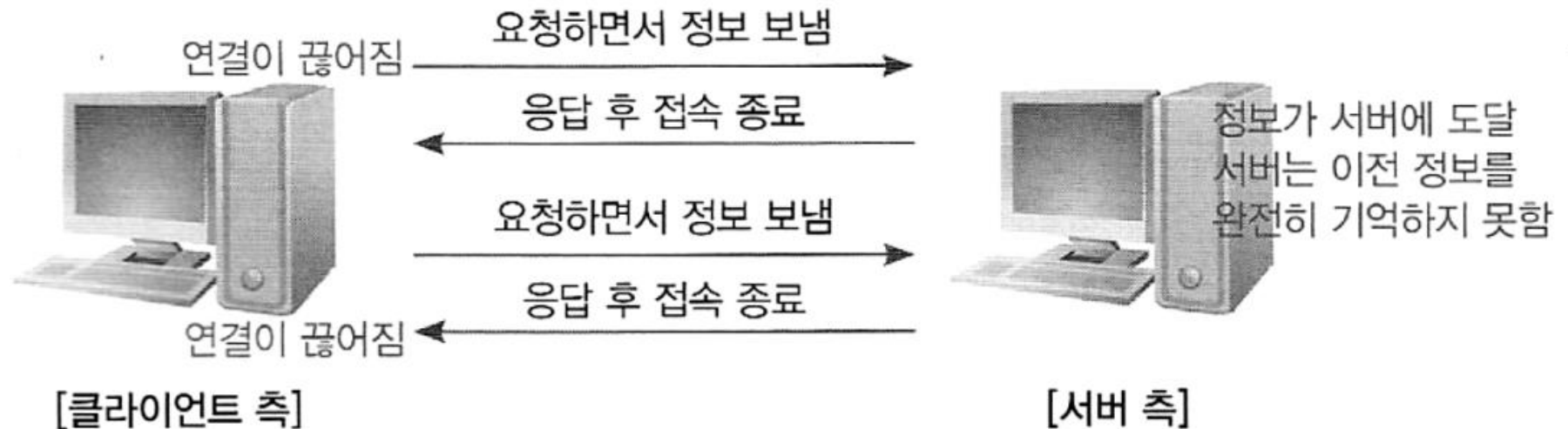


쿠키와 세션

쿠키

❖ HTTP 프로토콜의 특징

- 비연결 지향형 통신 프로토콜
 - 한 번 접속하여 요청을 보내고 처리한 후 접속을 끊음
 - 연결과 연결사이에 정보 유지 안됨
- Stateless 상태



쿠키

❖ HTTP 프로토콜의 특징

- 웹 애플리케이션에서는 연결과 연결 사이의 정보 유지 필요
 - 사용자 로그인 유지
 - 쇼핑 카트의 유지 등
- 이를 위해 사용되는 기법
 - 쿠키와 세션
 - 쿠키 : 브라우저에서 정보 관리
 - 세션 : 서버에서 정보 관리

쿠키

❖ 쿠키

- 정보를 담고 있는 속성(쿠키)을 요청을 서버에 보낼때 헤더에 항상 같이 보냄
- 쿠키 설정 절차
 - 쿠키 객체를 생성
 - 쿠키에 속성 값을 설정
 - `respons`객체의 `addCookie()` 메서드를 호출하여 쿠키를 추가
 - 다음 요청시부터는 항상 쿠키가 요청에 포함되어 전달

쿠키

❖ 쿠키 객체 생성

```
new Cookie(쿠키명, 쿠키값)
```

- 쿠키명과 쿠키값은 String 타입

```
Cookie info = new Cookie("testCookie", "I am First Cookie!");
```

쿠키

❖ 쿠키 메서드

메소드	설명
<code>void setComment(String)</code>	쿠키에 대한 설명을 설정
<code>void setDomain(String)</code>	쿠키의 유효한 도메인을 설정
<code>void setMaxAge(int)</code>	쿠키의 유효한 기간을 설정
<code>void setPath(String)</code>	쿠키의 유효한 디렉토리를 설정
<code>void setSecure(boolean)</code>	쿠키의 보안을 설정
<code>void setValue(String)</code>	쿠키의 값을 설정
<code>void setVersion(int)</code>	쿠키의 버전을 설정
<code>String getComment()</code>	쿠키에 대한 설명을 알려줌
<code>String getDomain()</code>	쿠키의 유효한 도메인 정보를 알려줌

쿠키

❖ 쿠키 메서드

메소드	설명
<code>int getMaxAge()</code>	쿠키의 사용할 수 있는 기간에 대한 정보를 알려줌
<code>String getName()</code>	쿠키의 이름을 알려줌
<code>String getPath()</code>	쿠키의 유효한 디렉토리 정보를 알려줌
<code>boolean getSecure()</code>	쿠키의 보안이 어떻게 설정되어 있는지를 알려줌
<code>String getValue()</code>	쿠키에 설정된 값을 알려줌
<code>int getVersion()</code>	쿠키의 버전을 알려줌

쿠키

❖ 유효기간 설정

- `setMaxAge()` 메서드로 초단위 설정

```
info.setMaxAge(365*24*60*60); // 1년 유효기간 설정
```

- 유효 기간을 설정하지 않은 경우
 - 브라우저가 닫히면 사라짐

쿠키

❖ 쿠키를 브라우저로 전송하기

```
response.addCookie(info);
```

쿠키

❖ 01_setCookies.jsp

```
<body>
  <%
    // 1. Cookie 객체 생성
    Cookie c = new Cookie("id", "pinksung");
    // 2. 유효기간 설정
    c.setMaxAge(365 * 24 * 60 * 60);
    // 3. 클라이언트에 쿠키 전송
    response.addCookie(c);
    //4. 쿠키를 생성하여 클라이언트에 전송
    response.addCookie(new Cookie("pwd", "test1234"));
    response.addCookie(new Cookie("age", "20"));
  %>
  <h3>쿠키 설정</h3>
</body>
</html>
```

쿠키

❖ 쿠키가 저장되는 곳(클라이언트)

C:\Users\컴퓨터이름\AppData\Roaming\Microsoft\Windows\Cookies

쿠키

❖ 서버에서 쿠키 정보 읽기

- response 객체에서 쿠키 객체 추출
- 쿠키 객체로 설정 값 추출

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

for(Cookie c : cookies) {
    out.println(c.getName() + " : " + c.getValue() + "<br>");
}
```

쿠키

❖ 02_getCookies.jsp - 설정된 모든 쿠키 얻고 출력하기

```
<body>
  <h3> 클라이언트로부터 얻어온 Cookie </h3>
  <%
    Cookie[] cookies = request.getCookies();
    for (Cookie c : cookies) {
      out.println(c.getName()+" : "+c.getValue()+"<br>");
    }
  %>
</body>
</html>
```

쿠키

❖ 쿠키 삭제

- 쿠키의 유효 기간을 0으로 설정

```
Cookie cookie = new Cookie("id", "");  
cookie.setMaxAge(0); // 쿠키의 유효기간을 만료시킴  
response.addCookie(cookie);
```

❖ 02_removeCookies.jsp - id 쿠키 삭제하기

```
<%  
    Cookie cookie = new Cookie("id", "");  
    cookie.setMaxAge(0); // 쿠키의 유효기간을 만료시킴  
    response.addCookie(cookie);  
%>  
<body>  
    <h3> id 쿠키가 삭제되었습니다. </h3>  
    <a href="02_getCookies.jsp"> 쿠키 확인</a>  
</body>  
</html>
```

세션(session)

❖ 쿠키의 특징

- 클라이언트의 파일시스템에 저장
 - 보안에 취약
 - 크기 제한(1.2MB)

❖ 세션

- 연결 정보를 서버에 저장
 - JSP/Servlet(서버)에서만 접근 가능 - 보안 강화
 - 저장 데이터의 크기 한계 없음
- 브라우저 단위당 한 개씩 존재
- 활용
 - 로그인 처리

세션(session)

❖ 세션 사용하기

메소드	설명
<code>Object getAttribute(String name)</code>	이름에 해당되는 객체 값을 가져온다. 없을 경우에는 null을 반환한다. 반환값은 Object 형이므로 반드시 형 변환을 하여 사용해야 한다.
<code>Enumeration getAttributeNames()</code>	세션에서 모든 객체의 이름을 Enumeration 형으로 얻어 준다.
<code>long getCreationTime()</code>	세션이 만들어진 시간을 반환한다.
<code>String getId()</code>	해당 세션을 가리키는 고유 id 값을 String 형으로 변환한다.
<code>long getLastAccessedTime()</code>	해당 세션이 클라이언트가 마지막으로 request를 보낸 시간을 long 형으로 반환한다.
<code>int getMaxInactiveInterval()</code>	사용자가 다음 요청을 보낼 때까지 세션을 유지하는 최대 시간(초 단위)을 반환한다.

세션(session)

❖ 세션 사용하기

메소드	설명
<code>boolean isNew()</code>	해당 세션이 처음 생성되었으면 true 값을 반환하고 이전에 생성된 세션이라면 false를 반환한다.
<code>void removeAttribute(String name)</code>	지정된 이름에 해당하는 객체를 세션에서 제거한다.
<code>void setAttribute(String name, Object value)</code>	세션에 지정된 이름에 객체를 추가한다.
<code>void setMaxInactiveInterval(int interval)</code>	사용자가 다음 요청을 보낼 때까지 세션을 유지하는 최대 시간(초 단위)을 설정한다. 이 시간을 넘기면 서블릿 컨테이너는 세션을 종료한다.
<code>void invalidate()</code>	해당 세션을 없애고 세션에 속해있는 값들을 없앤다.

세션(session)

❖ 세션

- session 내장 객체에서 관리
 - `getAttribute()`, `setAttribute()` 메서드로 정보 관리

세션(session)

❖ 04_setSession.jsp

```
<body>
  <%
    session.setAttribute("id", "pinksung");
    session.setAttribute("pwd", "test1234");
    session.setAttribute("age", 20);
  %>
  <h3> 세션 설정</h3>
</body>
</html>
```

세션(session)

❖ 05_getSession.jsp

```
<body>
  <h3> 세션 값 얻어오기 </h3>
  <%
    String id = (String) session.getAttribute("id");
    String pwd = (String) session.getAttribute("pwd");
    Integer age = (Integer) session.getAttribute("age");
  %>
  id : <%= id %><br>
  pwd : <%= pwd %><br>
  age : <%= age %><br>
</body>
</html>
```

세션(session)

❖ 서블릿에서 세션 사용하기

```
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

    HttpSession session = request.getSession();

}

```

② 변수에 저장함 ① 세션 객체를 얻음

세션(session)

❖ 세션에 있는 모든 정보 읽기

메소드	설명
boolean hasMoreElements()	Enumeration 객체에 뽑아올 데이터가 있는지를 검사하여 있다면 true를, 더 이상 데이터가 존재하지 않는다면 false를 리턴합니다.
E nextElement()	다음 아이템이 존재한다는 전제 하에서 아이템을 하나씩 뽑아오는 역할을 합니다.

```
Enumeration names = session.getAttributeNames();           — ①
while(names.hasMoreElements()){                             — ②
    String name=names.nextElement().toString() ;           — ③
    String value=session.getAttribute(name).toString();    — ④
    out.println(name + " : " + value + "<br>");
}
```

세션(session)

❖ 06_getSessions.jsp

```
<%@page import="java.util.Enumeration"%>
:
<body>
    <h3>세션에 설정된 모든 값 얻어오기</h3>
    <%
        Enumeration<String> names = session.getAttributeNames();
        while (names.hasMoreElements()) {
            String name = names.nextElement().toString();
            String value = session.getAttribute(name).toString();
            out.println(name + " : " + value + "<br>");
        }
    %>
</body>
</html>
```


세션(session)

❖ 세션의 유효기간 설정

- 기본값은 30분
- `setMaxInactiveInterval()`로 설정가능
- 디폴트 설정 값의 변경
 - 톰캣의 `conf/web.xml`에서 설정

```
<session-config>  
    <session-timeout>30</session-timeout>  
</session-config>
```

세션(session)

❖ 07_infoSession.jsp

```
<body>
  <h3>세션 정보를 얻어오는 메소드를 사용하기</h3>
  <%
    String id_str = session.getId();
    long lasttime = session.getLastAccessedTime();
    long createdtime = session.getCreationTime();
    long time_used = (lasttime - createdtime) / 60000;
    int inactive = session.getMaxInactiveInterval() / 60;
    boolean b_new = session.isNew();
  %>
```

세션(session)

❖ 07_infoSession.jsp

```
[1] 세션 ID는 [  
<%=session.getId()%>] 입니다.  
<br><hr>  
[2] 당신이 웹 사이트에 머문 시간은  
<%=time_used%>  
분입니다.  
<br><hr>  
[3] 세션의 유효 시간은  
<%=inactive%>  
분입니다.  
<br><hr>  
[4] 세션이 새로 만들어졌나요?  
<br>  
<%  
    if (b_new)  
        out.print(" 예! 새로운 세션을 만들었습니다.");  
    else  
        out.print(" 아니오! 새로운 세션을 만들지 않았습니다.");  
%>  
</body>  
</html>
```

세션(session)

❖ 세션 제거하기

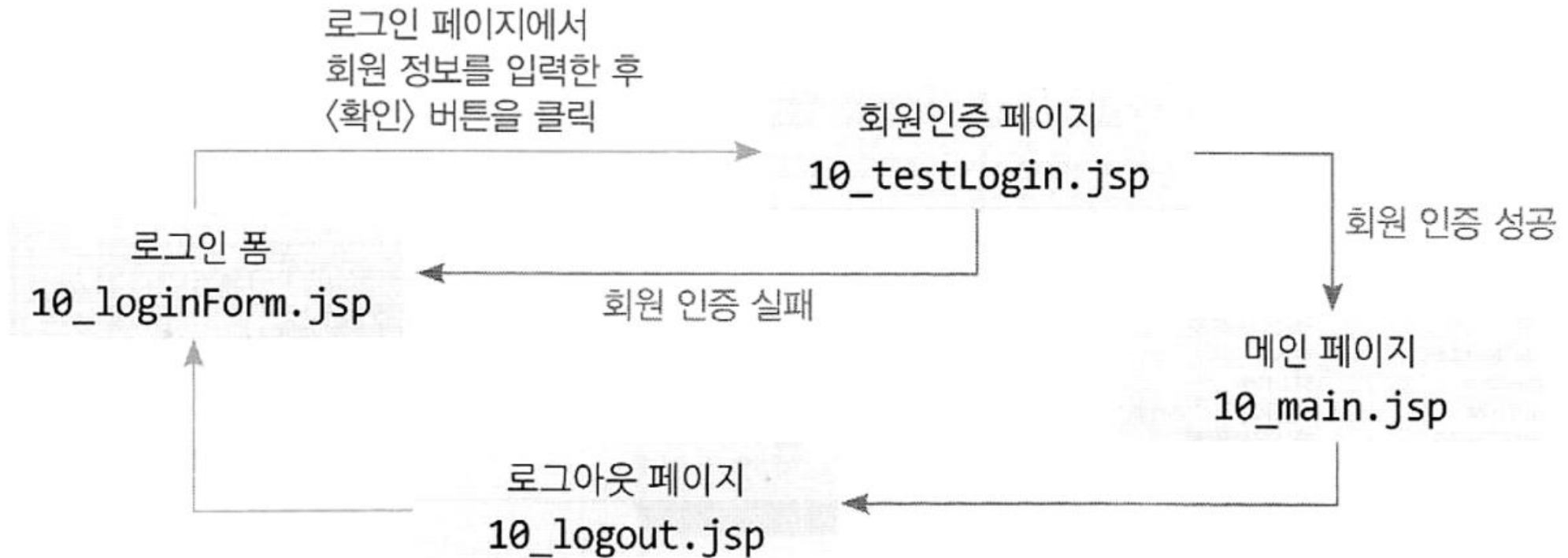
- 세션에서 속성 제거
 - `session.removeAttribute(속성명)`
- 세션의 모든 속성 제거
 - `session.invalidate()`

❖ 예제 : 세션을 이용한 로그인 처리

파일	설명
10_loginForm.jsp	클라이언트가 ID와 암호를 입력할 수 있는 화면을 제공하기 위한 페이지
10_testLogin.jsp	로그인 화면에서 제공한 정보를 가져와 회원 인증 절차를 거치는 페이지
10_main.jsp	로그인 인증 받은 회원이 다양한 정보를 제공받을 수 있는 페이지
10_logout.jsp	인증된 사용자의 인증을 무효화하는 페이지

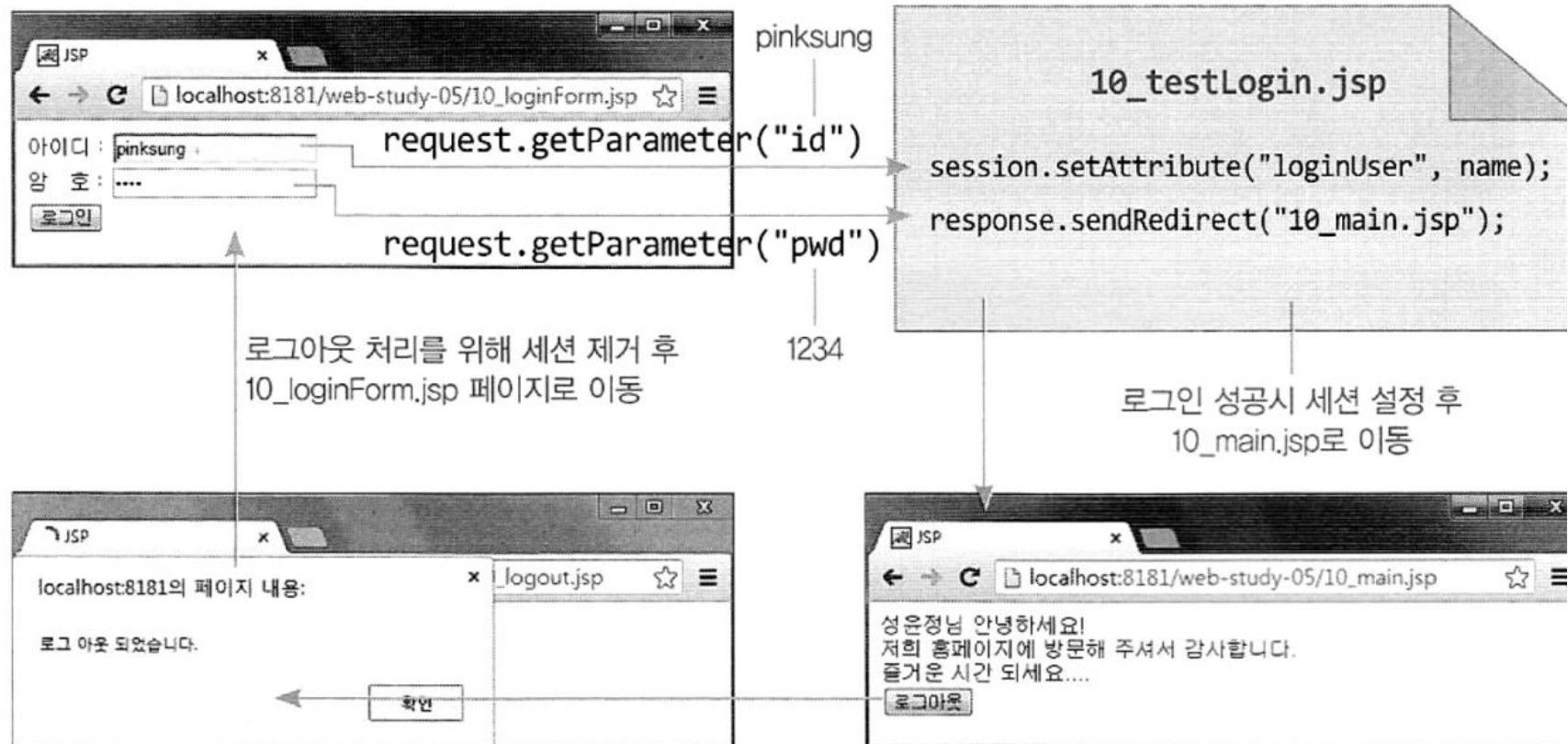
세션(session)

❖ 예제 : 세션을 이용한 로그인 처리



세션(session)

❖ 예제 : 세션을 이용한 로그인 처리



❖ 10_loginForm.jsp

```
<body>
  <form method="post" action="10_testLogin.jsp">
    <label for="userid"> 아이디 : </label>
    <input type="text" name="id" id="userid"><br>
    <label for="userpwd"> 암호 : </label>
    <input type="password" name="pwd" id="userpwd" ><br>
    <input type="submit" value="로그인">
  </form>
</body>
</html>
```


세션(session)

❖ 10_testLogin.jsp

```
<%
    String id="pinksung";
    String pwd="1234";
    String name="성윤정";

    if(id.equals(request.getParameter("id")) &&
        pwd.equals(request.getParameter("pwd"))) {
        session.setAttribute("loginUser", name);
        response.sendRedirect("10_main.jsp");
    }
    else{
        response.sendRedirect("10_loginForm.jsp");
    }
%>
```

세션(session)

❖ 10_main.jsp

```
<body>
<%
    if(session.getAttribute("loginUser")==null){
        response.sendRedirect("10_loginForm.jsp");
    }else{
%>
        <%=session.getAttribute("loginUser")%>님 안녕하세요!<br>
        저희 홈페이지에 방문해 주셔서 감사합니다.<br>
        즐거운 시간 되세요....<br>
        <form method="post" action="10_logout.jsp">
            <input type="submit" value="로그아웃">
        </form>

<%
    }
%>
</body>
</html>
```

세션(session)

❖ 10_logout.jsp

```
<body>
<%
    session.invalidate();
%>
<script>
    alert("로그 아웃 되었습니다.");
    location.href="10_loginForm.jsp";
</script>
</body>
</html>
```