一、jsp和servlet的区别：  
Servlet是服务器端程序，而jsp是服务器页面程序

jsp本质上就是一个Servlet，在访问jsp时，在服务器端会先将jsp转化成Servlet，再将生产的servlet的结果响应给浏览器。

jsp是html页面中内嵌的Java代码，侧重于页面显示；Servlet中书写Java代码，侧重逻辑控制。

二、过滤器的部署：

1、新建一个class，实现接口Filter。

2、重写过滤器的doFilter(request，response，chain)方法。在doFilter方法中进行过滤操作。

3、在web.xml中配置过滤器。

作用与所有web资源：<url-pattern>/\*</url-pattern>。则客户端请求访问任意资源文件时都要经过过滤器的过滤，通过则可以访问，否则不能访问。

作用于某一文件夹下所有文件：<url-pattern>/dir/\*</url-pattern>

作用于某一种类型的文件：<url-pattern>\*.扩展名</url-pattern>。如：<url-pattern>\*.jsp</url-pattern>过滤所有对jsp文件的访问请求。

作用于某一文件夹下某一类型文件：<url-pattern>/dir/\*.扩展名</url-pattern>

如果一个过滤器需要过滤多种文件，则可以配置多个<filter-mapping>，一个mapping定义一个url-pattern来定义过滤规则。

三、Session什么时候创建？什么时候销毁？

什么时候创建：server 端程序(如 Servlet )调用HttpServletRequest.getSession(true)这样的语句时才会被创建。

什么时候销毁：

session的持有者(即客户端浏览器)在最大无活动等待时间(MaxInactiveInterval)内无任何响应或请求   
2.session被调用invalidate()方法强制销毁了   
3.jsp服务器重起或发生中断(此时应该是所有在服务器上的session全都OVER了)

四、为什么要用session：

session对象是一种会话对象，用来记录每个客户端的访问状态，其实,HTTP协议是一种无状态协议，也就是客户端向服务器发送一个请求request，然后服务器返回一个相应response，之后这个连接就会被关闭，两者也没有任何关系了，也就是服务器中不会存储此次请求的有关信息，再次请求时服务器就无法知道这次请求和上次请求是否是一个客户了。所以我们就需要采用会话session来记录这次连接的信息了。

一个客户端访问服务器时，可能会在这个服务器的多个页面之间不断刷新、反复连接同一个页面或者向一个页面提交信息，有了session的记录，服务器就可以知道这就是同一个客户端在完成动作罢了

五、转发和重定向的区别：

转发在服务器端完成的；重定向是在客户端完成的   
转发的速度快；重定向速度慢   
转发的是同一次请求；重定向是两次不同请求   
转发不会执行转发后的代码；重定向会执行重定向之后的代码   
转发地址栏没有变化；重定向地址栏有变化   
转发必须是在同一台服务器下完成；重定向可以在不同的服务器下完成

六、九大内置对象：

| 对象名 | 描述 | 作用域 |
| --- | --- | --- |
| request | 请求对象 | Request |
| response | 响应对象 | Page |
| pageContext | 页面上下文对象 | Page |
| session | 会话对象 | Session |
| application | 应用程序对象 | Application |
| out | 输出对象 | Page |
| config | 配置对象 | Page |
| page | 页面对象 | Page |
| exception | 例外对象 | Page |

七、jsp的4个域对象及作用范围  
Page：作用范围是整个jsp，生命周期是当jsp的请求时开始，响应结束时销毁  
Request：作用范围是整个请求链（有转发就一直存在），生命周期是在service方法调用前由服务器创建，传入service方法。整个请求结束，request生命结束。  
Session：作用范围是一次会话，生命周期是第一次调用getSession()方法时创建。当一段时间没有使用该session的时候销毁（30min）；调用invalidate()方法销毁；服务器强制关闭也会销毁。  
Application：作用范围是整个服务器，生命周期是服务器启动创建，直到关闭销毁

八、C3P0：  
它是扩展了传统的jdbc数据库连接池  
当用户需要访问数据库时，并非建立一个新的连接，而是从连接池中取出一个已经建立的空闲连接对象。使用完毕后，用户不关闭连接，而是将数据库连接对象放回连接池中。数据库连接池管理数据连接的建立、断开，同时监视数据库连接数量和使用情况。使用数据库连接池会显著提高整个应用程序的伸缩性（大大提高了连接数量）和健壮性（能够应对大量用户频繁连接数据库，减少系统资源的消耗），提高应用程序的性能指标。

九、Session和cookie的区别：

1、cookie数据存放在客户的浏览器上，session数据放在服务器上。

2、cookie不是很安全，别人可以分析存放在本地的cookie并进行cookie欺骗，考虑到安全应当使用session。

3、session会在一定时间内保存在服务器上。当访问增多，会比较占用你服务器的性能，考虑到减轻服务器性能方面，应当使用cookie。

4、单个cookie保存的数据不能超过4K，很多浏览器都限制一个站点最多保存20个cookie。

5、可以考虑将登陆信息等重要信息存放为session，其他信息如果需要保留，可以放在cookie中。

十、事务的特性：

原子性、一致性、隔离性、持久性：

十一、JSTL 是什么：

Java Server Pages 标准标签库 (JSTL) 包含的各种标签可用于 JSP 页面中。

十二、JQ选择器：

基本选择器、层次选择器、过滤选择器、表单选择器

十三、Get和post的区别：

get重点在从服务器上获取资源，post重点在向服务器发送数据；  
get传输数据是通过URL请求，以field（字段）= value的形式，置于URL后，并用"?"连接，多个请求数据间用"&"连接，如http://127.0.0.1/Test/login.action?name=admin&password=admin，这个过程用户是可见的；  
post传输数据通过Http的post机制，将字段与对应值封存在请求实体中发送给服务器，这个过程对用户是不可见的；  
Get传输的数据量小，因为受URL长度限制，但效率较高；  
Post可以传输大量数据，所以上传文件时只能用Post方式；  
get是不安全的，因为URL是可见的，可能会泄露私密信息，如密码等；  
post较get安全性较高；  
get方式只能支持ASCII字符，向服务器传的中文字符可能会乱码。  
post支持标准字符集，可以正确传递中文字符。