应用程序都需要内核。  
JS引擎则：解析和执行javascript来实现网页的动态效果。

最开始渲染引擎和JS引擎并没有区分的很明确，后来JS引擎越来越独立，内核就倾向于只指渲染引擎。

常见的浏览器内核有哪些？  
Trident内核：IE,MaxThon,TT,The World,360,搜狗浏览器等。[又称MSHTML]  
Gecko内核：Netscape6及以上版本，FF,MozillaSuite/SeaMonkey等  
Presto内核：Opera7及以上。 [Opera内核原为：Presto，现为：Blink;]  
Webkit内核：Safari,Chrome等。 [ Chrome的：Blink（WebKit的分支）]

3：说一下从浏览器输入地址到后到显示HTML内容都经历了哪些过程。

首先浏览器会检查是否存在缓存（浏览器，电脑系统，路由）

如果没有则进行域名分析(DNS解析，将域名和ip的映射搭建起来)

浏览器会和服务器发起tcp三次招手。捂手成功，发起http请求，请求数据包

服务器收到请求，返回数据包。

浏览器收到请求的数据，开始渲染解析，解析HTML代码，生成DOM树，解析css，js。

4：为了保证HTML内容第一时间渲染出来，script标签放在什么位置最佳：

Body底部或者是html底部

5：浏览器加载script时是一个个加载完成还是一次同时加载多个，你觉得吧多个js压缩成一个好吗？

老的浏览器是一个加载完成才开始加载下一个，在新的浏览器中，加载是并行的，但是执行顺序是按照scripts的顺序来的。我觉的自己写的js代码可以压缩成一个js，但是不建议也吧插件带的js一起压缩，因为它们之间有很强的依赖。

6：映入标签import 和link的区别：

Link标签是html元素不存在兼容性：@import存在低版本的浏览器兼容问题

Link加载的文件和页面同时加载，@import在页面加载完成才开始加载

Link是html元素支持通过JavaScript控制样式 @import不支持

Link可以用来定义Rss等其他作用，@import只能用来加载css

7:页面乱码：

首先检查是页面字体乱码还是ajax请求乱码

网页源代码是GBK的编码，而内容中的中文文字是UTF-8编码的，这样会造成打开编码错误。

第二种：网页设置的是编码是GBK，而数据库存储的是utf-8，这个时候查询的时候将它专门转码

第三种：浏览器造成的乱码：可能是你网页没有设置meta chatset造成的，可以找到浏览器的编码方式。

Charset UTF-8 GBK GB2312

8：http和https的区别：

https = http + ssl

htttp在http应用层的基础上使用安全套接子层作为子层

超文本传输协议(HTTP)是一个用来通过互联网传输和接收信息的协议，HTTP使用请求和响应的过程，因此信息可以在服务器间快速，轻松而且精确的进行传输。当你访问web页面的时候你就是在使用HTTP协议，但是HTTP协议是不安全的，可以轻松对窃听你跟web服务器之间的数据交互和传输。在很多情况下，客户和服务器之间传输的是敏感信息，需要防止未经授权的访问。网景公司推出了HTTPS，也是基于安全套接字层的HTTP协议

http和https的区别：

1：http和https的url开头不同；

2：http是没有安全性的https的安全性高

3：http的标准端口是80,https的标准端口是443

4：在osi网络模型中。http工作于应用层，而https工作于传输层

5：http无需加密，而https对传输的数据需要加密。

6：http无需证书，而https需要认证的证书

9：如何设置浏览器对资源文件的缓存？cache-control max-age和expire的区别：如何设置不缓存文件；

HTML5的离线缓存：在用户没有与因特网链接的时候，可以正常访问站点和应用，在用户与因特网链接的时候，更新用户机器上的文件。原因：HTML5离线缓存是基于一个新建的。Appcache文件的缓存机制（不是存储机制），通过这个文件的解析清单离线存储资源，这些资源就像是cookie一样被存储下来，之后当网络在处于离线状态之下时，浏览器会通过被离线存储的数据进行页面展示。

如何设置： 在html标签上加入manifest属性，设置离线缓存的文件

<!DOCTYPE HTML>

<html manifest="demo.appcache">

<body>

The content of the document......

</body>

</html>

每个指定manifest的页面在用户对其访问时都会被缓存，如果未指定manifset属性，则页面不会被缓存；

Manifest文件的建议的文件扩展名是.appcache。

Manifest文件需要正确的配置MIME-TYPE 即text/cache-manifest .必须在web服务器上进行配置。

manifest 文件是简单的文本文件，它告知浏览器被缓存的内容

manifest文件可以分为三类：

CACHE MANIFEST –在此标题下出现的文件在首次下载之后被缓存

NETWORK 在此标题下列出的文件需要与服务器的链接，不会被缓存

FALLBACK 在此标题下列出的文件在页面无法访问的时候的回退页面

CACHE MANIFEST

/theme.css

/logo.gif

/main.js

一旦应用被缓存，它就会保持缓存直到发生下列情况：

* 用户清空浏览器缓存
* manifest 文件被修改（参阅下面的提示）
* 由程序来更新应用缓存

更加使用于移动端项目

10：localStorage与sessionStorage和cookie的区别？讲下他们的生命周期？

Cookie是网站为了识别用户身份而存储在用户本地终端上的数据

Cookie始终会被同源的http请求携带（即使不需要）也会在浏览器与服务端之间传递

Session和localStorage不会自动把数据发送给服务器，仅在本地保存；

Cookie一把是有作用时间的（人为的设置生命周期）

Session在当前页面没有关闭之前存在

Localstorage永久存在，除非用户手动删除

Cookie大小限制在4KB Storage限制在5M

11：http返回值是200（成功）

304 HTTP 304: Not Modified

服务器告诉客户端 原来缓存的文档还是可以使用的，一般是提供If-Modified-Since头表示客户只想比指定日期更新的文档

cache-control：no-chache说明强制每次请求直接发送给源服务器，而不经过本地缓存版本的校验

1：max-age>0 时 直接从游览器缓存中 提取   
2、max-age<=0 时 向server 发送http 请求确认 ,该资源是否有修改有的话 返回200 ,无的话 返回304.

404(找不到相应的url)

401（相应的参数错误）

500(后端问题)

12：跨域的几种请求方式：最常用的那种：

JSONP（动态的插入script标签，回调函数执行，参数通过回调函数的形参传递），window.name/ifrma window。Name通过同一个窗口公用一个.nmae属性来传递参数（必须在同一个窗口之内跳转）