1、HTTP三点注意事项

无连接：每次连接只能处理一个请求

无状态：对之前的处理对象没有记忆功能

媒体独立：能够传输任意类型的数据

2、什么是servlet

运行在服务器上的java程序

3、servlet主要功能

读取客户端提交的数据

读取浏览器发送的隐式数据

处理数据得到结果

向客户端发送处理好的显示数据

发送隐式的响应HTTP数据

4、servlet工作原理

当客户端向服务器发送请求时，服务器容器会创建请求响应对象，来判断请求的servlet是否存在。

如果存在，则直接调用service函数，并将请求-响应对象作为参数返回给客户端。

如果不存在，则需要加载和实例化servlet类，并且通过init方法初始化servlet类，再调用service方法。

service方法就是通过请求对象获取客户端提交的数据并进行处理，通过响应对象向客户端发送数据。

5、request

请求行、请求头、请求消息体

6、response

响应头、响应状态码，实体数据

7、请求转发和重定向区别

重定向不共享页面跳转的数据，请求转发能够共享页面跳转的数据。

重定向地址栏发生改变，请求转发地址栏不发生改变。

重定向的页面地址可以任意，请求转发只能转发同一服务器下的资源。

8、会话跟踪技术

URL重写：将会话ID写在URL地址结尾

隐藏表单域：将会话ID放在HTML的隐藏域表单中

cookie

session

9、cookie特点

cookie的数据存储在客户端服务器上。

cookie数据只能是非中文的字符串类型。

cookie数据的大小限制为4kb。

浏览器可以保存多个cookie数据，浏览器一般只允许存放300个左右cookie，且每个站点最多存放20个cookie。

10、session特点

session数据存储在服务器端。

session可以保存中文字符串。

session可以保存对象。

session数据可以超过4kb。

11、session结束的标志

用户关闭浏览器。

调用httpsession的invalidate（）函数

用户两次访问服务器的时间间隔大于session最大非活动时间间隔

12、设置session最大非活动时间间隔

<session-config>

<session-timeout>100</session-timeout>

</session-config>

13、servlet生命周期

加载和实例化servlet类

通过init函数初始化servlet实例

通过service函数对客户端请求进行处理

通过destroy函数销毁servlet实例，服务结束

14、jsp原理

服务器上的jsp引擎将jsp文件翻译成java文件

jsp引擎调用java编译器将java文件编译成编译文件

通过java虚拟机运行编译文件

15、jsp与servlet区别

jsp和servlet实质上都是servlet，只不过jsp侧重于视图，servlet侧重于控制逻辑。

servlet没有内置对象，jsp的对象必须通过httpservlet，httpservletrequest，httpservletresponse对象实现。

16、jsp元素

指令、动作、脚本

17、jsp指令

<%@page %>、<%@incule %>、<%@taglib %>

18、jsp动作

<jsp:useBean id= class= scope=>、<jsp:setproperty name= property= value=>、<jsp:getproperty name= property=>、<jsp:include>、<jsp:forward>

19、jsp对象

page，pageContext

request，response，session，application

config，exception，out

20、动态include和静态include

动态include会随着所包含资源文件的改变而改变，<jsp:include page="page.jsp" flush=true>

21、include和forward区别

include只是引用所包含文件，显示include的结果，页面本身不进行跳转。

forward完全跳转到新页面。

22、jdbc任务

建立与数据库的连接（建立连接）

发送对数据库的查询请求（发送请求）

处理数据库返回的查询结果（处理结果）

23、常见的jdbc组件

statement、driver，drivermanager、sqlexception，connection、resultset

24、jdbc程序实现

import java.sql.\*;

public void demo() throws exception(){

Class.forName("com.mysql.jdbc.driver");//加载驱动程序

String url=“jdbc:mysql：//localhost:3306/db"；//建立与数据库的连接

Connection con=DriverManager.getConnection(url,"root","root");

Statement st=con.createStatement();//建立执行sql语句实例

ResultSet rs=st.executeQuery();//获取查询结果集

while(rs.next()){

String cname=rs.getString("cname");

}

rs.close;

st.close;

con.close;//关闭对数据库的连接

}