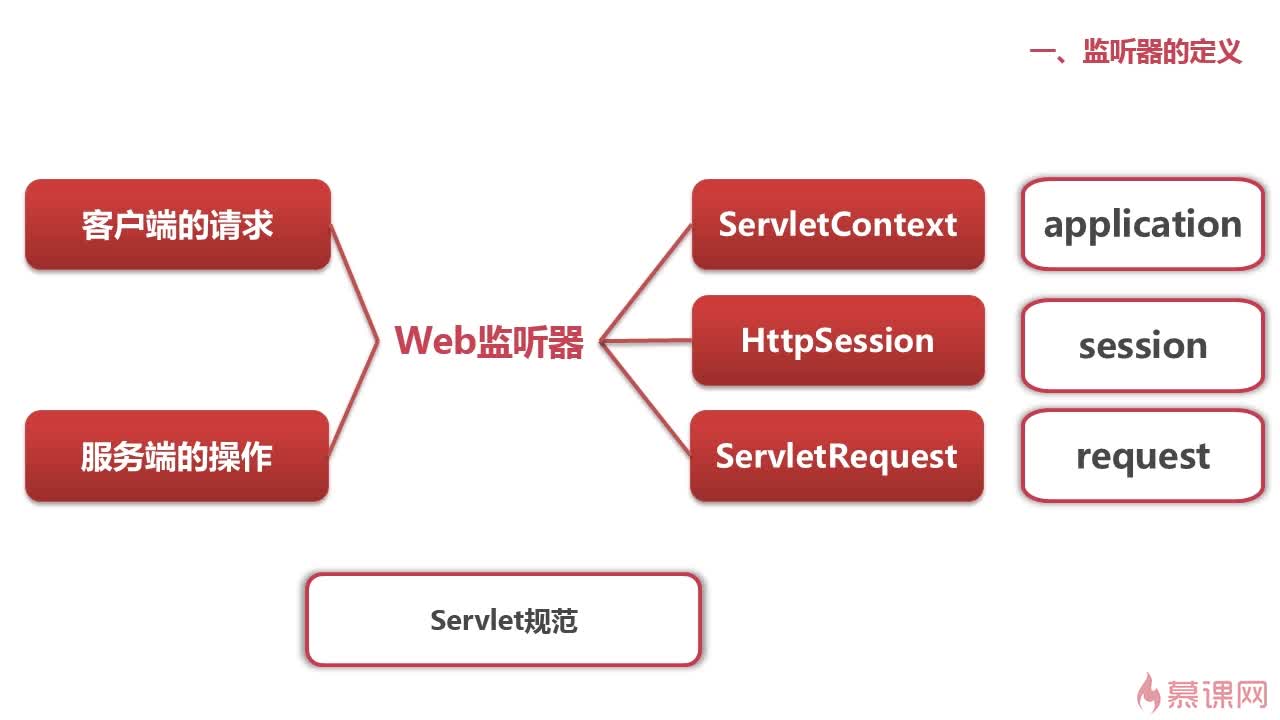
一、监听器的定义





二、监听器的用途

1、统计在线人数和在线用户。

2、系统启动时加载初始化信息。

3、统计网站的访问量。

4、跟Spring结合。

三、创建一个监听器

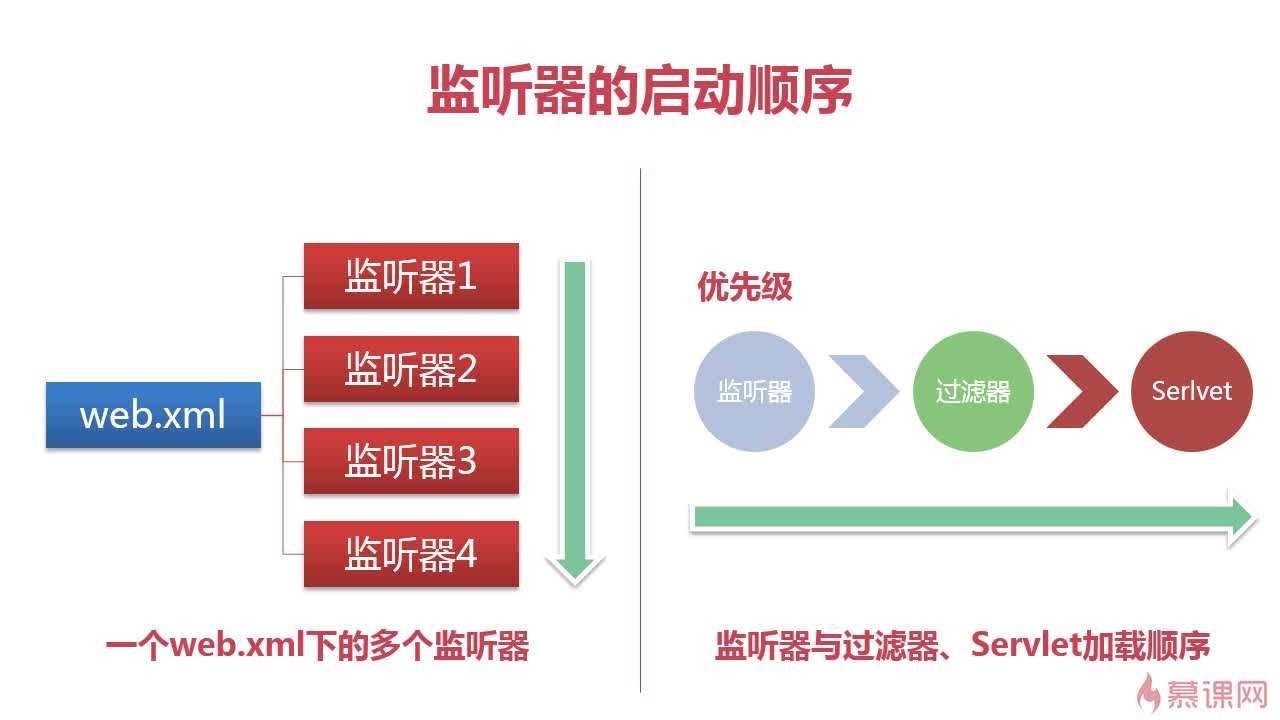
1、创建一个实现监听器接口的类。

实现Servlet中的接口

2、配置web.xml进行注册。

Add一个监听的类就行。

四、监听器的启动顺序



* 监听器的启动顺序：

当有多个监听器时，监听器的启动顺序按照在web.xml中的注册顺序启动

监听器、过滤器及Servlet的加载顺序：

优先级从大到小：监听器、过滤器、Servlet

五、监听器的分类

* 监听器按照监听的对象划分为：

1.用于监听应用程序环境对象（ServletContext）的事件监听器—ServletContextListener接口

2.用于监听用户会话对象（HttpSession）的事件监听器—HttpSessionListener接口

3.用于监听请求消息对象（ServletRequest）的事件监听器—ServletRequestListener接口

* 按监听的事件划分：

1.监听域对象自身的创建和销毁的事件监听器

2.监听域对象中的属性的增加和删除的事件监听器

3.监听绑定到HttpSession域中的某个对象的状态的事件监听器

* 自身的创建和销毁

1、ServletContext 对象的事件监听器---只有一个监听对象



* 获取初始化参数
* 在web.xml中配置

<context-param>

<param-name>InitParam</param-name>

<param-value>imooc</param-value>

</context-param>

* 在初始化方法中调用

String initParam = arg0.getServletContext().getInitParameter("InitParam");

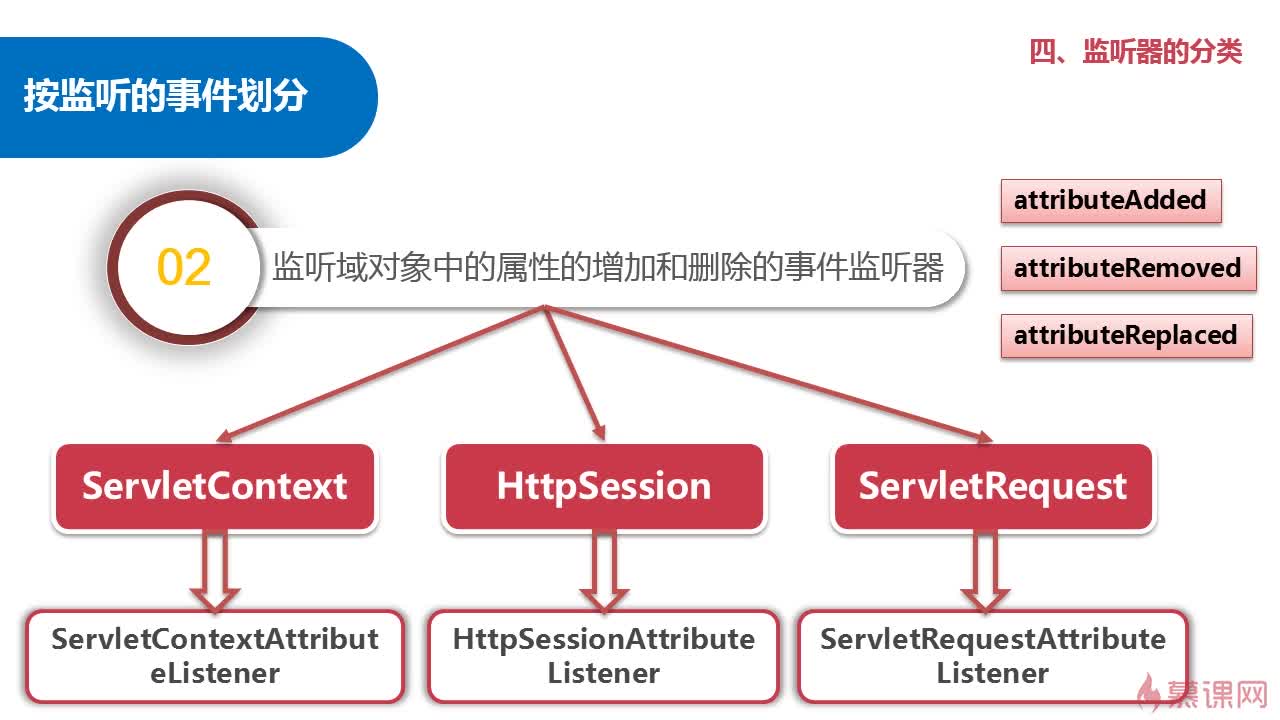
2、HttpSession对象的事件监听—可以有多个监听对象



3、ServletRequest 对象的事件监听—有多个监听对象



* 监听域对象中属性的增加和删除

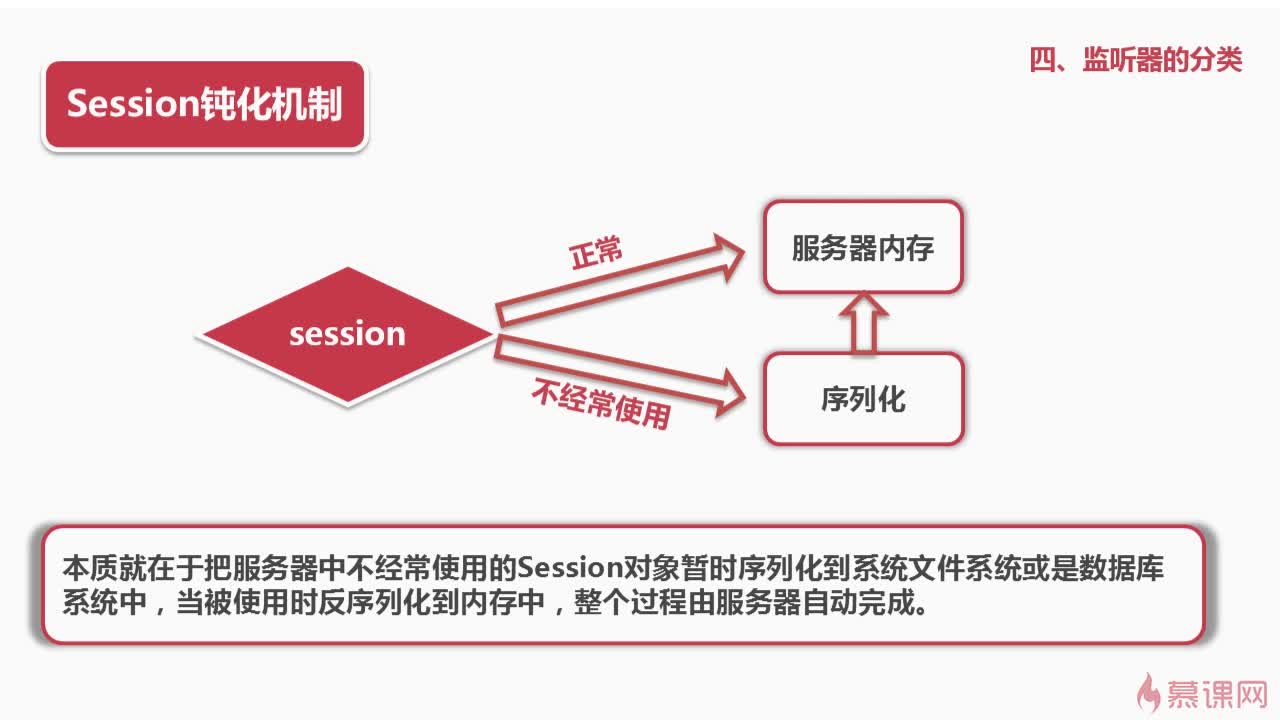


* 三个接口：ServletContextAttributeListener、HttpServletAtrributeListener、ServletRequestAttributeListener
* 三个方法：attributeAdded、attributeRemoved、attributeReplaced
* 处理步骤：在JSP页面中添加初始属性值以及删除的属性值

添加初始属性值方法—setAttribute

删除属性值方法—removeAttribute

* 监听绑定到HttpSession域中的某个对象的状态的事件监听器
* Session钝化机制



（1）当Tomcat服务器被关闭或者重启时，tomcat服务器会将当前服务器内存中的Session对象钝化到服务器文件系统中。

（2）Web应用程序被重新加载时，内存中的Session对象也会被钝化到服务器的文件系统中。

（3）可以配置主流内存中的Session对象数目，将不常使用的Session对象保存到文件系统或者数据库，当用时再重新加载。

（4）Tomcat服务器默认情况下提供两个钝化驱动类：FileStore和JDBCStore

* 绑定和钝化监听器

绑定就是将对象保存到Session中，解除绑定就是将对象从Session中删除。



（1）绑定和钝化监听器不用在web.xml中进行注册。

（2）绑定监听器只需要写一个javabean的类然后实现特定接口，然后当session绑定

了这个对象的时候就会调用这个javabean中实现接口的方法。

（3）钝化是当服务器关闭时服务器会将部分session序列化到本地，并且调用监听器的

监听钝化的方法。

反之当服务器再次运行时会调用监听session活化的方法并且将本地文件中的

session反序列化成session对象。

六、Servlet3.0下的监听器





* 代码实现：

@WebListener("This is My Servlet3.0 Listener.”)//省去了在web.xml中的注册的步骤

public class MyServletListener implement ServletContextListener{}

实验：监听器记录在线访问人数，通过session的监听器可以实现