监听器：

**监听器是接口，一个类去实现了特定的监听接口，那么此类就具备了监听特定内容的功能！**

**当监听的内容，比如域对象创建与销毁、值状态改变，那么该类实现的接口中的方法就会自动执行！**

**监听域对象的创建与销毁 3个**

**ServletContextListener，监听ServletContext域对象的创建与销毁**

**HttpSessionListener，监听HttpSession域对象的创建与销毁**

**ServletRequestListener，监听ServletRequest请求对象的创建与销毁**

**监听域对象中值状态的改变 3个**

**ServletContextAttributeListener，监听ServletContext域对象中值状态改变**

**HttpSessionAttributeListener，监听HttpSession域对象中值状态的改变**

**ServletRequestAttributeListener，监听ServletRequest域对象中值状态的改变**

**监听对象与session的关系2个**

**HttpSessionBindingListener，监听对象是否在Session中存在**

**HttpSessionActivationListener，可以实现Session的活化与钝化**

**钝化：**

**当服务器正常关闭时,还存活着的session(在设置时间内没有销毁) 会随着服务器的关闭被以文件(“SESSIONS.ser”)的形式存储在tomcat 的work 目录下,这个过程叫做Session 的钝化。**

**活化：**

**当服务器再次正常开启时,服务器会找到之前的“SESSIONS.ser” 文件，从中恢复之前保存起来的Session 对象，这个过程叫做Session的活化。**

**实现序列化接口Serializable!**

**注册方式2：注解@WebListener**

过滤器：

**Filter接口：一个类实现了过滤器接口Filter，具备了过滤、筛选请求的功能！！！**

**过滤器的作用：**

**在请求到达servlet和jsp之前，针对请求路径做出判断处理，执行不同的操作，用来简化servlet或者jsp的代码！从一定角度的出发的话，可以将过滤器理解为一个工具类！**

**过滤器如何使用**

**创建一个类作为过滤器，并且让其实现Filter接口！**

**然后注册**

**@WebFilter urlPatterns ==> String[] dispatcherTypes ==> Dispatcher[]**

**过滤器的生命周期：**

**init() 当项目被tomcat加载启动服务器是，就会初始化过滤器**

**doFilter() 当请求路径满足了过滤器设定的过滤路径时，就会执行过滤的方法+放行**

**destroy() 服务器正常关闭时会执行销毁方法**