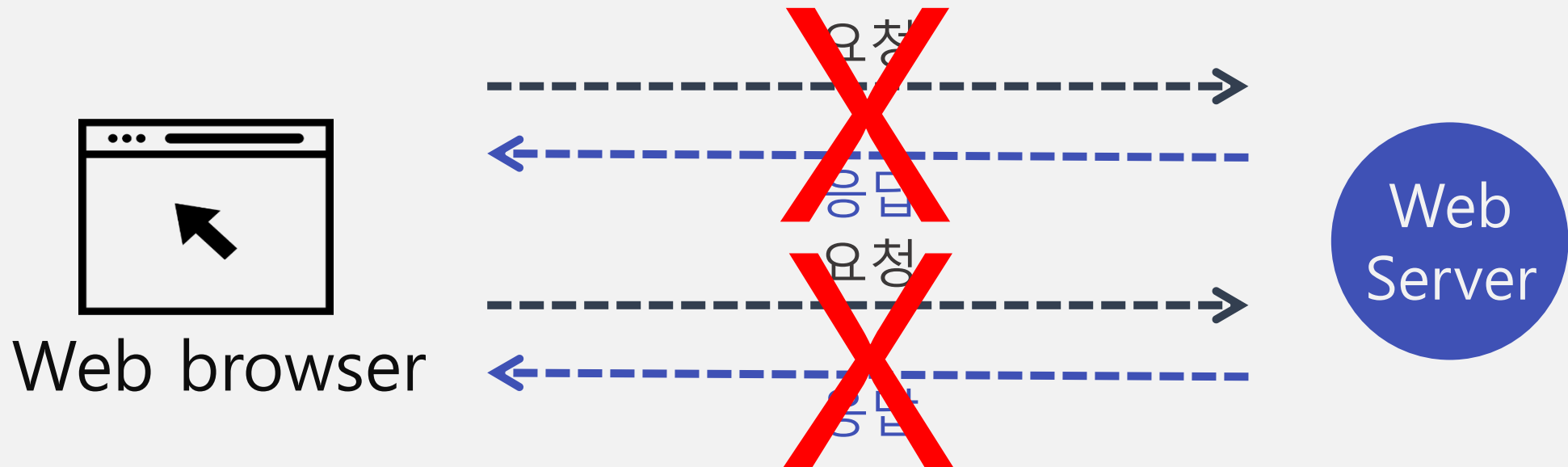


Cookie & Session

(쿠키와 세션)

~~HTTP~~
Hyper Text Transfer Protocol
~~비연결형 프로토콜~~

HTTP 한계

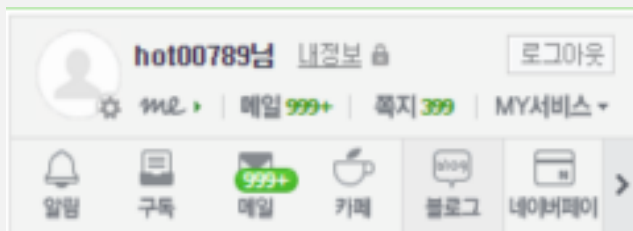


Client가 Server로 요청하면 해당 Page를
Client에게 전송(응답)한 후 연결종료

HTTP 한계



<네이버 메일>



<네이버 블로그>

Cookie & Session

Client의 정보를 지속적으로 유지하기 위한 방법

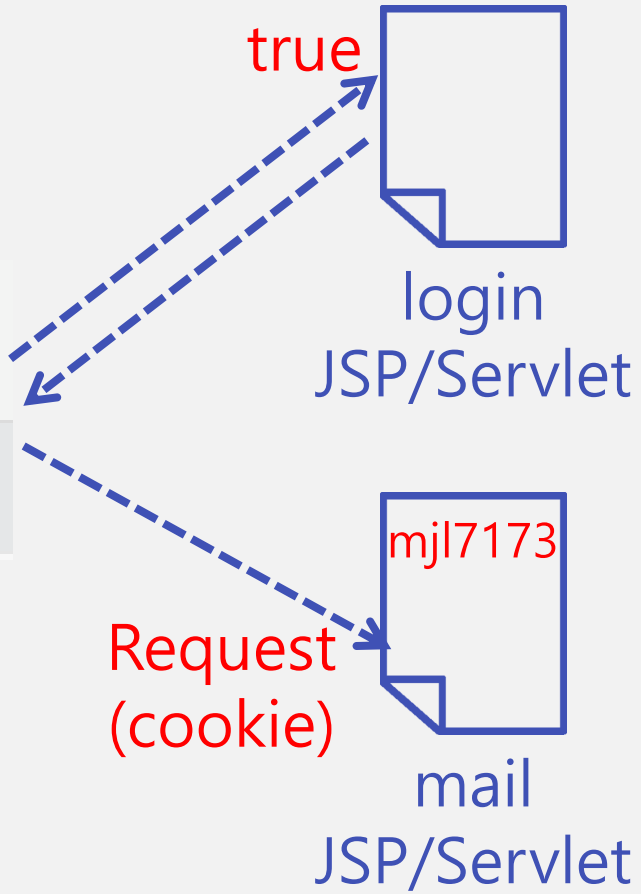
Cookie

전달할 데이터를 Web Browser(Client)로 보냈
다가 Web Server로 되돌려 받는 방법

Cookie

Client

ID : mjl7173



Cookie 저장 방법

cookie 이름

cookie 값

```
<% Cookie cookie = new Cookie("Id", "smhrd"); %>
```

1. Cookie 클래스 객체 생성
2. addCookie() 메소드 호출

Cookie 저장 방법

cookie 객체

```
<% response.addCookie(cookie); %>
```

- 1. Response 내장 객체에 addCookie() 호출
- 2. Cookie 객체를 Parameter로 전송

Cookie 저장 방법

▼ General

Request URL: http://localhost:8083/cookieSession/cookie.jsp

Request Method: GET

Status Code: 🟢 200 OK

Remote Address: [::1]:8083

▼ Response Headers

[view source](#)

Content-Length: 322

Content-Type: text/html; charset=EUC-KR

Date: Sun, 12 Mar 2017 15:28:42 GMT

Server: Apache/2.4.18 (Ubuntu)

Set-Cookie: Id=smhrd

▼ Request Headers

[view source](#)

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8

Accept-Encoding: gzip, deflate, sdch, br

Accept-Language: ko-KR,ko;q=0.8,en-US;q=0.6,en;q=0.4

Cache-Control: no-cache

Connection: keep-alive

Cookie: JSESSIONID=D478514EBD330ABD87FD046CBA8B3771; Id=smhrd

Host: localhost:8083

Pragma: no-cache

Upgrade-Insecure-Requests: 1

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/56.0.2924.87 Safari/537.36

Cookie 저장 방법

```
<!-- cookie 저장 -->  
<% Cookie cookie = new Cookie("Id", "smhrd"); %>  
  
<% response.addCookie(cookie); %>
```

쿠키를 저장하였습니다.

← → ↻ ⓘ localhost:8083/cookieSession/cookie.jsp

쿠키를 저장하였습니다.

Cookie 저장 조 회

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

Web Browser는 Web Server가 아무런 요청을 하지 않아도 URL을 보낼 때마다 그 URL에 포함된 Web Server에 해당하는 Cookie를 보냄

Cookie 저장 조회

```
<!-- 쿠키 조회 -->
```

```
<%
```

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

```
for(int i=0; i<cookies.length; i++){  
    out.println("쿠키 이름: "+cookies[i].getName());  
    out.println("쿠키 값 : "+cookies[i].getValue()+"<br>");  
}
```

```
%>
```

Insert title here



localhost:8083/cookieSession/cookieSelect.jsp

쿠키 이름: JSESSIONID 쿠키 값 : D478514EBD330ABD87FD046CBA8B3771
쿠키 이름: Id 쿠키 값 : smhrd

Cookie 저장 삭제

```
<% cookie.setMaxAge(0); %>
```

1. 쿠키의 유효 기간을 결정하는 메소드
2. addCookie()메소드 호출 전 설정
3. 쿠키를 삭제하기 위해서는 0 값을 입력

Cookie 실습



이벤트 뷰 페이지



Intro.jsp에서
체크박스를
선택했는지 검사

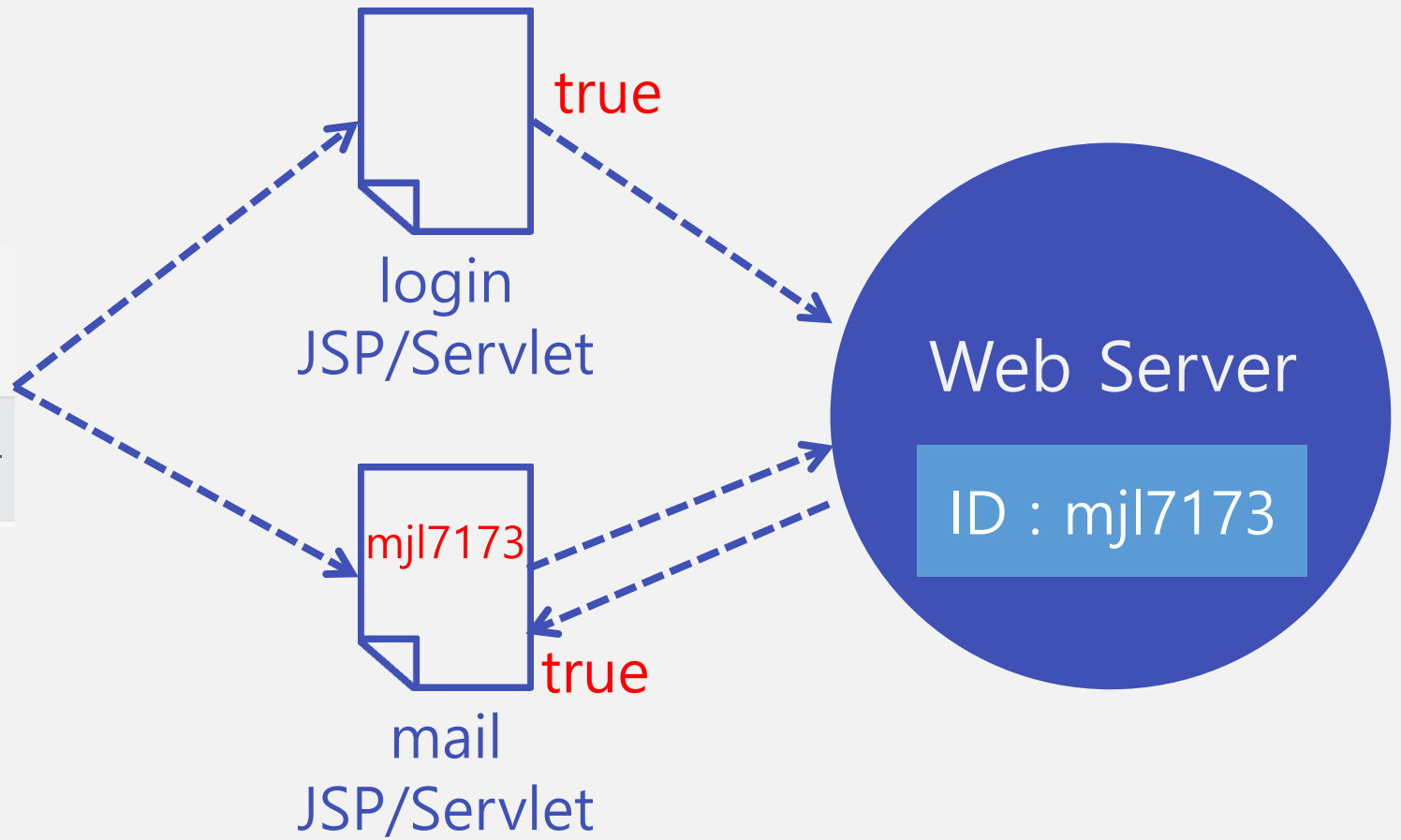
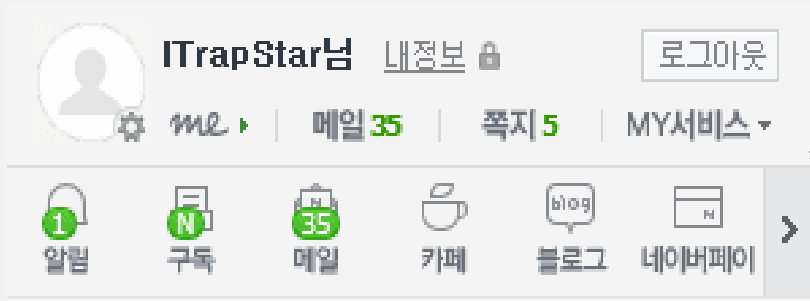


쿠키에 체크한
값이 있는지
없는지 체크

Session

Web Browser를 거치지 않고 Web Server에
있는
데이터 영역을 통해 데이터를 전달하는 방법

Session



Session 저장 방법

```
public class sessionServlet extends HttpServlet {  
    private static final long serialVersionUID = 1L;  
  
    protected void service(HttpServletRequest request, HttpServletResponse response)  
        throws ServletException, IOException {  
  
        HttpSession session = request.getSession();  
        session.setAttribute("age", "53");  
    }  
}
```

session의
데이터 값

session 데이터 영
역에 값 저장

Session 저장 방법

session 이름 session 값

The diagram shows a JSP code snippet: `<% session.setAttribute("age", "53"); %>`. The string `"age"` is enclosed in a blue box, and a dashed arrow points from the label 'session 이름' (session name) above to this box. The string `"53"` is also enclosed in a blue box, and a dashed arrow points from the label 'session 값' (session value) above to this box.

```
<% session.setAttribute("age", "53"); %>
```

setAttribute() 메소드 호출

setAttribute() 메소드는 같은 이름의 데이터가
이미 있으면 기존 값을 지우고 새로운 값을 저장하므로 데이터
수정 기능 구현

Session 조회 방법

```
<% String age = (String)session.getAttribute("age"); %>
```

getAttribute() 메소드를 호출하면 Session 영역의 데이터를 가져온다

메소드의 리턴 타입은 java.lang.Object 이므로 데이터 값을 본래의 타입

으로 사용하려면 타입 캐스팅을 해줘야 한다

Session 삭제 방법

```
<%= session.removeAttribute("age"); %>
```

removeAttribute() 메소드를 호출해
session 영역에 특정 데이터를 삭제할 수 있다

Session 유효 기간

```
<%= session.setMaxInactiveInterval(1800); %>  
<%= session.setMaxInactiveInterval(30 * 60); %>
```

일정 시간 동안 요청이 없으면 session 제거
(단위: 초)

Session 종료 방법

```
<% session.invalidate(); %>
```

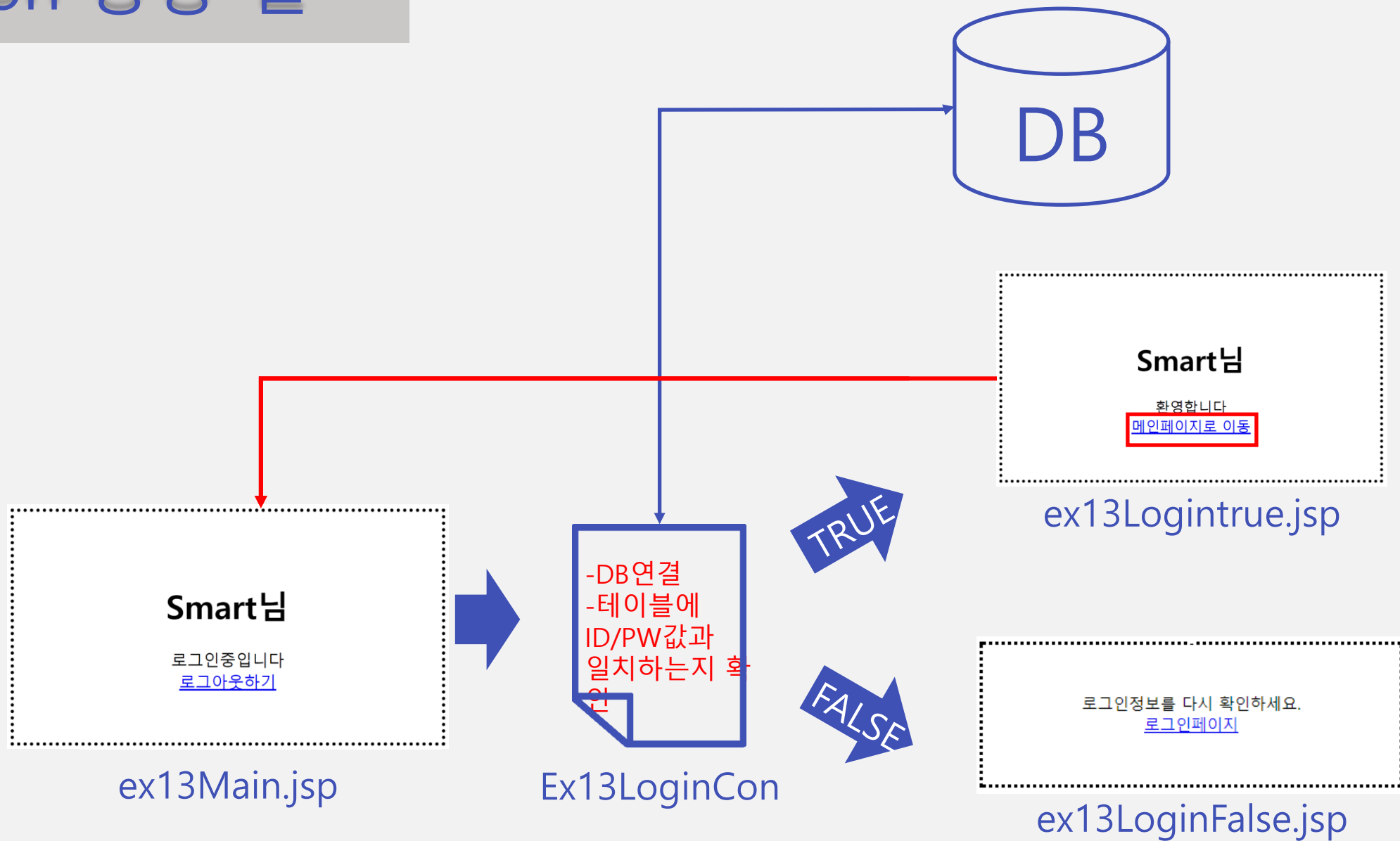
session을 이용해서 작업을 다 마친 후, 모든 값을 사라
지도록
하려면 invalidate() 메소드를 호출해 session을 종료

Cookie vs Session

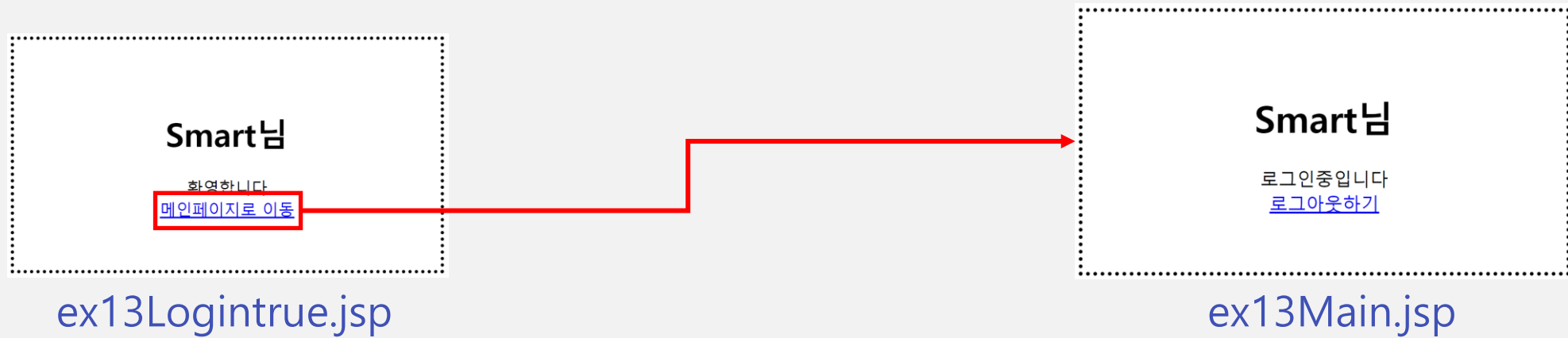
	Cookie	Session
저장위치	Client PC	Server PC
저장형식	Text	Object
보안	어려움	강력함
자원	Client 자원 사용	Server 자원 사용
용량제한	한 도메인 당 20개, 쿠키 하나 당 4KB, 총 300개 => 1.2MB	서버가 허용하는 한 용량에 제한이 없음

session 응용 실

습(3)



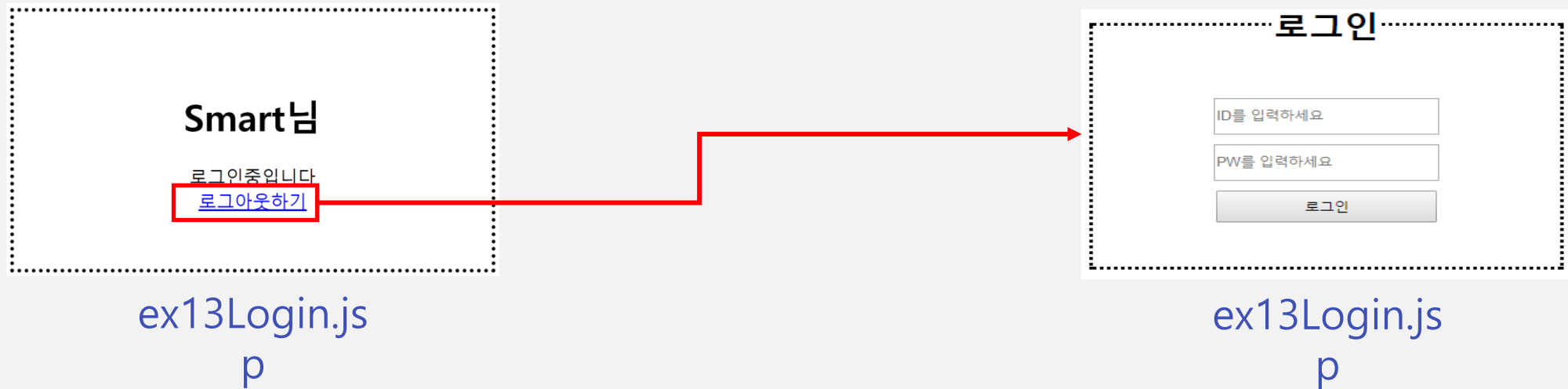
session 응용 실 습(3)



1. ex12Jointrue.jsp에서 세션영역에 nickname을 저장
2. ex12Join.jsp에서 세션영역을 확인하고 nickname이라는 id값이 있을 때 위와같이 출력하시오.

session 응용 실

습(3)



1. '로그아웃하기' 링크를 클릭했을 때 ex12Logout.java로 이동하게 하시오.
2. ex12Logout.java에서는 세션영역에 nickname을 삭제하고

121 : : 링크 이동하게 하시오

Client의 정보를 지속적으로 유지하기 위한 방법 Cookie & Session

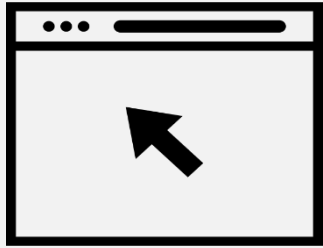
Cookie

전달할 데이터를 Web Browser(Client)로 보냈다가 Web Server로 되돌려 받는 방법

Session

Web Browser를 거치지 않고 Web Server에 있는 데이터 영역을 통해 데이터를 전달하는 방법

Cookie



Web browser

Client PC

Text

어려움

Session



Server PC

Object

강력함

Thank
you.