# Session/Cookie管理

**会话：**简单理解为用户打开一个浏览器访问某个网站，点击多个超链接，访问服务器多个web资源，然后关闭浏览器，整个过程称之为一个会话

**会话过程中需要解决的一些问题：**

每个用户在使用浏览器与服务器可进行会话的过程中，不可避免会产生一些数据，服务器要想办法为每个用户保存这些数据

**Cookie（客户端技术）**，服务器吧每个用户的数据（登录名，密码等）式写给用户浏览器，当用户浏览器再去访问服务器中的web资源时，就会带着各自数据去，这样web资源处理的就是用户各自的数据了

**Cookie作用：**保存用户登录信息，记录用户访问网站信息

**Cookie的基本用法：**

//创建cookie

Cookie cookie = new Cookie("name", value);

//设置cookie生命周期

cookie.setMaxAge(3600);

//把cookie信息回写给浏览器

response.addCookie(cookie);

//读取cookie

request.getCookies();

**案例：显示用户上次登录时间**



**Cookie小结：**

1. Cookie在服务端创建
2. Cookie保存在浏览器端
3. Cookie生命周期可以自定义设置，如果不设置的话，当浏览器关闭的话，cookie自动消亡
4. Cookie是以明文存放，安全性较低，可以对其进行加密

**Session（服务端技术）：**服务器在运行时，可以为每一个用户创建一个其独享的session对象，由于session为用户浏览器独享，所以用户在访问浏览器的web资源时，可以把各自的数据放在各自的session中，当用户再去访问服务器的其他资源时，其他web资源会再从用户的各自session中取处数据为用户服务。

**Session的机制：**用户打开浏览器，访问某个网站时，服务器就会在服务器的内存为该浏览器分配一个session对象，该session对象呗这个浏览器独占

**Session的作用：**保存登录用户的信息（购物车），将某些数据存放到session中，供同一用户的各个页面使用，防止非法用户登录到某个页面

**Session的基本使用：**

//访问session[当发现没有session的时候会自动创建session]

HttpSession session = request.getSession();

//给session添加属性

session.setAttribute("name", "孙常青");

session.setAttribute("age", "100");

//获得session

String name = (String) session.getAttribute("name");

//清除session

session.removeAttribute("name");

**案例：验证码处理**



注：项目见附件压缩文件包