## Servlet生命周期

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。

**servlet何时被创建？**

1. 默认情况下，当web客户第一次请求访问某个servlet时，web容器将创建这个servlet实例
2. 当web.xml文件中如果<servlet>元素中指定了<load-on-startup>子元素时，servlet容器在启动web服务器时，将按照顺序创建并初始化servlet对象。

一旦servlet被创建，web服务器会自动调用init()方法来初始化该servlet

## JSP内置的对象及方法

request表示HttpServletRequest对象。它包含了有关浏览器请求的信息，并且提供了几个用于获取cookie,

　　header, 和session数据的有用的方法。

　　response表示HttpServletResponse对象，并提供了几个用于设置送回

　　浏览器的响应的方法(如cookies,头信息等)

　　out对象是javax.jsp.JspWriter的一个实例，并提供了几个方法使你能用于向浏览器回送输出结果。

　　pageContext表示一个javax.servlet.jsp.PageContext对象。它是用于方便存取各种范围的名字空间、servlet相关的对象的API，并且包装了通用的servlet相关功能的方法。

　　session表示一个请求的javax.servlet.http.HttpSession对象。Session可以存贮用户的状态信息

　　applicaton

　　表示一个javax.servle.ServletContext对象。这有助于查找有关servlet引擎和servlet环境的信息

　　config表示一个javax.servlet.ServletConfig对象。该对象用于存取servlet实例的初始化参数。

　　page表示从该页面产生的一个servlet实例

## 过滤器和拦截器的区别

Servlet中的Filter实现了javax.servlet.Filter接口，里面有3个主要方法，doFilter()、init()、destroy()。其中doFilter()是主要的处理逻辑。还要在web.xml中进行配置。主要用途是过滤字符编码、做一些业务逻辑判断等。其大致的执行过程就是在客户端输入一个URL，根据URL在web.xml中检查是否有对这一类的URL的过滤，如果有先走过滤逻辑，过滤走完之后如果没有问题再按照URL传入制定的Servlet或者JSP。



拦截器就是在某个类的方法前面做一些操作或者在之后做一些操作，原理就是动态代理。