

cookie

cookie和session一样，都是实现会话机制的一种技术，session是保存在服务端的，cookie是保存在客户端的。

通过sessionID来区分不同的会话，sessionID是由客户端产生的，保存在客户端的cookie，传到服务端之后，再将sessionID自动保存到session中，session和cookie中都保存有sessionID。

cookie是服务器在http响应中附带传给浏览器的一个文本文件，浏览器会保存cookie，在下一次访问服务器时再将cookie传给服务端，服务端再次做出响应时，又会将cookie带回给客户端，cookie就是这样在客户端和服务端之间来回传递。

Java中有一个Cookie类，该类专门用来描述cookie。

- 创建cookie

```
Cookie cookie = new Cookie("name", "lisi");
response.addCookie(cookie);
```

每一次客户端的请求都会在request对象中附带当前客户端中的所有cookie，服务器在做响应的时候，如果返回cookie对象，客户端接收到该响应之后，会更新当前的本地cookie：（如果响应的cookie的key不存在，则添加一个新的cookie，如果响应的cookie的key存在，则覆盖之前的cookie），在下一次请求的时候同样会将当前的cookie带入，服务器在做响应的时候重复之前的步骤。

读取cookie

```
Cookie[] cookies = request.getCookies();
```

void setMaxAge(int age) //设置cookie的有效时间，单位为秒

int getMaxAge() //获取cookie的有效时间

void setValue(String value) //修改cookie的值

String getName() //获取cookie的名称

String getValue() //获取cookie的值

session和cookie的区别

session：保存在服务端

保存的数据类型是Object

随着会话的结束而销毁

保存重要信息

cookie：保存在客户端

保存的数据类型是String

cookie可以长期的保存在客户端

保存不重要的信息

JSP内置对象作用域

4大作用域

page作用域：对应的内置对象是pageContext

request作用域：对应的内置对象是request

session作用域：对应的内置对象是session

application作用域：对应的内置对象是application

作用域的区别：

page作用域只在当前页面有效

request作用域对应一次请求，在同一次请求中有效

session作用域对应一次会话，在同一次会话中有效，关闭浏览器，重新打开就是一次新的会话

application作用域对应整个WEB应用程序，对应WEB服务，tomcat

作用域从小到大排序：page < request < session < application

服务端资源之间来回跳转的时候，一般的数据都保存在request中。

用户的登录信息，一般保存在session中。

所有用户共享的数据，网站访问量，文章的阅读量...，保存在application中。