1.利用 cookie 对象

Cookie 是服务器保存在客户端中的一小段数据信息。使用 Cookie 有一个前提，就是客户端浏览器允许使用 Cookie 并对此做出相应的设置。一般不赞成使用 Cookie。

（1）后台代码

```
1  Cookie cookie=new  Cookie("name",  "hello");
2  response.addCookie(cookie);
```

（2）前台代码

```
1  Cookie[] cookies=request.getCookies();
2  for(int  i=0;i<cookies.length;i++){
3                  if(cookies[i].getName().toString().equals("name")){
4                          out.print(cookies[i].getValue());
5                  }
6  }
```

2.利用 session 对象

session 对象表示特定会话 session 的用户数据。客户第一次访问支持 session 的 JSP 网页，服务器会创建一个 session 对象记录客户的信息。当客户访问同一网站的不同网页时，仍处于同一个 session 中。

（1）后台代码

```
1  request.getSession().setAttribute("name", name);
2  request.getSession().setMaxInactiveInterval(2);
3  response.sendRedirect("welcome.jsp");
```

（2）前台代码（jsp 页面）

```
1  Object user=request.getSession().getAttribute("name");
```

3.利用 request 重定向，设置 setAttribute

（1）后台代码

```
request.setAttribute("name",  "cute");
request.getRequestDispatcher("welcome.jsp").forward(request,
response);     //网址不会改变
```

PS:如果后台使用的转发代码为 response.sendRedirect("welcome.jsp"); //网址变为 welcome.jsp

则 request 设置的参数无效，因为已经切换到另一个请求了，request 参数的有效期为本次请求。

（2）前台代码

```
String name=request.getAttribute("name").toString();
```

4.利用 Ajax 进行异步数据请求（得到的数据可以以 json 或 xml 格式返回，便于处理）

(1)后台代码案例（运用 servlet 传输数据）

```
public  class  TestServlet  extends  HttpServlet {

        /**
          * Constructor of the object.
          */
        public  TestServlet() {
                super();
        }

        public  void  doGet(HttpServletRequest request, HttpServletResponse r
                        throws  ServletException, IOException {
                doPost(request, response);
        }

        public  void  doPost(HttpServletRequest request, HttpServletResponse
                        throws  ServletException, IOException {

                response.setContentType("text/html");
                PrintWriter out = response.getWriter();
                String
data="[{\"name\":\"apple\",\"price\":23},{\"name\":\"banana\",\"price\":12},{
                out.write(data);
```

```
                out.flush();
                out.close();
        }

        /**
          * Initialization of the servlet. <br>
          *
          * @throws ServletException if an error occurs
          */
        public  void  init()  throws  ServletException {
                // Put your code here
        }

}
```

2.前台 js 请求处理数据代码

```
function createXMLHttpRequest(){
        var xmlrequest;
        if(window.XMLHttpRequest){
                xmlrequest=new  XMLHttpRequest();
        }else  if(window.ActiveXObject){
                try{
                        xmlrequest=new  ActiveXObject("Msxml2.XMLHTTP");
                }catch(e){
                        try{
                                xmlrequest=new  ActiveXObject("Microsoft.XMLH
                        }catch(e){
                                xmlrequest="";
                        }
                }
        }
        return  xmlrequest;
}
//获取数据的函数
function change(){
        var xmlrequest=createXMLHttpRequest();
        xmlrequest.open("POST","TestServlet",true);
        xmlrequest.onreadystatechange=function(){
                if(xmlrequest.readyState==4&&xmlrequest.status==200){
                        var data=JSON.parse(xmlrequest.responseText);
                        var content="<table border=1>";
```

```
                        for(var i=0;i<data.length;i++){
                                content+="<tr>";
                                for(o in data[i]){
                                        content+="<td>"+data[i][o]+"</td>";
                                }
                                content+="</tr>";
                        }
                        content+="</table>";
                        document.getElementById("test").innerHTML=content;
                }
        };
        xmlrequest.send();
}
```

总结：在用户访问网站整个生命周期中都会用到的数据用 session 来存储，例如用户名，登录状态，购物车信息

显示在网页上的信息数据大多通过 request 或 Ajax 方式获取