# Web前端方面

**1、CSS引入的方式有哪些?**

1）外联：<link>标签

2）内联：<style>标签

3）元素内嵌：元素的style属性

**2、CSS选择符有哪些?**

标签选择符、类选择符、id选择符、组合选择符

**3、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)

**4、”==”和“===”的不同**

==只比较值，===比较值和数据类型

**5、什么是盒子模型？**

在网页中，一个元素占有空间的大小由几个部分构成，其中包括元素的内容（content），元素的内边距（padding），元素的边框（border），元素的外边距（margin）四个部分。这四个部分占有的空间中，有的部分可以显示相应的内容，而有的部分只用来分隔相邻的区域或区域。4个部分一起构成了css中元素的盒模型。

**6、$(this) 和 this 关键字在 jQuery 中有何不同？**

前者是jQuery对象，使用jQuery方法和属性；后者是JavaScript对象，使用JavaScript方法和属性。

**7、jQuery 里的 each() 是什么函数？你是如何使用它的？**

each() 函数就像是 Java 里的一个 Iterator，它允许你遍历一个元素集合。你可以传一个函数给 each() 方法，被调用的 jQuery 对象会在其每个元素上执行传入的函数。

**8、$(document).ready() 是个什么函数？为什么要用它？**

ready() 函数用于在文档进入ready状态时执行代码。当DOM 完全加载（例如HTML被完全解析DOM树构建完成时），jQuery允许你执行代码。

**9、网页上有 5 个 <div> 元素，如何使用 jQuery来选择它们？**

jQuery 代码：$("div")，这样会返回一个包含所有 5 个 div 标签的 jQuery 对象。

**10、列举常用的几种jQuery选择器**

1）基本选择器：包括id选择器、class选择器和元素选择器。

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：如属性选择器、表单选择器等等

**11、AJAX应用和传统Web应用有什么不同？**

在传统的Javascript编程中，如果想得到服务器端数据库或文件上的信息，或者发送客户端信息到服务器，需要建立一个HTML form然后GET或者POST数据到服务器端。用户需要点击”Submit”按钮来发送或者接受数据信息，然后等待服务器响应请求，页面重新加载。因为服务器每次都会返回一个新的页面， 所以传统的web应用有可能很慢而且用户交互不友好。使用AJAX技术， 就可以使Javascript通过XMLHttpRequest对象直接与服务器进行交互。通过HTTP Request， 一个web页面可以发送一个请求到web服务器并且接受web服务器返回的信息(不用重新加载页面)，展示给用户的还是同一个页面，用户感觉不到页面刷新，也看不到到Javascript后台进行的发送请求和接受响应。

**12、AJAX的全称是什么？ 介绍一下AJAX**

AJAX的全称是Asynchronous JavaScript And XML.。AJAX是2005年由Google发起并流行起来的编程方法， AJAX不是一个新的编程语言，但是它是一个使用已有标准的新的编程技术。使用AJAX可以创建更好，更快，更用户界面友好的Web应用。AJAX技术基于Javascript和HTTP Request。

**13、AJAX的优缺点都有什么？**

优点：

1）最大的一点是页面无刷新，用户的体验非常好。

2）使用异步方式与服务器通信，具有更加迅速的响应能力。

3）可以把以前一些服务器负担的工作转嫁到客户端，利用客户端闲置的能力来处理，减轻服务器和带宽的负担，节约空间和宽带租用成本。并且减轻服务器的负担，ajax的原则是“按需取数据”，可以最大程度的减少冗余请求，和响应对服务器造成的负担。

4）基于标准化的并被广泛支持的技术，不需要下载插件或者小程序。

缺点：

1）不支持浏览器back按钮。

2）安全问题 AJAX暴露了与服务器交互的细节。

3）对搜索引擎的支持比较弱。

4）破坏了程序的异常机制。

5）不容易调试。

**14、简述Ajax的工作原理？**

Ajax的核心是JavaScript对象XmlHttpRequest。该对象在Internet Explorer 5中首次引入，它是一种支持异步请求的技术。简而言之，XmlHttpRequest使您可以使用JavaScript向服务器提出请求并处理响应，而不阻塞用户。在创建Web站点时，在客户端执行屏幕更新为用户提供了很大的灵活性。

**15、请介绍一下XmlHttpRequest对象**

Ajax的核心是JavaScript对象XmlHttpRequest。该对象在Internet Explorer 5中首次引入，它是一种支持异步请求的技术。简而言之，XmlHttpRequest使您可以使用JavaScript向服务器提出请求并处理响应，而不阻塞用户。通过XMLHttpRequest对象，Web开发人员可以在页面加载以后进行页面的局部更新。

# XML方面

**1、什么是XML**

XML(eXtensible Markup Language)是可扩展标记语言，能够用一系列简单的标记描述数据。

**2、XML的解析方式**

常用的用dom解析和sax解析。dom解析是一次性读取xml文件并将其构造为DOM对象供程序使用，优点是操作方便，但是比较耗内存。Sax是按事件驱动的方式解析的，占用内存少，但是编程复杂。

# JSP方面

**1、forward 和redirect的区别**

forward是服务器请求资源，服务器直接访问目标地址的URL，把那个URL的响应内容读取过来，然后把这些内容再发给浏览器，其实客户端浏览器只发了一次请求，所以它的地址栏中还是原来的地址，session,request参数都可以获取。

redirect就是服务端根据逻辑,发送一个状态码,告诉浏览器重新去请求那个地址，相当于客户端浏览器发送了两次请求。

**2、JSP有哪些内置对象?作用分别是什么?**

JSP共有以下9种基本内置组件（可与ASP的6种内部组件相对应）：

　 request 用户端请求，此请求会包含来自GET/POST请求的参数

response 网页传回用户端的回应

pageContext 网页的属性是在这里管理

session 与请求有关的会话期

application servlet 正在执行的内容

out 用来传送回应的输出

config servlet的构架部件

page JSP网页本身

exception 针对错误网页，未捕捉的例外

**3、列举JSP有哪些动作?作用分别是什么?**

jsp:include：在页面被请求的时候引入一个文件。

jsp:useBean：寻找或者实例化一个JavaBean。

jsp:setProperty：设置JavaBean的属性。

jsp:getProperty：输出某个JavaBean的属性。

jsp:forward：把请求转到一个新的页面。

**4、JSP中动态INCLUDE与静态INCLUDE的区别？**

动态INCLUDE用jsp:include动作实现

<jsp:include page="included.jsp" flush="true" />它总是会检查所含文件中的变化，适合用于包含动态页面，并且可以带参数

静态INCLUDE用include伪码实现,定不会检查所含文件的变化，适用于包含静态页面

<%@ include file="included.htm" %>

**5、两种跳转方式分别是什么?有什么区别?**

有两种，分别为：

<jsp:include page="included.jsp" flush="true">

<jsp:forward page= "nextpage.jsp"/>

前者页面不会转向include所指的页面，只是显示该页的结果，主页面还是原来的页面。执行完后还会回来，相当于函数调用。并且可以带参数.后者完全转向新页面，不会再回来。相当于goto 语句。

# Servlet、JSP方面

**1、说一说Servlet的生命周期?**

Servlet有良好的生存期的定义，包括加载和实例化、初始化、处理请求以及服务结束。这个生存期由javax.servlet.Servlet接口的init,service和destroy方法表达。 Servlet被服务器实例化后，容器运行其init方法，请求到达时运行其service方法，service方法自动派遣运行与请求对应的doXXX方法（doGet，doPost）等，当服务器决定将实例销毁的时候调用其destroy方法。

与cgi的区别在于Servlet处于服务器进程中，它通过多线程方式运行其service方法，一个实例可以服务于多个请求，并且其实例一般不会销毁，而CGI对每个请求都产生新的进程，服务完成后就销毁，所以效率上低于Servlet。

**2、JAVA SERVLET API中forward() 与redirect()的区别？**

forward是服务器请求资源，服务器直接访问目标地址的URL，把那个URL的响应内容读取过来，然后把这些内容再发给浏览器，其实客户端浏览器只发了一次请求，所以它的地址栏中还是原来的地址，session,request参数都可以获取。

redirect就是服务端根据逻辑,发送一个状态码,告诉浏览器重新去请求那个地址，相当于客户端浏览器发送了两次请求。

**3、Servlet的基本架构**

public class ServletName extends HttpServlet {

public void doPost(HttpServletRequest request, HttpServletResponse response) throws

ServletException, IOException {

}

public void doGet(HttpServletRequest request, HttpServletResponse response) throws

ServletException, IOException {

}

}

**4、什么情况下调用doGet()和doPost()？**

JSP页面中的form标签里的method属性为get时调用doGet()，为post时调用doPost()；超链接跳转页面时调用doGet()

**5、servlet的生命周期**

web容器加载servlet，生命周期开始。通过调用servlet的init()方法进行servlet的初始化。通过调用service()方法实现，根据请求的不同调用不同的do\*\*\*()方法。结束服务，web容器调用servlet的destroy()方法。

**6、页面间对象传递的方法**

request，session，application，cookie等

**7、JSP和Servlet有哪些相同点和不同点，他们之间的联系是什么？**

JSP是Servlet技术的扩展，本质上是Servlet的简易方式，更强调应用的外表表达。JSP编译后是"类servlet"。Servlet和JSP最主要的不同点在于，Servlet的应用逻辑是在Java文件中，并且完全从表示层中的HTML里分离开来。而JSP的情况是Java和HTML可以组合成一个扩展名为.jsp的文件。JSP侧重于视图，Servlet主要用于控制逻辑。

**8、四种会话跟踪技术**

会话作用域ServletsJSP 页面描述

1）page否是代表与一个页面相关的对象和属性。一个页面由一个编译好的 Java servlet 类（可以带有任何的 include 指令，但是没有 include 动作）表示。这既包括 servlet 又包括被编译成 servlet 的 JSP 页面

2）request是是代表与 Web 客户机发出的一个请求相关的对象和属性。一个请求可能跨越多个页面，涉及多个 Web 组件（由于 forward 指令和 include 动作的关系）

3）session是是代表与用于某个 Web 客户机的一个用户体验相关的对象和属性。一个 Web 会话可以也经常会跨越多个客户机请求

4）application是是代表与整个 Web 应用程序相关的对象和属性。这实质上是跨越整个 Web 应用程序，包括多个页面、请求和会话的一个全局作用域

**9、Request对象的主要方法**

setAttribute(String name,Object)：设置名字为name的request的参数值

getAttribute(String name)：返回由name指定的属性值

getAttributeNames()：返回request对象所有属性的名字集合，结果是一个枚举的实例

getCookies()：返回客户端的所有Cookie对象，结果是一个Cookie数组

getCharacterEncoding()：返回请求中的字符编码方式

getContentLength()：返回请求的Body的长度

getHeader(String name)：获得HTTP协议定义的文件头信息

getHeaders(String name)：返回指定名字的request Header的所有值，结果是一个枚举的实例

getHeaderNames()：返回所以request Header的名字，结果是一个枚举的实例

getInputStream()：返回请求的输入流，用于获得请求中的数据

getMethod()：获得客户端向服务器端传送数据的方法

getParameter(String name)：获得客户端传送给服务器端的有name指定的参数值

getParameterNames()：获得客户端传送给服务器端的所有参数的名字，结果是一个枚举的实例

getParameterValues(String name)：获得有name指定的参数的所有值

getProtocol()：获取客户端向服务器端传送数据所依据的协议名称

getQueryString()：获得查询字符串

getRequestURI()：获取发出请求字符串的客户端地址

getRemoteAddr()：获取客户端的IP地址

getRemoteHost()：获取客户端的名字

getSession([Boolean create])：返回和请求相关Session

getServerName()：获取服务器的名字

getServletPath()：获取客户端所请求的脚本文件的路径

getServerPort()：获取服务器的端口号

removeAttribute(String name)：删除请求中的一个属性

**10、我们在web应用开发过程中经常遇到输出某种编码的字符，如iso8859-1等，如何输出一个某种编码的字符串？**

Public String translate (String str) {

String tempStr = "";

try {

tempStr = new String(str.getBytes("ISO-8859-1"), "GBK");

tempStr = tempStr.trim();

}

catch (Exception e) {

System.err.println(e.getMessage());

}

return tempStr;

}

**11、Servlet执行时一般实现哪几个方法？**

public void init(ServletConfig config)

public ServletConfig getServletConfig()

public String getServletInfo()

public void service(ServletRequest request,ServletResponse response)

public void destroy()

**12、描述Cookie和Session的作用，区别和各自的应用范围，Session工作原理。**

1）cookie 是一种发送到客户浏览器的文本串句柄，并保存在客户机硬盘上，可以用来在某个WEB站点会话间持久的保持数据。

2）session其实指的就是访问者从到达某个特定主页到离开为止的那段时间。 Session其实是利用Cookie进行信息处理的，当用户首先进行了请求后，服务端就在用户浏览器上创建了一个Cookie，当这个Session结束时，其实就是意味着这个Cookie就过期了。

注：为这个用户创建的Cookie的名称是aspsessionid。这个Cookie的唯一目的就是为每一个用户提供不同的身份认证。

3）cookie和session的共同之处在于：cookie和session都是用来跟踪浏览器用户身份的会话方式。

4）cookie 和session的区别是：cookie数据保存在客户端，session数据保存在服务器端。

5）session工作原理：session技术中所有的数据都保存在服务器上，客户端每次请求服务器的时候会发送当前会话的sessionid，服务器根据当前sessionid判断相应的用户数据标志，以确定用户是否登录或具有某种权限。

**13、什么是Servlet？**

Servlet是用来处理客户端请求并产生动态网页内容的Java类。Servlet主要是用来处理或者是存储HTML表单提交的数据，产生动态内容，在无状态的HTTP协议下管理状态信息。

**14、说一下Servlet的体系结构。**

所有的Servlet都必须要实现的核心的接口是javax.servlet.Servlet。每一个Servlet都必须要直接或者是间接实现这个接口，或者是继承javax.servlet.GenericServlet或者javax.servlet.http.HTTPServlet。最后，Servlet使用多线程可以并行的为多个请求服务。

**15、Applet和Servlet有什么区别？**

Applet是运行在客户端主机的浏览器上的客户端Java程序。而Servlet是运行在web服务器上的服务端的组件。applet可以使用用户界面类，而Servlet没有用户界面，相反，Servlet是等待客户端的HTTP请求，然后为请求产生响应。

**16、GenericServlet和HttpServlet有什么区别？**

GenericServlet是一个通用的协议无关的Servlet，它实现了Servlet和ServletConfig接口。继承自GenericServlet的Servlet应该要覆盖service()方法。最后，为了开发一个能用在网页上服务于使用HTTP协议请求的Servlet，你的Servlet必须要继承自HttpServlet。这里有Servlet的例子。

**17、解释下Servlet的生命周期。**

对每一个客户端的请求，Servlet引擎载入Servlet，调用它的init()方法，完成Servlet的初始化。然后，Servlet对象通过为每一个请求单独调用service()方法来处理所有随后来自客户端的请求，最后，调用Servlet(译者注：这里应该是Servlet而不是server)的destroy()方法把Servlet删除掉。

**18、doGet()方法和doPost()方法有什么区别？**

doGet：GET方法会把名值对追加在请求的URL后面。因为URL对字符数目有限制，进而限制了用在客户端请求的参数值的数目。并且请求中的参数值是可见的，因此，敏感信息不能用这种方式传递。

doPOST：POST方法通过把请求参数值放在请求体中来克服GET方法的限制，因此，可以发送的参数的数目是没有限制的。最后，通过POST请求传递的敏感信息对外部客户端是不可见的。

**19、什么是Web应用程序？**

Web应用程序是对Web或者是应用服务器的动态扩展。有两种类型的Web应用：面向表现的和面向服务的。面向表现的Web应用程序会产生包含了很多种标记语言和动态内容的交互的web页面作为对请求的响应。而面向服务的Web应用实现了Web服务的端点(endpoint)。一般来说，一个Web应用可以看成是一组安装在服务器URL名称空间的特定子集下面的Servlet的集合。

**20、什么是服务端包含(Server Side Include)？**

服务端包含(SSI)是一种简单的解释型服务端脚本语言，大多数时候仅用在Web上，用servlet标签嵌入进来。SSI最常用的场景把一个或多个文件包含到Web服务器的一个Web页面中。当浏览器访问Web页面的时候，Web服务器会用对应的servlet产生的文本来替换Web页面中的servlet标签。

**21、什么是Servlet链(Servlet Chaining)？**

Servlet链是把一个Servlet的输出发送给另一个Servlet的方法。第二个Servlet的输出可以发送给第三个Servlet，依次类推。链条上最后一个Servlet负责把响应发送给客户端。

**22、如何知道是哪一个客户端的机器正在请求你的Servlet？**

ServletRequest类可以找出客户端机器的IP地址或者是主机名。getRemoteAddr()方法获取客户端主机的IP地址，getRemoteHost()可以获取主机名。看下这里的例子。

**23、HTTP响应的结构是怎么样的？**

HTTP响应由三个部分组成：

状态码(Status Code)：描述了响应的状态。可以用来检查是否成功的完成了请求。请求失败的情况下，状态码可用来找出失败的原因。如果Servlet没有返回状态码，默认会返回成功的状态码HttpServletResponse.SC\_OK。

HTTP头部(HTTP Header)：它们包含了更多关于响应的信息。比如：头部可以指定认为响应过期的过期日期，或者是指定用来给用户安全的传输实体内容的编码格式。如何在Serlet中检索HTTP的头部看这里。

主体(Body)：它包含了响应的内容。它可以包含HTML代码，图片，等等。主体是由传输在HTTP消息中紧跟在头部后面的数据字节组成的。

**24、什么是cookie？session和cookie有什么区别？**

cookie是Web服务器发送给浏览器的一块信息。浏览器会在本地文件中给每一个Web服务器存储cookie。以后浏览器在给特定的Web服务器发请求的时候，同时会发送所有为该服务器存储的cookie。下面列出了session和cookie的区别：

• 无论客户端浏览器做怎么样的设置，session都应该能正常工作。客户端可以选择禁用cookie，但是，session仍然是能够工作的，因为客户端无法禁用服务端的session。

• 在存储的数据量方面session和cookies也是不一样的。session能够存储任意的Java对象，cookie只能存储String类型的对象。

**25、浏览器和Servlet通信使用的是什么协议？**

浏览器和Servlet通信使用的是HTTP协议。

**26、什么是HTTP隧道？**

HTTP隧道是一种利用HTTP或者是HTTPS把多种网络协议封装起来进行通信的技术。因此，HTTP协议扮演了一个打通用于通信的网络协议的管道的包装器的角色。把其他协议的请求掩盖成HTTP的请求就是HTTP隧道。

**27、sendRedirect()和forward()方法有什么区别？**

sendRedirect()方法会创建一个新的请求，而forward()方法只是把请求转发到一个新的目标上。重定向(redirect)以后，之前请求作用域范围以内的对象就失效了，因为会产生一个新的请求，而转发(forwarding)以后，之前请求作用域范围以内的对象还是能访问的。一般认为sendRedirect()比forward()要慢。

**28、什么是URL编码和URL解码？**

URL编码是负责把URL里面的空格和其他的特殊字符替换成对应的十六进制表示，反之就是解码。

JSP

**29、什么是JSP页面？**

JSP页面是一种包含了静态数据和JSP元素两种类型的文本的文本文档。静态数据可以用任何基于文本的格式来表示，比如：HTML或者XML。JSP是一种混合了静态内容和动态产生的内容的技术。这里看下JSP的例子。

**30、JSP请求是如何被处理的？**

浏览器首先要请求一个以.jsp扩展名结尾的页面，发起JSP请求，然后，Web服务器读取这个请求，使用JSP编译器把JSP页面转化成一个Servlet类。需要注意的是，只有当第一次请求页面或者是JSP文件发生改变的时候JSP文件才会被编译，然后服务器调用servlet类，处理浏览器的请求。一旦请求执行结束，servlet会把响应发送给客户端。这里看下如何在JSP中获取请求参数。

**31、JSP有什么优点？**

下面列出了使用JSP的优点：

• JSP页面是被动态编译成Servlet的，因此，开发者可以很容易的更新展现代码。

• JSP页面可以被预编译。

• JSP页面可以很容易的和静态模板结合，包括：HTML或者XML，也可以很容易的和产生动态内容的代码结合起来。

• 开发者可以提供让页面设计者以类XML格式来访问的自定义的JSP标签库。

• 开发者可以在组件层做逻辑上的改变，而不需要编辑单独使用了应用层逻辑的页面。

**32、什么是JSP指令(Directive)？JSP中有哪些不同类型的指令？**

Directive是当JSP页面被编译成Servlet的时候，JSP引擎要处理的指令。Directive用来设置页面级别的指令，从外部文件插入数据，指定自定义的标签库。Directive是定义在<%@ 和 %>之间的。下面列出了不同类型的Directive：

• 包含指令(Include directive)：用来包含文件和合并文件内容到当前的页面。

• 页面指令(Page directive)：用来定义JSP页面中特定的属性，比如错误页面和缓冲区。

• Taglib指令： 用来声明页面中使用的自定义的标签库。

**33、什么是JSP动作(JSP action)？**

JSP动作以XML语法的结构来控制Servlet引擎的行为。当JSP页面被请求的时候，JSP动作会被执行。它们可以被动态的插入到文件中，重用JavaBean组件，转发用户到其他的页面，或者是给Java插件产生HTML代码。下面列出了可用的动作：

• jsp:include-当JSP页面被请求的时候包含一个文件。

• jsp:useBean-找出或者是初始化Javabean。

• jsp:setProperty-设置JavaBean的属性。

• jsp:getProperty-获取JavaBean的属性。

• jsp:forward-把请求转发到新的页面。

• jsp:plugin-产生特定浏览器的代码。

**34、什么是Scriptlets？**

JSP技术中，scriptlet是嵌入在JSP页面中的一段Java代码。scriptlet是位于标签内部的所有的东西，在标签与标签之间，用户可以添加任意有效的scriplet。

**35、声明(Decalaration)在哪里？**

声明跟Java中的变量声明很相似，它用来声明随后要被表达式或者scriptlet使用的变量。添加的声明必须要用开始和结束标签包起来。

**36、什么是表达式(Expression)？**

【列表很长，可以分上、中、下发布】

JSP表达式是Web服务器把脚本语言表达式的值转化成一个String对象，插入到返回给客户端的数据流中。表达式是在<%=和%>这两个标签之间定义的。

**37、隐含对象是什么意思？有哪些隐含对象？**

JSP隐含对象是页面中的一些Java对象，JSP容器让这些Java对象可以为开发者所使用。开发者不用明确的声明就可以直接使用他们。JSP隐含对象也叫做预定义变量。下面列出了JSP页面中的隐含对象：

• application

• page

• request

• response

• session

• exception

• out

• config

• pageContext