**Init-param 和context-param区分**

init-param与context-param都是在web.xml里以键值对形式定义的变量，不同的是init-param是定义在servlet标签里，而context-param是定义在web-app标签里。示例如下：

定义init-param：

<servlet>

<servlet-name>ReadContext</servlet-name>

<servlet-class>file.ReadContext</servlet-class>

<init-param>

<param-name>user1</param-name>

<param-value>user1-ps</param-value>

</init-param>

</servlet>

定义param-name:

<context-param>

<param-name>user</param-name>

<param-value>user</param-value>

</context-param>

context-param使用方法：由于context-param是配置在web下面，属于上下文参数，在整个环境中都可使用，存放在getServletContext对像中，因此使用方法是：getServletContext().getInitParameter(“user”)，如：

public void service(HttpServletRequest request, HttpServletResponse response)throws ServletException,IOException{

String user=getServletContext().getInitParameter("user");

System.out.println(getServletContext().getInitParameter("user"));

System.out.println(user);

}

init-param使用方法：由于init-param是配置在servlet中，属于某一下servlet，存放在getServletConfig中，因此使用方法是：getServletConfig().getInitParameter(“user1”); 由于它属于当前的servlet类，所以用this替代getServletConfig(), 使用this.getInitParmeter(“user1”) , 如：

public void service(HttpServletRequest request, HttpServletResponse response)throws ServletException,IOException{

String user=getServletContext().getInitParameter("user1");

System.out.println(getServletConfig().getInitParameter("user1"));

//或this.getInitParmeter("user1");

System.out.println(user1);

}

总结：

context-param在所有的servlet中都能使用，init-param只能在当前的servlet中使用，如果不在当前的servlet中使用，取的值为null

---------------------