

JSP 3-1

JSP 2-4

JSP - 쿠키(Cookie) & 세션(Session)

쿠키 (Cookie)

클라이언트가 서버에 요청을 보낼 때마다 클라이언트 정보나 상태를 기억하기 위한 방법

- HTTP는 **비연결성(Stateless)** → 응답 후 연결 종료됨
- 상태 유지를 위해 **서버가 클라이언트에 정보를 저장**하는 방식이 쿠키
- **4KB 용량 제한, 도메인 당 약 300개 저장 가능**

쿠키 생성 절차

쿠키 생성 → 속성 설정 → `response` 객체에 추가 → 웹 브라우저 저장

- `javax.servlet.http.Cookie` 객체 생성
- 쿠키에 속성 설정 (`name`, `value`, `maxAge`, `path` 등)
- `response.addCookie()` 로 클라이언트에 전송

주요 메소드

메소드	설명
<code>setMaxAge(int seconds)</code>	쿠키 유효 기간 설정
<code>setPath(String path)</code>	쿠키가 유효한 디렉토리 설정
<code>setValue(String value)</code>	쿠키의 값 설정
<code>getMaxAge()</code>	유효 기간 조회
<code>getPath()</code>	유효한 경로 조회
<code>getName()</code>	쿠키 이름 조회
<code>getValue()</code>	쿠키 값 조회

 브라우저 닫을 시 삭제 여부는 **유효시간 설정 유무**에 따라 달라짐

```

1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta charset="UTF-8">
7 <title>Insert title here</title>
8 </head>
9 <body>
10
11 <%
12 Cookie cookie = new Cookie("cookieName", "cookieValue");
13 //쿠키 생성 new Cookie("쿠키이름", "쿠키 값")
14
15 cookie.setMaxAge(60*60); //1시간동안 쿠키 유지
16 //쿠키는 보통 유효시간을 정해주지 않으면 웹 브라우저를 닫을 때 쿠키가 삭제
17 //그러나 유효시간을 설정해놓으면 유효시간 동안 쿠키가 존재
18 //웹 브라우저를 닫아도 유효시간이 남아있으면 쿠키는 삭제되지 않고 남아있게된다
19
20 //cookie.setValue("newCookieValue"); 쿠키값을 변경한다
21
22 response.addCookie(cookie);
23 //addCookie 메소드로 cookie객체를 웹 브라우저에 저장한다
24
25 %>
26
27 <h2>쿠키가 합쳐되었습니다.</h2>
28 <a href="cookieGet.jsp">쿠키 보기</a>
29
30 </body>
31 </html>

```

```

1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta charset="UTF-8">
7 <title>Insert title here</title>
8 </head>
9 <body>
10
11 <%
12 Cookie[] cookies = request.getCookies();
13 //쿠키 객체들을 배열로 받는다
14 for(int i = 0; i < cookies.length; i++)
15 {
16     //getName(): 쿠키 이름을 얻는다
17     //getValue(): 쿠키 값을 얻는다
18
19     String str = cookies[i].getName();
20     if(str.equals("cookieName"))
21     {
22         out.print("쿠키이름: " + cookies[i].getName() + "<br>");
23         out.print("쿠키 값: " + cookies[i].getValue() + "<br>");
24     }
25 }
26 %>
27
28 <hr>
29 <a href="cookieDelete.jsp">쿠키 삭제</a>
30
31 </body>
32 </html>

```

```

1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta charset="UTF-8">
7 <title>Insert title here</title>
8 </head>
9 <body>
10
11 <%
12 Cookie[] cookies = request.getCookies();
13 for(int i = 0; i < cookies.length; i++)
14 {
15     String str = cookies[i].getName();
16     if(str.equals("cookieName"))
17     {
18         cookies[i].setMaxAge(0);
19         //쿠키의 유효시간을 0으로 설정
20
21         response.addCookie(cookies[i]);
22         //변경된 쿠키객체를 웹 브라우저에 저장
23
24         out.print("<h2>" + cookies[i].getName() + " 쿠키가 삭제되었습니다.</h2>");
25     }
26 }
27 %>
28
29 <hr>
30 <a href="cookieTest.jsp">쿠키 확인</a>
31
32 </body>
33 </html>

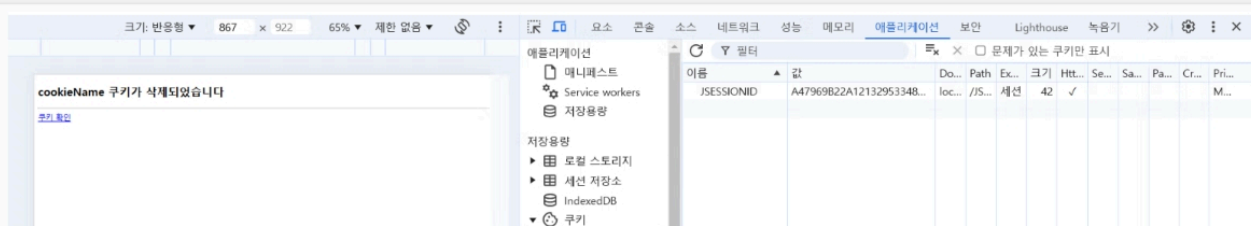
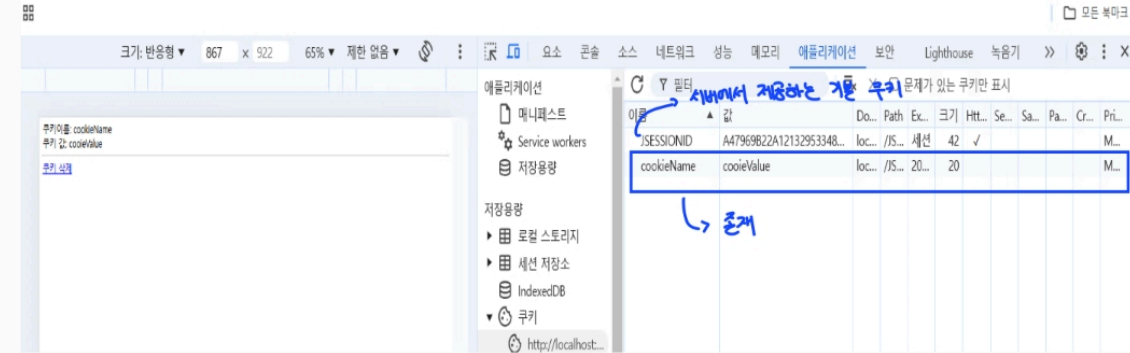
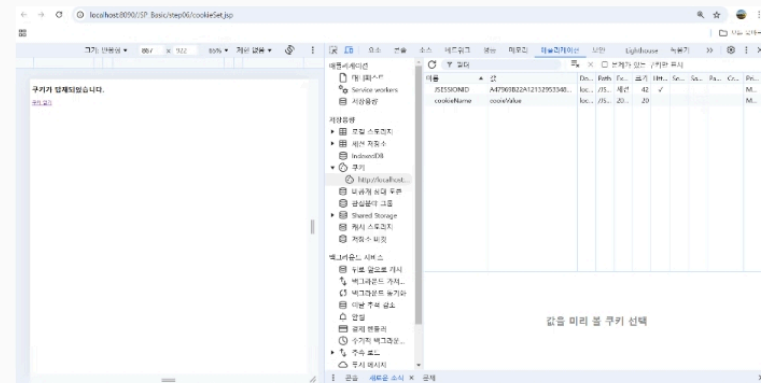
```

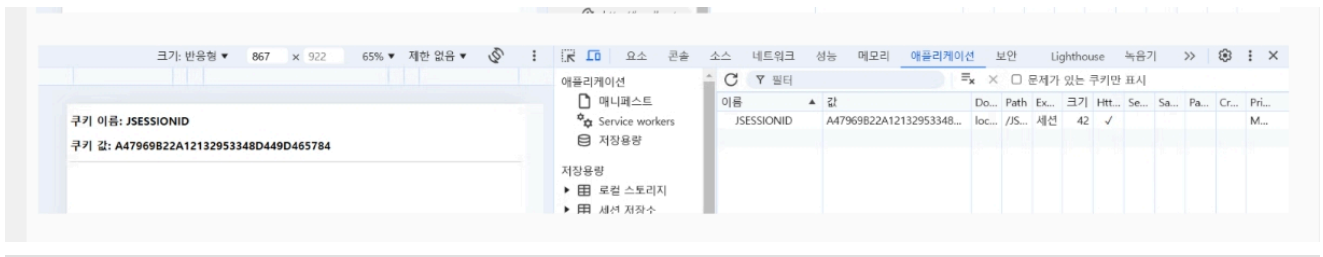
우제하려는 쿠키
유효시간 0으로 설정하고
웹 브라우저에 저장

```

1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta charset="UTF-8">
7 <title>Insert title here</title>
8 </head>
9 <body>
10
11 <%
12 //쿠키 배열을 얻은 후 모든 쿠키의 이름과 값을 웹 브라우저에 출력하기
13 Cookie[] cookies = request.getCookies();
14
15 if(cookies != null)
16 {
17     for(int i = 0; i < cookies.length; i++)
18     {
19         out.print("<h2>쿠키 이름: " + cookies[i].getName() + "</h2>");
20         out.print("<h2>쿠키 값: " + cookies[i].getValue() + "</h2>");
21         out.print("<hr>");
22     }
23 }
24
25 %>
26
27 </body>
28 </html>

```





세션 (Session)

서버가 클라이언트와의 연결 상태를 **서버 내부에 객체로 저장**하여 유지하는 방식

- **보안성 높음** (클라이언트 저장 X)
- **서버 메모리에 저장**되므로 저장 용량 제한 없음
- 브라우저 최초 요청 시 세션 자동 생성

클라이언트 요청 → 세션 자동 생성 → 속성 설정 → 세션 ID로 관리

주요 메소드

메소드	설명
<code>setAttribute(String name, Object value)</code>	세션 속성 설정
<code>getAttribute(String name)</code>	세션 속성 값 반환 (없으면 null)
<code>getAttributeNames()</code>	모든 세션 속성 이름 목록 반환 (Enumeration)
<code>getId()</code>	세션에 할당된 고유 식별자 반환

```
sessionSet.jsp x sessionGet.jsp
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6   <meta charset="UTF-8">
7   <title>Insert title here</title>
8 </head>
9 <body>
10 <%
11
12 session.setAttribute("sessionName", "sessionValue");
13 session.setAttribute("sessionNumber", 12345);
14 //세션객체는 jsp에서 생성하지 않아도 사용할 수 있는 내장 객체이다.
15 //setAttribute("세션이름", 세션값);
16 //이름은 String 타입이고 값은 Object 타입이다.
17 %>
18
19 <h2>세션이 생성되었습니다</h2>
20 <hr>
21 <a href="sessionGet.jsp">세션 얻기</a>
22 </body>
23 </html>
```

```
sessionSet.jsp x sessionGet.jsp
5 <html>
6 <head>
7   <meta charset="UTF-8">
8   <title>Insert title here</title>
9 </head>
10 <body>
11 <%
12 String sessionNameValue = (String)session.getAttribute("sessionName");
13 //getAttribute 메소드의 반환타입은 Object타입이다
14
15 out.print("sessionName값: " + sessionNameValue + "<br>");
16
17 int sessionNumberValue = (int)session.getAttribute("sessionNumber");
18 out.print("sessionNumber 값: " + sessionNumberValue + "<br>");
19
20 out.print("<hr>");
21
22 String sName, sValue;
23
24
25 Enumeration<String> e = session.getAttributeNames();
26
27 while(e.hasMoreElements())
28 {
29   sName = e.nextElement();
30   sValue = session.getAttribute(sName).toString();
31   out.print("세션이름: " + sName + "<br>");
32   out.print("세션 값: " + sValue + "<br>");
33 }
34 %>
35 </body>
36 </html>
```

반과다양 Objec
다양변환 해야함
이름으로 value 리턴.



```

42 int sessionInter = session.getMaxInactiveInterval();
43 //세션의 유효시간 (기본 30분)
44 out.print ("세션의 유효시간: " + sessionInter + "<br>");
45 out.print("<hr>");
46
47
48 session.removeAttribute("sessionName");
49 //세션 삭제: removeAttribute("세션이름");
50
51 e = session.getAttributeNames();
52 while(e.hasMoreElements())
53 {
54     sName = e.nextElement();
55     sValue = session.getAttribute(sName).toString();
56     out.print("세션이름: " + sName+"<br>");
57     out.print("세션 값: " + sValue+"<br>");
58 }
59
60 out.print("<hr>");
61 session.invalidate();
62 //세션 무효화
63 //세션의 모든 데이터를 삭제한다
64
65 if(request.isRequestedSessionIdValid())
66 {
67     //유효한 세션이 있으면 true, 없으면 false
68     out.print("유효한 세션이 있음");
69 }
70 else
71 {
72     out.print("유효한 세션이 없음");
73 }

```

→ 세션의 유효시간 확인

→ 세션 삭제

→ 삭제 후 세션확인

→ 세션 무효화

localhost:8090/JSP_Basic/step07/sessionGet.jsp

sessionName값: sessionValue
sessionNumber 값: 12345

세션이름: sessionNumber
세션 값: 12345
세션이름: sessionName
세션 값: sessionValue

세션 아이디: A47969822A12132953348D449D465784
세션의 유효시간: 1800

세션이름: sessionNumber
세션 값: 12345

유효한 세션이 없음

```

String sessionID = session.getId();
//한 브라우저당 고유한 아이디가 생성이 된다
//브라우저별로 생성되어진 세션 구분할 수 있다
out.print("세션 아이디: " + sessionID + "<br>");

```

세션 유효시간확인