



JSP/Servlet

ch5. Cookie, Session

1. 쿠키와 세션의 필요성과 개념을 이해할 수 있다.
2. 쿠키와 세션을 비교할 수 있다.

쿠키와 세션은 왜 필요할까?



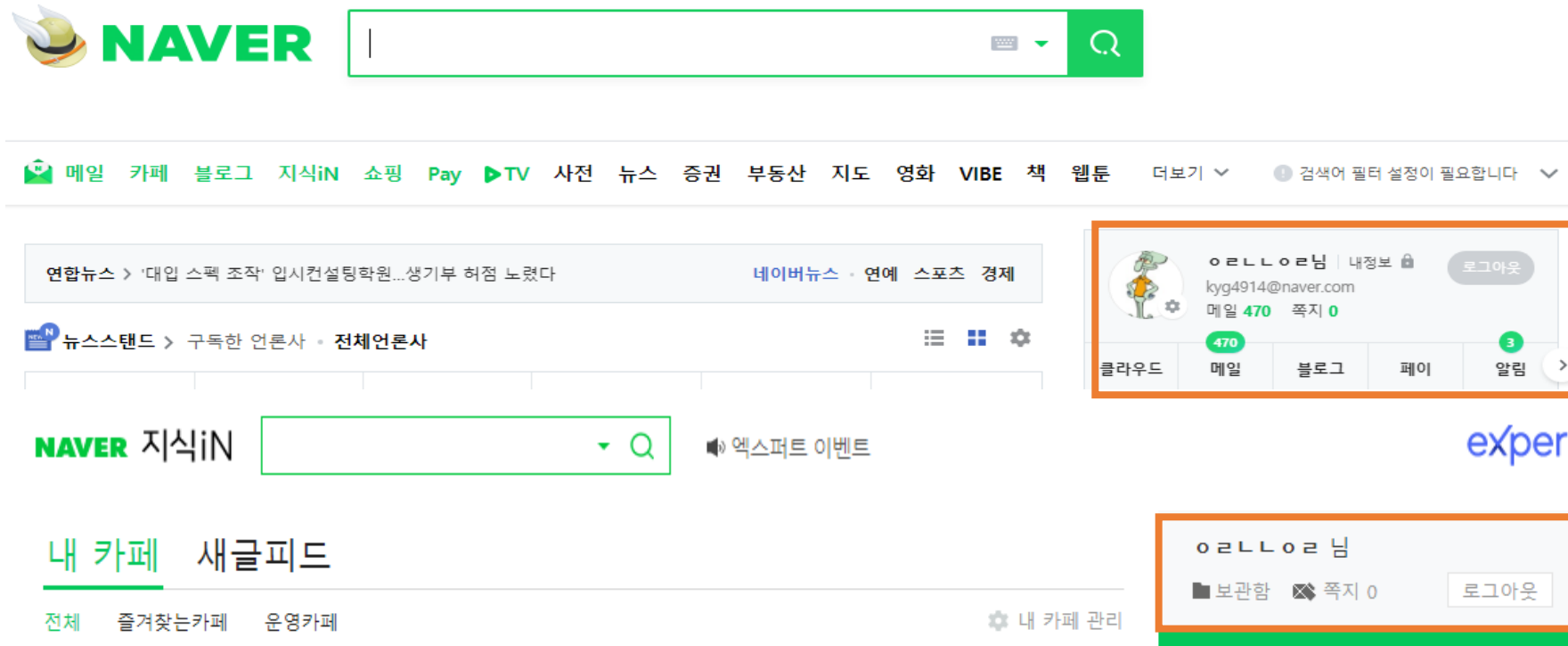
- connectionless(비연결성)

클라이언트가 요청을 한 후 응답을 받으면
그 연결을 끊어 버리는 특징

- stateless(무상태)

통신이 끝나면 상태를 유지하지 않는 특징

쿠키와 세션은 왜 필요할까?



로그인
상태 유지



쿠키와 세션을 사용하여 사용자의 상태를 유지할 수 있다

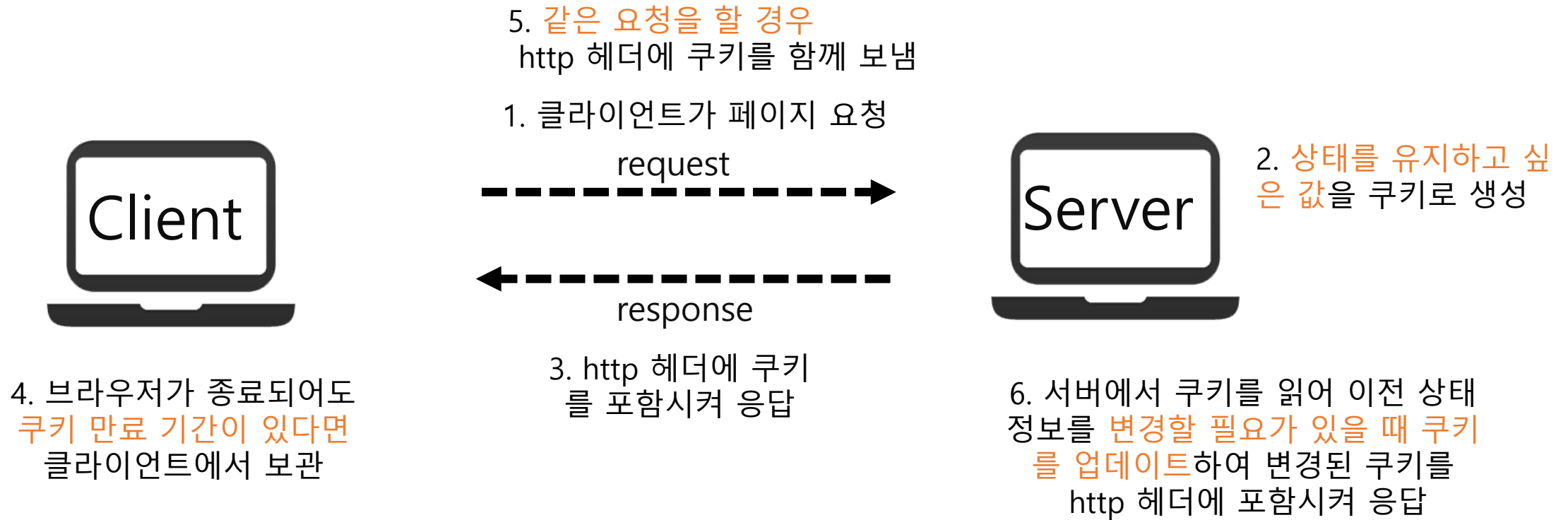
쿠키(Cookie)

Cookie(쿠키)

: 클라이언트(브라우저)에 저장되는 키와 값이 들어있는 작은 데이터 파일

Name	Value	Domain	Path	Expires / Max-Age	Size
NID_SAUTO	-1585137199	.nid.naver.c...	/	2020-11-06T11:41:40.420Z	20
nid_buk	KP5TSBAF3RGF6	.nid.naver.c...	/	2030-08-30T09:10:07.000Z	20
BMR		.naver.com	/	Session	3
NID_JKL	IBimHFueKgplAOydgjtj+bJR6SAkr1YPAqJCrpkDi/HM=	.naver.com	/	Session	51
NID_SES	AAABqOGRCXb+lza8kKzylvA6t06Eia9Lmr01GTmh31FAfjqksl4wf2zP...	.naver.com	/	Session	591
NID_AUT	B27+U/GfKTthyjU2x6aa4BVSXuNufTyntPZ7LZtO9tLCt1t9PX0rGGE7...	.naver.com	/	Session	71
nid_slevel	-1	.nid.naver.c...	/	2021-10-29T09:47:00.000Z	12
NM_THEME_EDIT		www.naver...	/	2030-10-27T10:43:02.000Z	13
nid_enctp	1	.nid.naver.c...	/	2021-01-27T09:47:06.939Z	10
NM_TOP_PROMOTION	1	www.naver...	/	2020-11-01T09:46:59.000Z	17
NNB	T4WWAQSIR2NF6	.naver.com	/	2050-01-01T09:00:00.282Z	16
NM_THEME_LAST_FIXED	SHOW	www.naver...	/	2020-10-29T11:37:33.000Z	23
NRTK	ag#all_gr#1_ma#-2_si#0_en#0_sp#0	.naver.com	/	2021-10-29T10:43:02.000Z	36

쿠키(Cookie) 동작방식



Cookie 사용 예

1. ID 저장, 로그인 상태 유지
2. 일주일간 다시 보지 않기
3. 쇼핑몰 장바구니 기능

Cookie 단점

1. 보안에 취약
2. 작은 허용 용량(4kb)
3. 다른 브라우저에서 저장한 값 사용불가

쿠키(Cookie) 사용

- 쿠키 사용하는 순서

- ① 쿠키를 생성.

- ```
Cookie cookie = new Cookie(String name, String value);
```

- ② 쿠키에 필요한 설정.

- 예) 쿠키 유효시간, 쿠키 설명, 도메인, 패스, 보안 정보등

- ```
cookie.setMaxAge(int);
```

- ③ 웹 브라우저에 생성된 쿠키를 전송.

- ```
response.addCookie(cookie);
```



# 쿠키 정보 생성 및 확인

## 1. 쿠키 정보 생성

<%

```
Cookie cookie = new Cookie("id", "smart");
response.addCookie(cookie);
response.addCookie(new Cookie("age", "22"));
response.addCookie(new Cookie("name", "홍길동"));
```

%>

createCookie.jsp

## 2. 웹브라우저에서 쿠키 정보 확인

웹브라우저 화면>우클릭> [검사] 실행

클릭

클릭

| 이름          | 값                 | D.   | Path        | Expires / Max-Age | 크기 | HttpO... |
|-------------|-------------------|------|-------------|-------------------|----|----------|
| name        | 홍길동               | I... | /webStudy04 | 세션                | 13 |          |
| age         | 22                | I... | /webStudy04 | 세션                | 5  |          |
| id          | smart             | I... | /webStudy04 | 세션                | 7  |          |
| JSESSION... | C2B76D008C482D... | I... | /webStudy04 | 세션                | 42 | ✓        |

id

이름

id

콘텐츠

smart

도메인

localhost

경로

/webStudy04

다음에 대해 보내기:

동일한 사이트에서 연결하는 경우에만

스크립트에 액세스 가능  
예

생성됨

2022년 8월 10일 수요일 오후 8:14:25

만료

탐색 세션이 종료될 때

# 쿠키 정보 조회

## 1. Request 객체를 이용한 쿠키 정보 조회

```
<table border>
<tr><th> 쿠키 이름 </th><th> 쿠키값 </th></tr>
<%
 Cookie [] cookies = request.getCookies();

 for(int i = 0; i < cookies.length; i++){
%>
 <tr><td><%=cookies[i].getName() %></td>
 <td><%=cookies[i].getValue() %></td></tr>
 <%
 }
%>
```

getCookie.jsp

| 쿠키 이름      | 쿠키값                              |
|------------|----------------------------------|
| JSESSIONID | C2B76D008C482DAE4AC4D24CABE2F173 |
| id         | smart                            |
| age        | 22                               |
| name       | 홍길동                              |

# 쿠키 정보 삭제

1. 쿠키 정보를 생성할 때에는 유효기간을 지정한다.
2. 유효기간을 0으로 하면 쿠키 정보는 즉시 삭제됨

<%

```
Cookie cookie = new Cookie("id", "");
```

expire.jsp

```
cookie.setMaxAge(0); //유효시간을 0으로 하면 쿠키 바로 삭제됨
response.addCookie(cookie);
```

%>

getCookie.jsp 실행 결과

| 쿠키 이름      | 쿠키값                              |
|------------|----------------------------------|
| JSESSIONID | C2B76D008C482DAE4AC4D24CABE2F173 |

# 쿠키로 관심 상품 정보 저장하기

<%

```
Cookie cookie = new Cookie("item1", URLEncoder.encode("서블릿", "UTF-8"));
cookie.setMaxAge(60*60); //쿠키를 한시간 유지
response.addCookie(cookie);
```

%>

book1.jsp

<%

```
Cookie cookie = new Cookie("item2", URLEncoder.encode("자바", "UTF-8"));
cookie.setMaxAge(60*60); //쿠키를 한시간 유지
response.addCookie(cookie);
```

%>

book2.jsp

<%

```
Cookie cookie = new Cookie("item3", URLEncoder.encode("인공지능", "UTF-8"));
cookie.setMaxAge(60*60); //쿠키를 한시간 유지
response.addCookie(cookie);
```

%>

book2.jsp

# 쿠키로 관심 상품 정보 목록 가져오기

```
<%
 response.setCharacterEncoding("UTF-8");
 //1. 쿠키 정보 가져오기
 Cookie [] cookies = request.getCookies();
 if(cookies != null){
 for(int i = 0; i < cookies.length; i++){
 if(cookies[i].getName().contains("item")){
 out.print(URLEncoder.decode(cookies[i].getValue(), "UTF-8") + "
");
 }
 }
 }
%>
```

bookList.jsp

## 관심 분야 책 리스트

서블릿  
자바  
인공지능

# 실습:쿠키로 장바구니 구현

shoppingBook.jsp

| 책 제목 | 가격    | 장바구니 담기                             |
|------|-------|-------------------------------------|
| 서블릿  | 22000 | <input checked="" type="checkbox"/> |
| 자바   | 24000 | <input type="checkbox"/>            |
| 파이썬  | 20000 | <input checked="" type="checkbox"/> |
| 머신러닝 | 28000 | <input type="checkbox"/>            |
| 영상처리 | 33000 | <input checked="" type="checkbox"/> |

장바구니 담기

장바구니 목록

shoppingBook.jsp

1. 장바구니 박스 체크 후 클릭  
장바구니 담기 버튼

shoppingCart.jsp

1. 선택 목록을 가져오기
2. 목록들에 대한 쿠키 생성
3. 모든 쿠키 유효기간 설정
4. Response객체에 쿠키 추가
5. sendRedirect("shoppingBook.jsp")

| 책 제목 | 가격    | 장바구니 담기                  |
|------|-------|--------------------------|
| 서블릿  | 22000 | <input type="checkbox"/> |
| 자바   | 24000 | <input type="checkbox"/> |
| 파이썬  | 20000 | <input type="checkbox"/> |
| 머신러닝 | 28000 | <input type="checkbox"/> |
| 영상처리 | 33000 | <input type="checkbox"/> |

장바구니 담기

장바구니 목록  
서블릿  
파이썬  
영상처리

shoppingBook.jsp

# Session(세션)

- : 브라우저가 종료되기전까지 클라이언트의 정보를 유지하게 해주는 기술
- : 사용자 정보 파일을 서버 측에서 관리

# 세션(Session) 동작방식

## Session id

5. 이후 서버에 요청할때 Session Id  
가 담긴 쿠키를 요청헤더에 추가하  
여 요청



4. 쿠키에 Session Id 저장

## Session id

1. 클라이언트가 페이지 요청

request

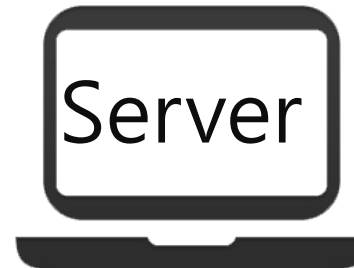


response



## Session Id

3. 응답헤더에 Session Id  
를 포함해서 전송



## Session Id

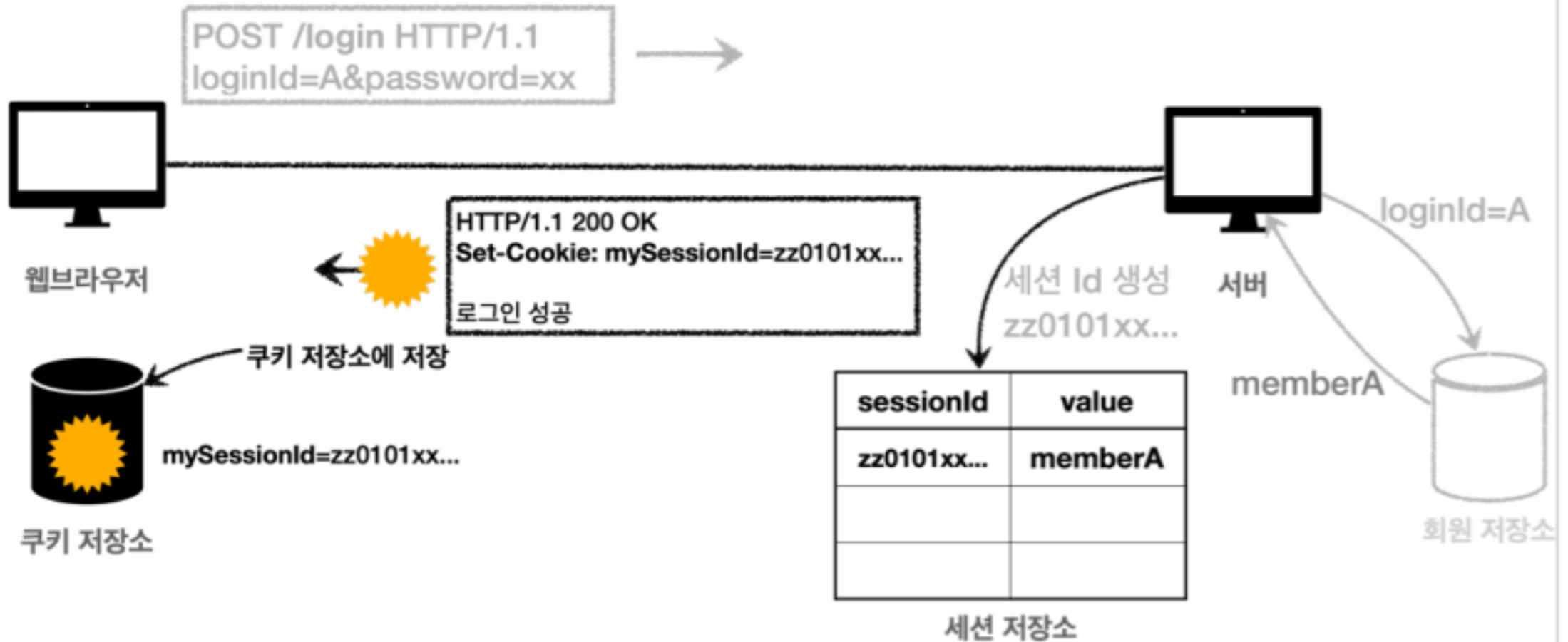
2. 해당 브라우저(클라이언트)에  
유일한 ID(Session ID)를 부여

6. Session Id를 확인하고 해당 세  
션에 관련된 정보를 확인한 후 응  
답



# 세션

## 세션id를 쿠키로 전달



## Session 사용 예

로그인 같이 보안상  
중요한작업을 수행할 때  
사용

## Session 특징

1. 각 client에게 고유 ID 부여
2. 보안 면에서 쿠키보다 우수
3. 사용자가 많아지면 서버 메모리많이 차지하게 됨

# 세션 메소드

## 메소드 : 리턴 타입

getAttribute(java.lang.String name) : java.lang.Object

getAttributeNames() : java.util.Enumeration

getCreationTime() : long

세션 생성된 시간 리턴(1/1000초 값으로 리턴)

getLastAccessedTime() : long

클라이언트가 마지막으로 request를 보낸 시간을 리턴

getId() : java.lang.String

getMaxInactiveInterval() : int

세션 유효시간 리턴

# 세션 메소드

invalidate() : void

세션 제거

removeAttribute(java.lang.String name) : void

setAttribute(java.lang.String name, java.lang.Object value) : void

setMaxInactiveInterval(int interval) : void

세션 유효 시간 설정(단위= 초)

웹서버의 세션 기본 유효 시간: 30분(web.xml)

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

# 세션 정보 생성 및 확인

## 1. 세션 정보 생성

```
<%
 session.setAttribute("id", "smart");
 session.setAttribute("name", "홍길동");
%>
```

createSession.jsp

## 2. 세션 정보 확인

```
<%
 String id = (String)session.getAttribute("id");
 String name = (String)session.getAttribute("name");
%>
id: <%=id %>

name: <%=name %>
```

getSession.jsp

# 세션 메소드 활용

```
<%
String id_str = session.getId();
long lastTime = session.getLastAccessedTime();
long createdTime = session.getCreationTime();
long timeUsed = (lastTime - createdTime) / 60000;
int inactive = session.getMaxInactiveInterval() / 60;
boolean b_new = session.isNew();
```

```
%>
세션 ID: <%=id_str %>

웹사이트에 머문 시간: <%=timeUsed %>분

세션 유효시간: <%=inactive %>분

```

```
<%
if(b_new)
out.print("새로운 세션입니다.");
else out.print("기존 세션입니다.");
%>
```

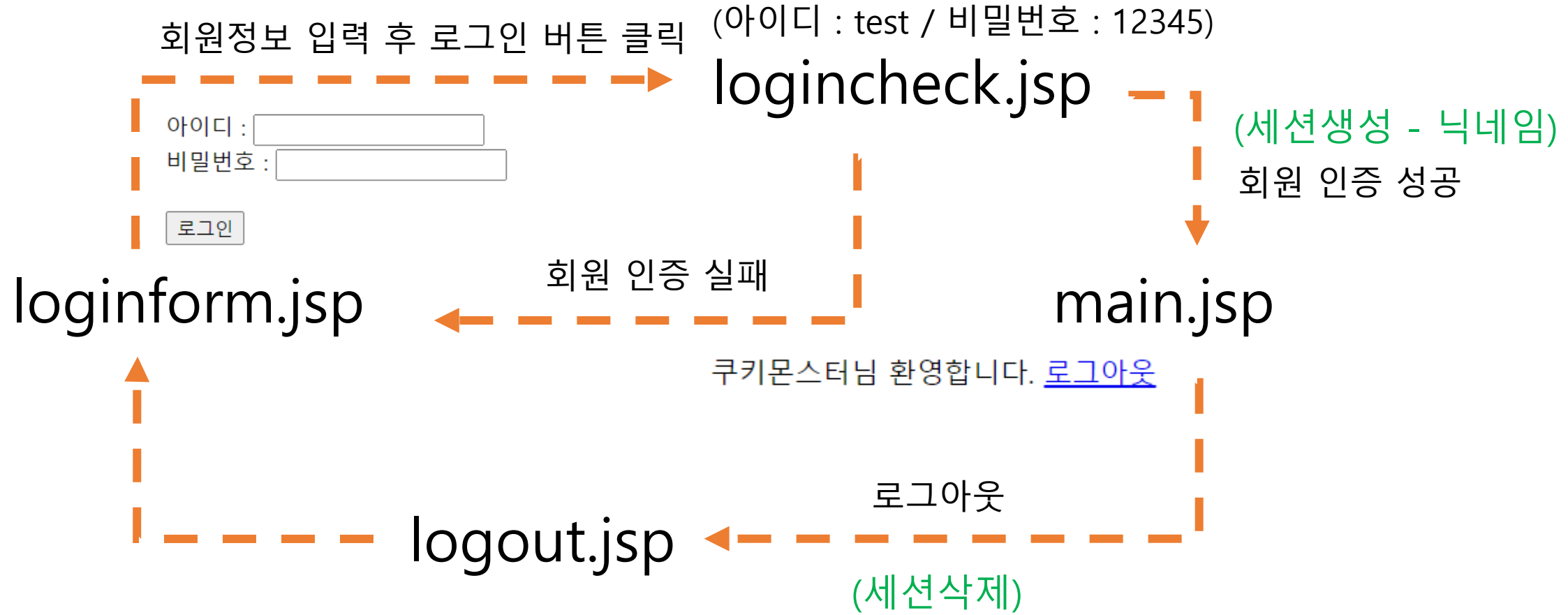
세션 ID: E1AC5C99B9957CB865192EFB564B841F  
웹사이트에 머문 시간: 0분  
세션 유효시간: 30분  
새로운 세션입니다.

새 창으로 연결

세션 ID: BB68EEE99902DC5D904CC381467D2376  
웹사이트에 머문 시간: 1분  
세션 유효시간: 30분  
기존 세션입니다.

새탭에서 연결할 경우

# 세션(Session) 활용 실습



# Cookie vs Session

	Cookie(쿠키)	Session(세션)
사용자 정보 저장 위치	클라이언트	서버
보안	상대적으로 불안전	상대적으로 안전
속도	상대적으로 빠름	상대적으로 느림
생명주기	만료일이 되면 삭제	브라우저 종료 시 삭제



# Session 이용한 로그인 구현시 주의사항

---

- 로그인 체크 후 세션 생성, main페이지로 redirect 할 때
- main 페이지에서는 반드시 세션이 존재하는지 확인한 후 main 페이지를 보여준다.
- 세션이 없다면 방문자는 침입자이다.