

# JSP & Servlet #7

## – 세션 & 쿠키

### 목 차

1. 세션 트래킹
2. 세션 사용하기
3. 쿠키 사용하기

# Readings



## Readings:

### Chapter 6:

대화 상태 유지하기 : 세션 관리

## Objective

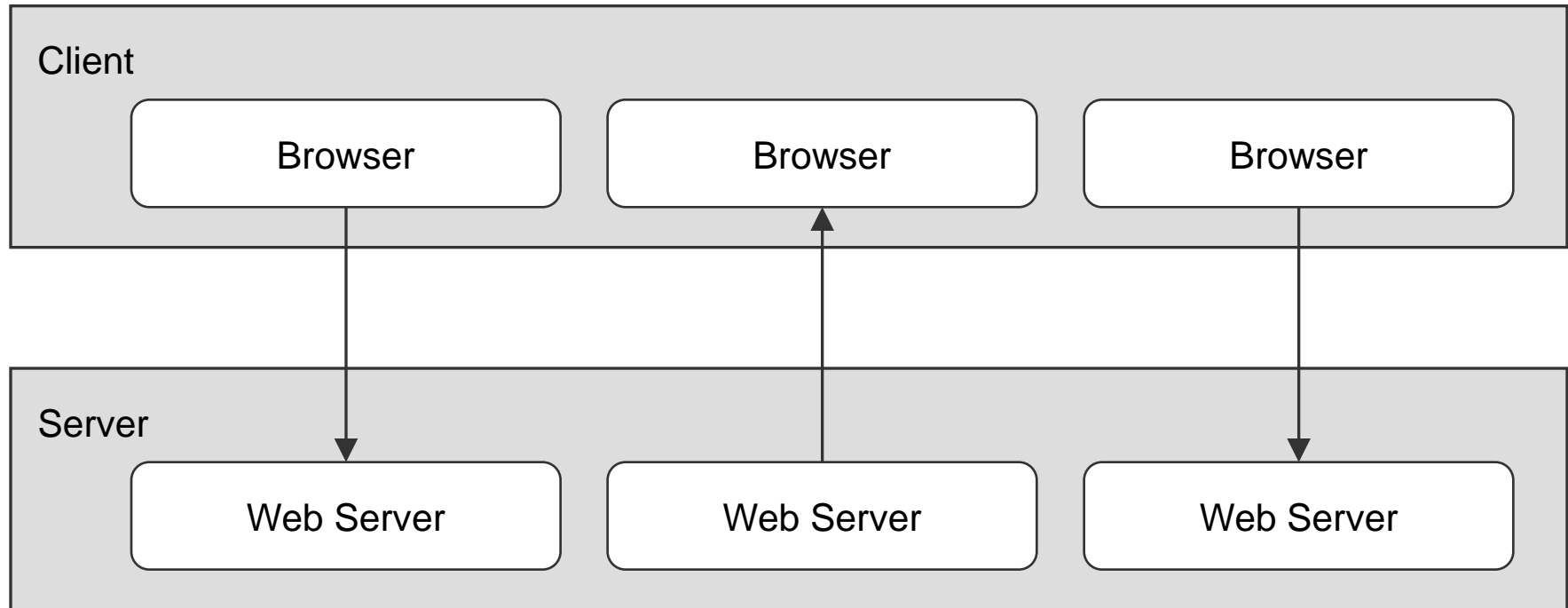
- ❑ 쿠키와 **URL** 인코딩을 활용하여 세션 트래킹을 하는 웹 애플리케이션을 개발할 수 있다.
- ❑ 브라우저로부터 특정 쿠키의 정보를 얻는 유틸리티 클래스를 작성할 수 있다.
- ❑ 세션 트래킹을 위해 쿠키를 사용하는 방식을 설명 할 수 있다.
- ❑ 세션 트래킹을 위해 **URL** 인코딩이 동작하는 방식을 설명 할 수 있다.

# **1.세션 트래킹**

# 1. 세션 트래킹

## I. 세션 트래킹

### □ HTTP로 세션 트래킹이 어려운 이유



#### 최초 HTTP 요청 :

브라우저에서 웹 페이지를 요청한다.

#### 최초 HTTP 응답 :

서버는 요청된 페이지를 보내고 연결을 끊는다.

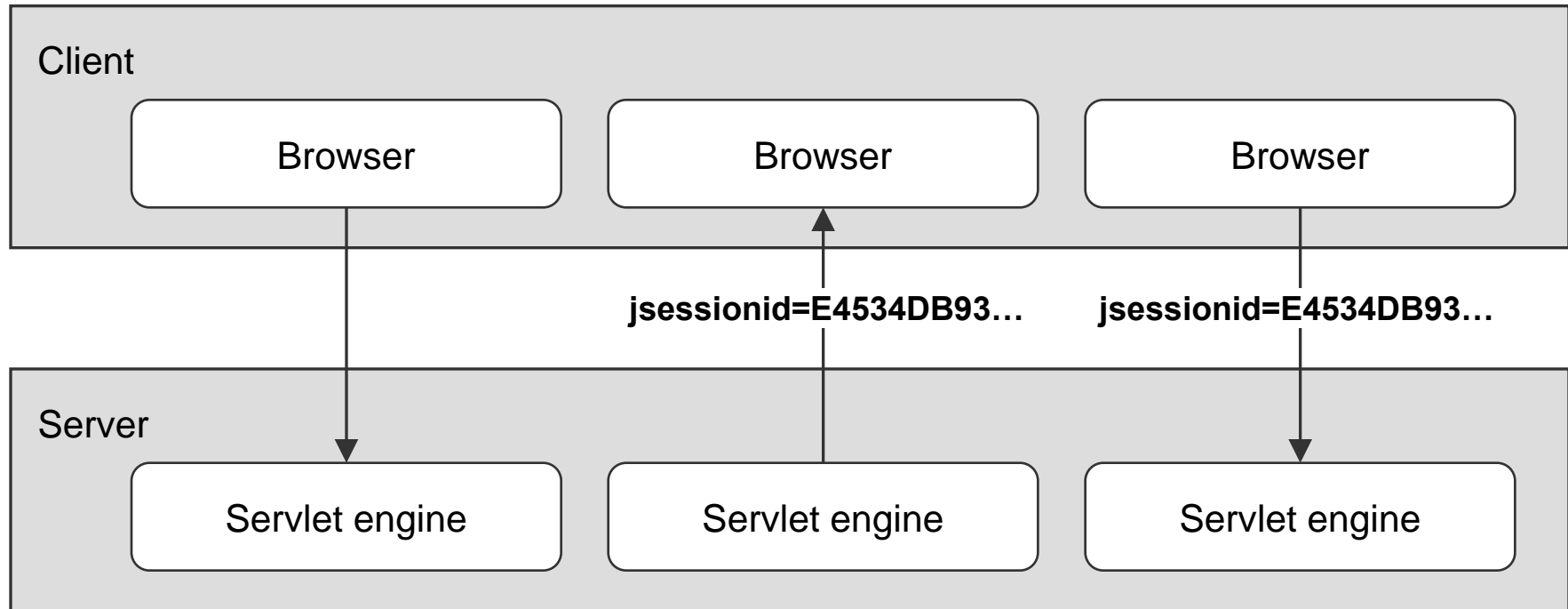
#### 이후의 HTTP 요청 :

브라우저에서 웹 페이지를 요청한다. 서버는 브라우저가 이전에 요청했던 내용을 알지 못한다.

# 1. 세션 트래킹

## I. 세션 트래킹

### □ 서블릿 엔진에서 세션을 유지하는 방법



#### 최초 HTTP 요청 :

브라우저에서 JSP 또는 서블릿을 요청한다. 서블릿 엔진은 세션 객체를 생성하고 ID를 할당한다.

#### 최초 HTTP 응답 :

서버는 요청된 페이지와 세션에 할당된 ID를 함께 보낸다.

#### 이후의 HTTP 요청 :

브라우저에서 웹 페이지를 요청한다. 서블릿 엔진은 세션ID를 이용해 브라우저에 할당된 세션 객체를 확인한다.

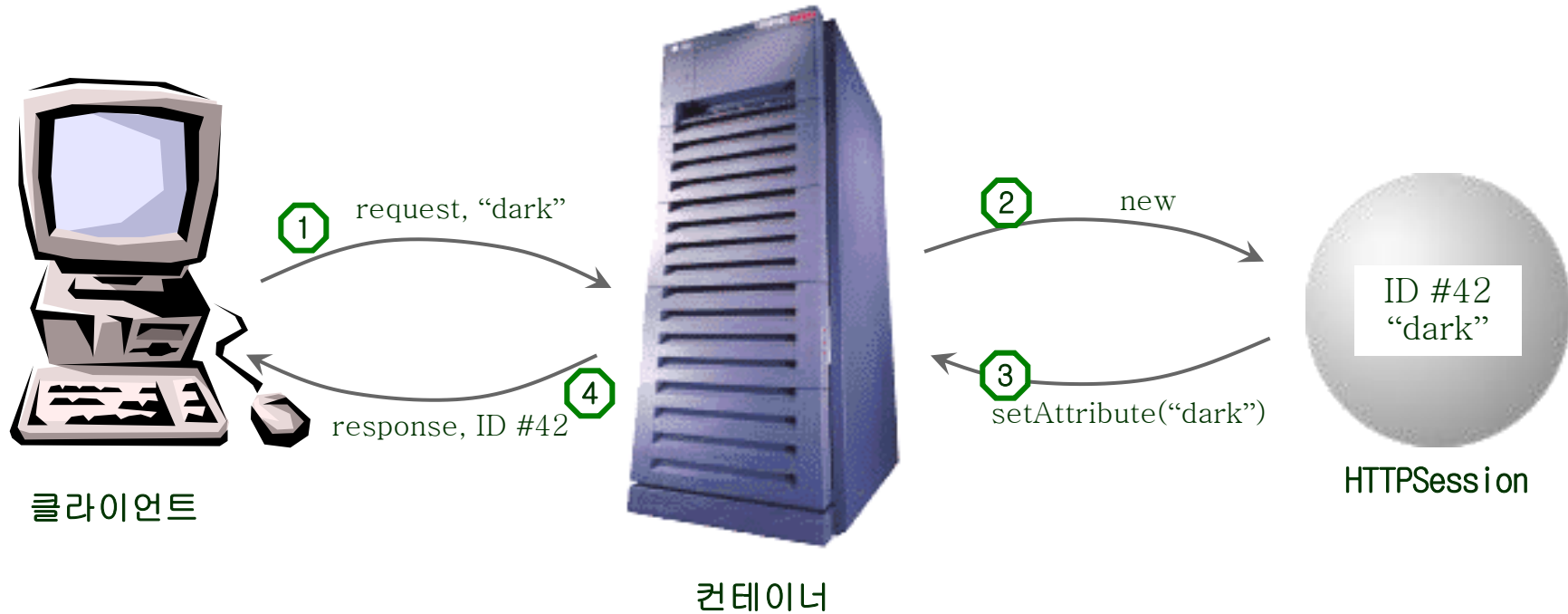
### □ 세션 트래킹 개요

- HTTP는 *무상태 프로토콜(stateless protocol)*이다. 브라우저가 요청을 하고 응답을 받은 뒤 연결을 끊는다. 그러므로 *상태(state)*를 유지하기 위해선 *세션 트래킹(session tracking)*을 해야 한다.
- 서블릿 API는 각 브라우저에 세션 ID를 저장하기 위해 *쿠키(cookie)*를 사용한다. 브라우저가 서버에 요청을 보낼 때마다 쿠키를 함께 전달한다.
- 브라우저에서 쿠키를 사용하지 못하게 설정하는 경우가 있다. 이런 경우에는 웹 페이지의 URL에 세션 ID를 저장하기 위해 URL 인코딩(URL encoding)을 사용할 수 있다.
- 서버는 각 세션에 대한 데이터를 저장하기 위해서 *세션 객체(session object)*를 사용한다.

# 1. 세션 트래킹

## I. 세션 트래킹

### □ 세션 처리 흐름





## **2.세션 사용하기**

# 1. 세션 객체의 기본 메소드

## II. 세션 사용하기

### □ 세션 객체에 관련한 메소드

- 세션 객체 얻기 (request 객체의 메소드)

```
public HttpSession getSession()
```

- 세션 객체의 기본 메소드

```
public void setAttribute(String name, Object value)
```

```
public Object getAttribute(String name)
```

```
public void removeAttribute(String name)
```

### □ 세션 객체에 관련한 Java 코드

- 세션 객체를 얻는 코드

```
HttpSession session = request.getSession();
```

- 세션에 String 객체를 속성(attribute)으로 설정하는 코드

```
session.setAttribute("productCode", productCode);
```

- 세션에 사용자가 정의한 객체를 속성으로 설정하는 코드

```
Cart cart = new Cart(productCode);  
session.setAttribute("cart", cart);
```

- 세션에서 String 객체를 얻는 코드

```
String productCode = (String) session.getAttribute("productCode");
```

- 세션에서 사용자가 정의한 객체를 얻는 코드

```
Cart cart = (Cart) session.getAttribute("cart");  
if (cart == null)  
    cart = new Cart();
```

- 세션에 설정한 속성(attribute)을 삭제하는 코드

```
session.removeAttribute("productCode");
```

### □ 세션 속성(session attribute) 사용하기

- 세션 객체는 JSP 내장객체이기 때문에, JSP내에서 별도로 세션 객체를 생성할 필요는 없다.
- 브라우저가 사이트에 처음 요청을 하는 시점에 세션 객체가 생성된다.
- 추가적인 요청 없는 상태로 일정 시간이 지나거나 사용자가 브라우저를 종료하는 경우에 세션이 종료된다.
- `HttpServletRequest` 객체의 `getSession` 메소드는 세션 객체에 접근하도록 한다. 이 메소드는 세션 객체가 존재하지 않는 경우에만 세션 객체를 생성한다.

### □ 세션 객체의 추가 메소드

- `public Enumeration getAttributeNames()`

세션 객체에 할당되어있는 모든 속성들의 이름을 `Enumeration` 타입으로 리턴한다.

- `public String getId()`

서블릿 엔진이 세션에 할당한 ID 값을 리턴한다.

- `public boolean isNew()`

클라이언트가 아직 세션에 관하여 알지 못하는 상황, 즉 세션이 바로 만들어진 상태이면 `true`를 리턴한다.

- `public void setMaxInactiveInterval(int seconds)`

해당 세션에 대한 요청과 요청간의 최대 허용 시간(초 단위)을 지정한다.

- `public void invalidate()`

세션을 종료한다. 현재 세션에 저장된 모든 세션 속성을 제거하는 작업이 포함된다.

### □ 세션 객체 사용 예

- 세션의 모든 속성 이름을 얻는 메소드

```
Enumeration names = session.getAttributeNames();  
while (names.hasMoreElements()) {  
    System.out.println((String) names.nextElement());  
}
```

- 세션의 ID를 얻는 메소드

```
String jSessionId = session.getId();
```

- 세션의 최대 허용 시간을 지정하는 메소드

```
session.setMaxInactiveInterval(60*60*24);    // 하루  
session.setMaxInactiveInterval(-1);         // 브라우저를 끄기 전까지
```

- 현재 세션에 저장된 모든 세션 속성을 제거하고 세션을 종료하는 메소드

```
session.invalidate();
```

### 3. URL 인코딩

## II. 세션 사용하기

#### ❑ encodeURL 메소드 문법 (response 객체)

```
public String encodeURL(String url)
```

#### ❑ URL 인코딩 방법

- JSP에서 서블릿 호출

```
<%
```

```
    String encodedURL = response.encodeURL("/servlet/CartServlet");
```

```
%>
```

```
<form action="<%=encodedURL%>" method="post">
```

- JSP에서 서블릿 호출 - 파라미터 포함

```
<a href="
```

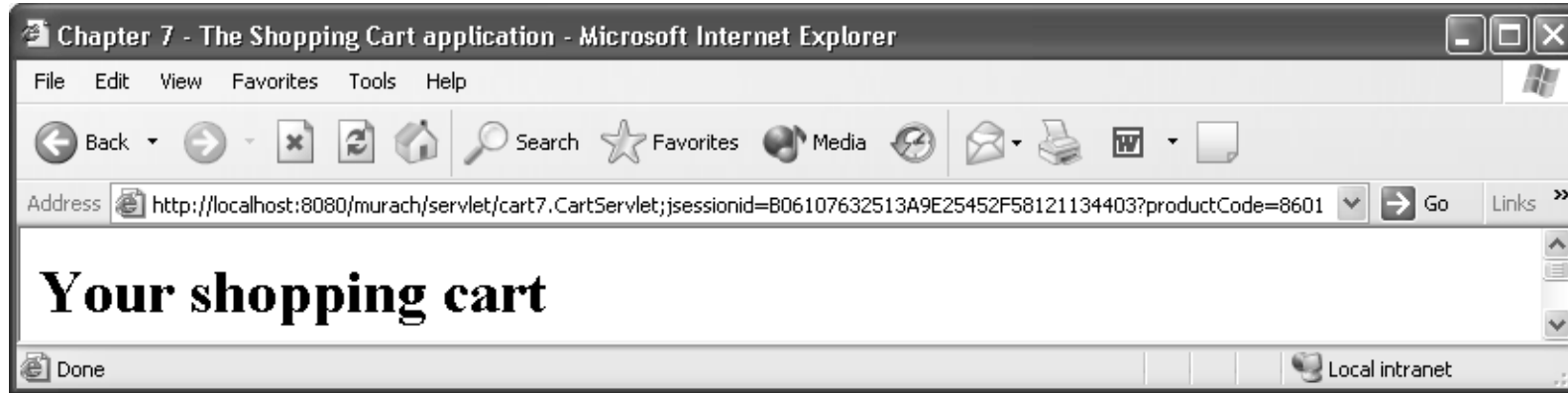
```
    <%=response.encodeURL("/servlet/CartServlet?productCode=jr01") %>
```

```
"> Add to Cart </a>
```

### 3. URL 인코딩

## II. 세션 사용하기

#### □ 인코딩 처리된 URL



#### □ 쿠키 없이 세션을 유지하기 위한 URL 인코딩

- 사용자가 쿠키를 사용 못하게 설정해 놓았다면 URL 인코딩(URL encoding)을 이용하여 세션에 대한 ID를 유지할 수 있다. 이를 위해서 모든 HTML 페이지를 JSP로 변환해야 하고, 연관된 모든 URL을 인코딩해야 한다.
- URL을 인코딩하는 경우에 세션ID는 URL에 포함되어 브라우저에 전달된다.



### **3.쿠키 사용하기**

# 1. 쿠키 개요

## III. 쿠키 사용하기

### □ 쿠키 예

jsessionid=D1F15245171203E8670487F020544490

user\_id=87

email=jsmith@hotmail.com

userName=jsmith

passwordCookie=opensesame

### □ 쿠키 동작 방식

- 쿠키는 이름/값(name/value) 쌍으로 브라우저에 저장된다.
- 서버 측에서는 웹 애플리케이션이 쿠키를 생성하고 브라우저로 전달한다. 클라이언트 측에서는 브라우저가 쿠키를 저장하고 웹 페이지에 접근할 때마다 쿠키를 서버로 전달한다.
- 쿠키는 브라우저에 3년까지 유지될 수 있다.
- 브라우저에서 쿠키 사용을 금지 시킬 수 있기 때문에, 모든 사용자의 쿠키가 사용 가능하다고 전제할 수 없다.
- 쿠키는 하나 이상의 서브도메인 명과 연관 지어진다.

### □ 전형적인 쿠키 사용

- 사용자가 로그인 또는 입력 폼(사용자명, 패스워드, 주소, 신용카드 정보 등)을 건너 뛸 수 있도록 한다.
- 사용자에게 개인화된 페이지를 제공한다. - 날씨, 시험 결과, 주식 시세 등
- 사용자에게 맞는 타겟 광고 등에 사용한다.

### □ 쿠키

- 일반적으로 쿠키는 브라우저와 세션 객체를 연결하기 위해 사용된다.
- 추가적으로 쿠키를 생성하여 사용자의 브라우저에 전달할 수 있다. 이 쿠키들을 이용하여 개별 사용자에게 특화된 정보들을 유지할 수 있다.

## 2. 쿠키 생성 및 사용

## III. 쿠키 사용하기

### □ 생성자 및 메소드

#### – 쿠키 클래스의 생성자

- `public Cookie(String cookieName, String cookieValue)`

#### – 쿠키 클래스의 메소드

- `public void setMaxAge(int lifeInSeconds)`
- `public void setPath(String path)`
- `public String getName()`
- `public String getValue()`

#### – `response` 객체의 메소드

- `public void addCookie(Cookie cookieName)`

#### – `request` 객체의 메소드

- `public Cookie[] getCookies()`

## 2. 쿠키 생성 및 사용

## III. 쿠키 사용하기

### □ 쿠키 사용 예

- 쿠키를 생성하고 값을 지정하는 코드

```
Cookie userIdCookie = new Cookie("userIdCookie", userId);
userIdCookie.setMaxAge(60*60*24*365*2);
userIdCookie.setPath("/");
response.addCookie(userIdCookie);
```

- 쿠키로부터 값을 얻는 코드

```
Cookie[] cookies = request.getCookies();
String cookieName = "userIdCookie";
String cookieValue = "";
for (int i=0; i<cookies.length; i++) {
    Cookie cookie = cookies[i];
    if (cookieName.equals(cookie.getName()))
        cookieValue = cookie.getValue();
}
```

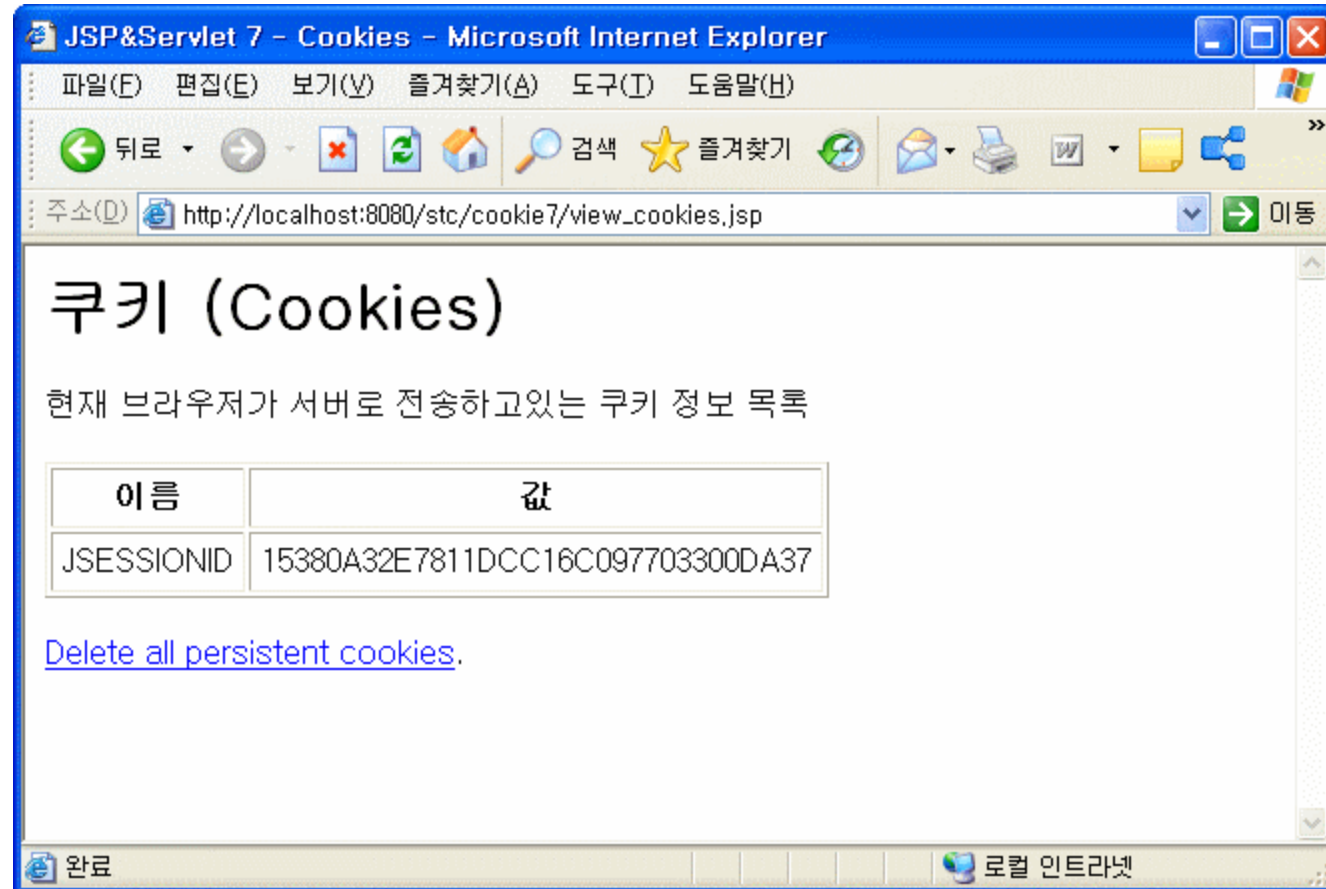
### □ 쿠키 생성 및 사용 방법

- 지속적인 쿠키를 생성하려면 `setMaxAge` 메소드로 양수의 값을 갖도록 한다.  
예를 들어 쿠키의 `age` 값을  $60 \times 60 \times 24 \times 365$ 로 지정했다면, 쿠키는 1년 동안 유효하게 된다.  
(60초 \* 60분 \* 24시간 \* 365일)
- 세션 동안만 유효한 쿠키를 생성하려면 쿠키의 `age` 값을 -1로 지정하면 된다.  
그러면 사용자가 브라우저를 종료하는 시점에 쿠키가 삭제된다.
- 모든 애플리케이션에서 쿠키에 접근하는걸 허용하려면 `setPath` 메소드를 이용하여 `path`를 "/" 로 지정한다.

### 3. 쿠키 확인 및 삭제

### III. 쿠키 사용하기

#### □ 모든 쿠키 정보를 보여주는 화면





## 3. 쿠키 확인 및 삭제

## III. 쿠키 사용하기

### □ 쿠키 값을 확인하는 코드

```
<% Cookie[] cookies = request.getCookies();
    for (int i=0; i<cookies.length; i++) {
        Cookie cookie = cookies[i];

%>

<tr>
    <td align="right"><%= cookie.getName() %></td>
    <td><%= cookie.getValue() %></td>
</tr>
<% } %>
```

### □ 쿠키를 삭제하는 코드

```
<%
    Cookie[] cookies = request.getCookies();
    for (int i=0; i<cookies.length; i++) {
        Cookie cookie = cookies[i];
        cookie.setMaxAge(0);    //쿠키를 삭제한다.
        cookie.setPath("/");    //모든 애플리케이션이 접근하도록 허용한다.
        response.addCookie(cookie);
    }
%>
```

## 4. 쿠키 관련 유틸리티 클래스

### III. 쿠키 사용하기

#### ❑ 쿠키로부터 값을 얻는 유틸리티 클래스 코드

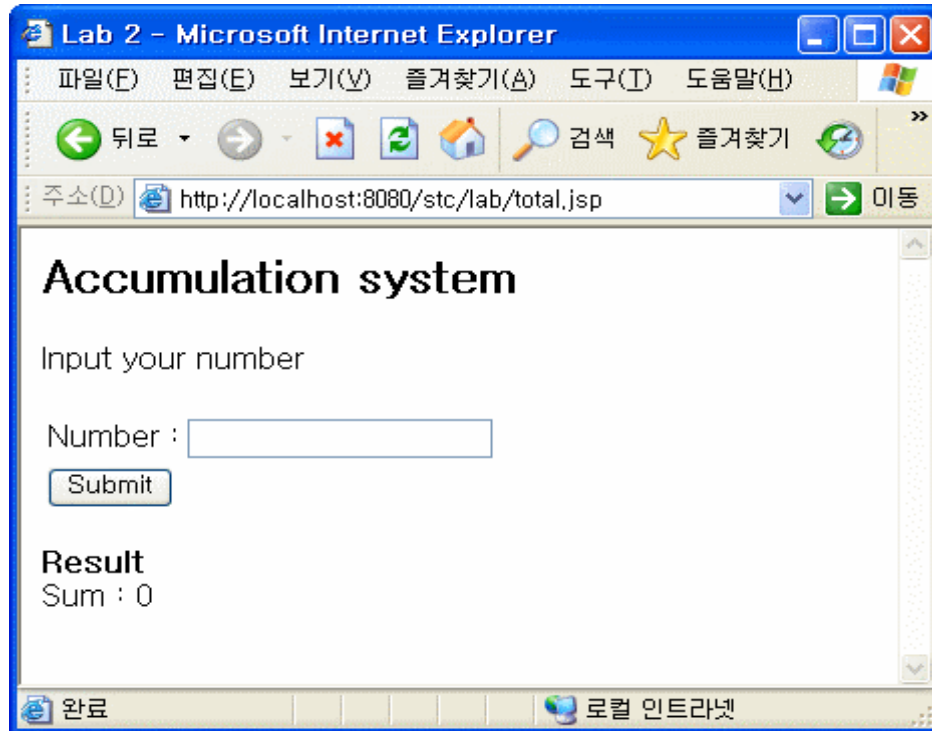
```
package util;
import javax.servlet.*;
import javax.servlet.http.*;
public class CookieUtil {
    public static String getCookieValue(Cookie[] cookies, String cookieName) {
        String cookieValue = "";
        Cookie cookie;
        for (int i=0; i<cookies.length; i++) {
            cookie = cookies[i];
            if (cookieName.equals(cookie.getName())) {
                cookieValue = cookie.getValue();
            }
        }
        return cookieValue;
    }
}
```

#### ❑ CookieUtil 클래스를 사용하는 코드

```
Cookie[] cookies = request.getCookies();
String emailAddress = CookieUtil.getCookieValue( cookies, "emailCookie" );
```

# Lab #1

## ❑ 합산 JSP 페이지 만들기



## 문제 기술

1. 숫자를 입력 받는다.
2. 'Submit' 버튼을 누르면 숫자를 합산한다.
3. 아무것도 입력하지 않고 'Submit'을 누르면 결과 값에는 변화가 없어야 한다. (No error)
4. 0을 입력하면 Sum을 reset한다.
5. 문자열을 입력 받는 경우에도 에러가 발생하지 않는다