

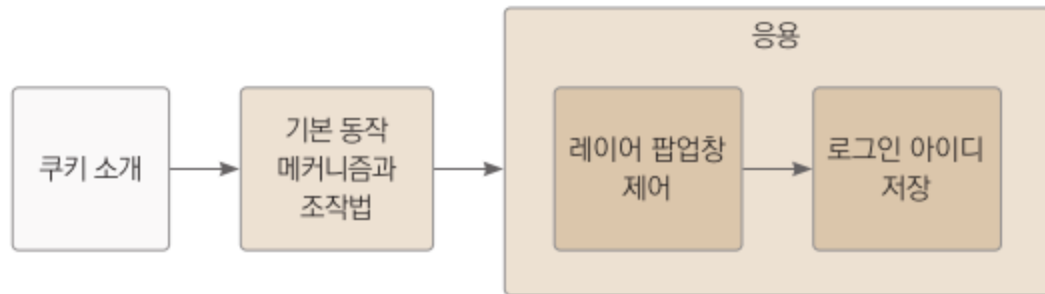
04

쿠키(Cookie)

■ 학습 목표

- 클라이언트의 상태 정보를 클라이언트의 PC에 저장할 수 있는 쿠키에 대해 학습합니다.

■ 학습 순서



■ 활용 사례

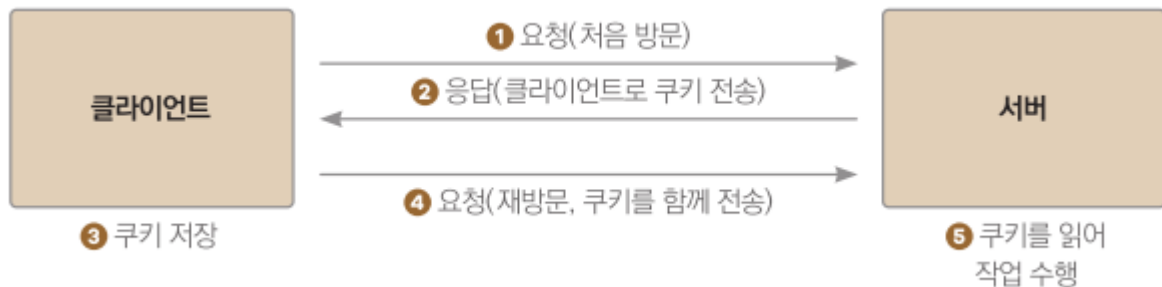
- 쿠키는 클라이언트 PC에 저장되어 웹 사이트가 방문자의 상태를 기억하는 수단임.
- 방문시점, 클릭한 페이지, 구매한 상품 등 여러가지 정보를 저장할 수 있음

■ 쿠키란..??

- 클라이언트의 상태 정보를 유지하기 위한 기술
- 주로 클라이언트 측의 웹브라우저에 key 와 value 형태로 저장
- 웹 서버는 브라우저가 전송한 쿠키로부터 필요한 데이터를 읽을 수 있음
- 쿠키 표준 RFC 6265에 명시된 제약
 - 3000개까지 만들 수 있다.
 - 쿠키 하나의 최대 크기는 4096바이트이다.
 - 하나의 호스트나 도메인에서 최대 50개까지 만들 수 있다.
- 따라서 쿠키는 대략 1.2Mb정도까지 저장할 수 있음

■ 쿠키의 동작 매커니즘

1. 클라이언트가 서버에 요청(첫 방문)
2. 서버가 쿠키를 생성하여 HTTP 응답 헤더에 실어 클라이언트에 전송
3. 클라이언트는 쿠키를 받아 웹브라우저에 저장
 - 이 시점에는 쿠키가 클라이언트에만 저장된 상태이므로 서버는 아직 쿠키를 사용할 수 없음
4. 클라이언트의 두번째 요청 시 저장해둔 쿠키를 HTTP 요청 헤더에 실어 서버로 전송
5. 서버는 쿠키의 정보를 읽어 필요한 작업을 수행할 수 있음



■ 속성과 API

구분	set계열 메서드	get계열 메서드
이름(쿠키명)	없음	String getName()
값(쿠키값)	void setValue(String value)	String getValue()
경로	void setPath(String path)	String getPath()
유지시간(초단위)	void setMaxAge(int expire_seconds)	int getMaxAge()
도메인	void setDomain(String domain)	String getDomain()

- set 계열 메서드는 생성, get 계열 메서드는 값을 반환
- 쿠키의 이름은 생성자를 통해서만 설정 가능
- 또한 한번 생성된 쿠키는 이름을 변경할 수 없음

형식	<code>new Cookie(String name, String value)</code>
-----------	--



4.2 기본 동작 확인

■ 기본 조작법

예제 4-1] 04Cookie/CookieMain.jsp

<h2>1. 쿠키(Cookie) 설정</h2>

<%

Cookie cookie = new Cookie("myCookie", "쿠키맛나요"); // 쿠키 생성

1 cookie.setPath(request.getContextPath()); // 경로를 컨텍스트 루트로 설정 2

cookie.setMaxAge(3600); // 유지 기간을 1시간으로 설정

response.addCookie(cookie); // 응답 헤더에 쿠키 추가

%>

<h2>2. 쿠키 설정 직후 쿠키값 확인하기</h2>

<%

Cookie[] cookies = request.getCookies(); // 요청 헤더의 모든 쿠키 얻기 4

3 if (cookies!=null) {

for (Cookie c : cookies) { // 쿠키 각각의.. 5

String cookieName = c.getName(); // 쿠키 이름 얻기 6

String cookieValue = c.getValue(); // 쿠키 값 얻기 7

// 화면에 출력

out.println(String.format("%s : %s
", cookieName, cookieValue)); 8

}

}

%>

1 쿠키 생성을 위한 코드

2 경로, 유지시간을 설정한 후 응답 헤더를 통해 클라이언트로 전송

3 생성된 쿠키를 읽기위한 코드

4 기존에 생성된 모든 쿠키 얻기

5 갯수만큼 반복하여 쿠키명과 값을 읽어 출력

[Note] 최초 실행시에는 myCookie가 출력되지 않음



4.2 기본 동작 확인

■ 기본 조작법

예제 4-2] 04Cookie/CookieResult.jsp

```
<body>
  <h2>쿠키값 확인하기(쿠키가 생성된 이후의 페이지)</h2>
  <%
    Cookie[] cookies = request.getCookies();
    if (cookies != null) {
      for (int i = 0; i < cookies.length; i++) {
        String cookieName = cookies[i].getName();
        String cookieValue = cookies[i].getValue();
        out.println(String.format("쿠키명 : %s - 쿠키값 : %s<br/>",
                                   cookieName, cookieValue));
      }
    }
  %>
</body>
```

① 쿠키를 읽어오기 위한 코드는 예제4-1과 동일함.

[Note] 페이지 이동을 하거나 새로 고침 해야 myCookie가 화면에 출력됨

쿠키값 확인하기(쿠키가 생성된 이후의 페이지)

쿠키명 : JSESSIONID - 쿠키값 : A47F68902D1037537C4AD26F49378AA0
쿠키명 : myCookie - 쿠키값 : 쿠키맞나요

■ 레이어 팝업창 제어

- CSS의 position속성을 이용한 레이어 팝업창에 “오늘 하루 열지 않음” 기능 구현



- 첫 진입시에는 팝업창 보임
- “체크” 없이 닫았을때는 새로고침 했을때 다시 보임
- “체크” 후 닫으면 새로고침 해도 하루동안은 보이지 않음

■ 쿠키 없이 기본 기능 구현하기

예제 4-3] 04Cookie/PopupMain_0.1.jsp

```
<style> ② CSS로 레이어 팝업창의 위치 지정
  div#popup {
    position: absolute; top:100px; left:50px; color:yellow;
    width:270px; height:100px; background-color:gray;
  }
  div#popup>div {
    position: relative; background-color:#ffffff; top:0px;
    border:1px solid gray; padding:10px; color:black;
  }
</style>
```

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.0/jquery.min.
js"></script>
<script>
$(function() {
  $('#closeBtn').click(function() { ④ 닫기 버튼(id="closeBtn")을 누르면
    $('#popup').hide(); ⑤ 팝업창(id="popup")을 숨김 처리합니다.
  });
});
</script>
```

② position 속성을 absolute로 부여하면 웹 브라우저의 좌측상단을 기준으로 레이어를 배치할 수 있음

③ jQuery의 hide() 함수를 이용해서 팝업창 닫기를 구현. 해당 함수는 display의 속성에 none을 부여해서 엘리먼트를 숨김 처리.

■ 쿠키 없이 기본 기능 구현하기

예제 4-3] 04Cookie/PopupMain_0.1.jsp(계속)

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
String popupMode = "on"; ❶ 레이어 팝업창 띄울지 여부
%>
<!DOCTYPE html>
```

```
<h2>팝업 메인 페이지(ver 0.1)</h2>
<%
    for (int i = 1 ; i <= 10 ; i++) {
        out.print("현재 팝업창은 " + popupMode + " 상태입니다.<br/>");
    }

    if (popupMode.equals("on")) { ——— ❸ popupMode 값이 "on"일 때만 팝업창 표시
%>
<div id="popup"> ❷ 공지사항 팝업 화면
    <h2 align="center">공지사항 팝업입니다.</h2>
    <div align="right"><form name="popFrm">
        <input type="checkbox" id="inactiveToday" value="1" /> ❸ 체크박스
        하루 동안 열지 않음
```

❶ 팝업창의 기본 상태는 “보임” 으로 설정. 차후 쿠키를 읽어와서 최종 상태를 결정할 것임

❸ ❶ 에서 “on”으로 설정했으므로 팝업창은 웹 브라우저에 표시됨.
이와 같이 스크립트릿 내부의 if문을 통해 HTML태그의 출력여부를 동적으로 제어할 수 있다.

▣ 쿠키를 이용해 상태 정보 유지하기

예제 4-4] 04Cookie/PopupMain.jsp(4-3에서 추가)

```
<%  
String popupMode = "on";  
  
Cookie[] cookies = request.getCookies(); ——— ❶ 쿠키를 읽어 popupMode 값 설정  
if (cookies != null) {  
    for (Cookie c : cookies) {  
        String cookieName = c.getName();  
        String cookieValue = c.getValue();  
        if (cookieName.equals("PopupClose")) { ❷ "PopupClose" 쿠키가 존재하면  
            popupMode = cookieValue; ❸ popupMode의 값 갱신  
        }  
    }  
}  
}%>
```

- ❶ 웹 브라우저에 생성된 모든 쿠키를 배열로 얻어옴
- ❷ 쿠키명이 "PopupClose"가 있다면
- ❸ popupMode의 값을 변경

▣ 쿠키를 이용해 상태 정보 유지하기

예제 4-4] 04Cookie/PopupMain.jsp(계속)

```
$(function() {
    $('#closeBtn').click(function() { ④ 닫기 버튼을 누르면 실행되는 함수
        $('#popup').hide();
        var chkVal = $("input:checkbox[id=inactiveToday]:checked").val();
        if(chkVal==1){
            ⑤ 체크 여부
            $.ajax({ ⑥ jQuery의 ajax() 함수로 비동기 요청
                url : './PopupCookie.jsp',
                type : 'get',
                data : {inactiveToday : chkVal},
                dataType : "text",
                success : function(resData) { ⑦ 요청 성공 시 호출되는 함수
                    if (resData != '') location.reload();
                }
            });
        }
    });
});
</script>
```

- ④ 팝업창의 닫기 버튼을 클릭했을때 실행
- ⑤ “오늘 하루 열지않음”에 체크했을때 값을 얻어옴
- ⑥ \$.ajax() 함수를 통해 비동기 방식으로 체크한 값을 전송. 체크하지 않았다면 빈값 전송됨

▣ 쿠키를 이용해 상태 정보 유지하기

예제 4-5] 04Cookie/PopupCookie.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8" trimDirectiveWhitespaces="true"%>

<%
    String chkVal = request.getParameter("inactiveToday");

    if (chkVal != null && chkVal.equals("1")) {
        Cookie cookie = new Cookie("PopupClose", "off"); // 쿠키 생성
        cookie.setPath(request.getContextPath()); // 경로 설정
        cookie.setMaxAge(60*60*24); // 유지 기간 설정
        response.addCookie(cookie); // 응답 객체에 추가
        out.println("쿠키 : 하루 동안 열지 않음");
    }
%>
```

① "inactiveToday" 매개변수의 값 얻기

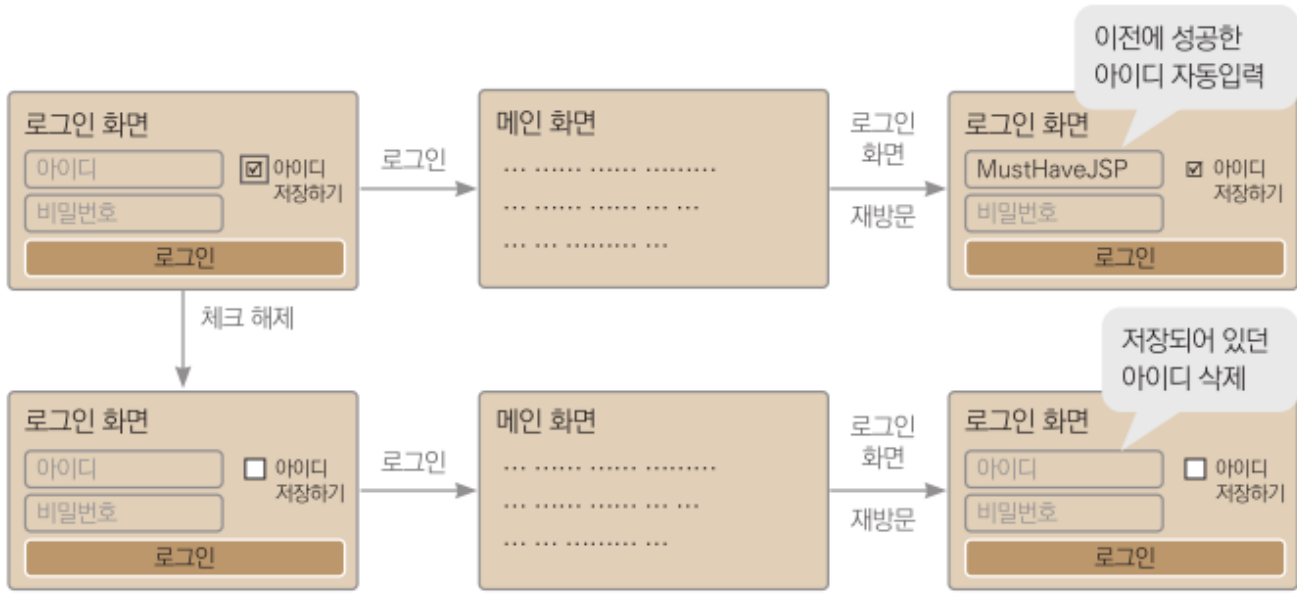
② 값이 "1"이면 쿠키를 생성해 응답 객체에 추가

③ 콜백 데이터 출력

- ① “오늘 하루 열지않음”에 체크한 경우 전송된 값
- ② ①의 값이 1이라면 쿠키값을 “off”로 하루동안 유지되는 쿠키를 생성
- ③ 콜백 데이터 출력

로그인 아이디 저장

- 로그인에 성공한 경우에만 쿠키를 생성 및 삭제
- 저장된 쿠키가 있다면 로그인 페이지에서는 아이디가 자동 입력
- [아이디 저장하기] 체크박스를 해제하고 로그인에 성공하면 쿠키 삭제





4.4 로그인 아이디 저장

▣ 편의 기능 구현하기 : 자바스크립트 코드 추가

- JSP에 Javascript코드를 삽입하기 위한 유틸리티 클래스 생성
- 소스가 전체적으로 지저분해지고 반복되는 코드가 많아지는 것을 방지

예제 4-7] Java Resources/utils/JSFunction.java

```
package utils;

import jakarta.servlet.jsp.JspWriter; // 필요한 클래스 импорт ❶

public class JSFunction {
    // 메시지 알림창을 띄운 후 명시한 URL로 이동합니다.
    public static void alertLocation(String msg, String url, JspWriter out) { ❷
        try {
            String script = "" // 삽입할 자바스크립트 코드
                + "<script>"
                + "    alert("'" + msg + "');"
                + "    location.href='" + url + "';"
                + "</script>"; ❸
            out.println(script); // 자바스크립트 코드를 out 내장 객체로 출력(삽입) ❹
        }
        catch (Exception e) {}
    }
}
```

❷ static 메서드로 정의. out 내장객체를 메서드 내에서 사용하기 위해 JspWriter 타입의 매개변수를 통해 전달받음.

❸ Javascript 코드를 String으로 정의

❹ 웹 브라우저에 스크립트 출력

▣ 편의 기능 구현하기 : 쿠키 관리자

- 쿠키 생성 : 쿠키 객체생성, 경로 및 유지시간 설정 필요
- 쿠키 읽기 : 배열로 가져온 후 배열명이 있는지 확인
- 위 2가지의 작업을 위해 코드의 반복이 많아지므로 별도의 유틸리티 클래스로 정의

예제 4-8] Java Resources/utls/CookieManager.java

```
public class CookieManager {  
    // 명시한 이름, 값, 유지 기간 조건으로 새로운 쿠키를 생성합니다.  
    public static void makeCookie(HttpServletResponse response, String cName,  
        String cValue, int cTime) {  
        Cookie cookie = new Cookie(cName, cValue); // 쿠키 생성 ❶  
        cookie.setPath("/"); // 경로 설정 ❷  
        cookie.setMaxAge(cTime); // 유지 기간 설정 ❸  
        response.addCookie(cookie); // 응답 객체에 추가 ❹  
    }  
}
```

매개변수를 통해 전달되는 쿠키명, 쿠키값, 유지시간으로 새로운 쿠키를 생성



4.4 로그인 아이디 저장

■ 편의 기능 구현하기 : 쿠키 관리자

예제 4-8] Java Resources/utls/CookieManager.java(계속)

```
// 명시한 이름의 쿠키를 찾아 그 값을 반환합니다.
public static String readCookie(HttpServletRequest request, String cName) {
    String cookieValue = ""; // 반환 값

    Cookie[] cookies = request.getCookies(); ⑤
    if (cookies != null) {
        for (Cookie c : cookies) {
            String cookieName = c.getName();
            if (cookieName.equals(cName)) { ⑥
                cookieValue = c.getValue(); // 반환 값 갱신
            }
        }
    }
    return cookieValue; ⑦
}

// 명시한 이름의 쿠키를 삭제합니다.
public static void deleteCookie(HttpServletResponse response, String cName) {
    makeCookie(response, cName, "", 0); ⑧
}
}
```

- ⑤ 쿠키를 배열로 얻어온 후
- ⑥ 쿠키명이 일치하면 쿠키값을 읽어옴
- ⑦ 쿠키값 반환
- ⑧ 쿠키는 유지시간이 0이면 삭제



4.4 로그인 아이디 저장

로그인 페이지 작성하기

예제 4-9] 04Cookie/IdSaveMain.jsp

```
<%  
String loginId = CookieManager.readCookie(request, "loginId"); ❶  
  
String cookieCheck = "";  
if (!loginId.equals("")) { ❸  
    cookieCheck = "checked"; ❹  
} ❷  
%>  
<html>
```

```
<h2>로그인 페이지</h2>  
<form action="IdSaveProcess.jsp" method="post">  
아이디 : <input type="text" name="user_id" value="<%= loginId %>" /> ❺  
        <input type="checkbox" name="save_check" value="Y" <%= cookieCheck %> /> ❻  
        아이디 저장하기  
<br />  
패스워드 : <input type="text" name="user_pw" />
```

- ❶ 앞서 작성한 유틸리티 클래스를 통해 “loginId”라는 쿠키를 읽어옴
- ❷ 쿠키에 저장된 아이디가 있다면 “checked” 속성값 저장
- ❸
- ❹
- ❺ 아이디 입력상자에 값 설정
- ❻ 체크박스에 “checked” 속성값 설정

로그인 및 아이디 저장 기능 구현하기

- 사용자 인증은 하드코딩된 문자열 비교로 수행

예제 4-10] 04Cookie/IdSaveProcess.jsp

```
if ("must".equals(user_id) && "1234".equals(user_pw)) { ❷ 사용자 인증
    // 로그인 성공
    if (save_check != null && save_check.equals("Y")) { ❸ [아이디 저장하기] 체크 확인
        CookieManager.makeCookie(response, "loginId", user_id, 86400); ❹ 쿠키 생성
    }
    else {
        CookieManager.deleteCookie(response, "loginId"); ❺ 쿠키 삭제
    }

    JSFunction.alertLocation("로그인에 성공했습니다.", "IdSaveMain.jsp", out); ❻
}
else {
    // 로그인 실패
    JSFunction.alertBack("로그인에 실패했습니다.", out); ❼
}
%>
```

- ❹ 로그인에 성공하고 “아이디 저장”에 체크한 경우에는 쿠키 생성
- ❺ 로그인 성공 및 체크를 해제한 경우에는 쿠키 삭제
- ❻ 성공을 알림창으로 띄운 후 페이지 이동
- ❼ 실패를 알림창으로 띄운 후 뒤로 이동

- 쿠키는 클라이언트의 상태 정보를 저장 하는 기술
- 쿠키를 이용해 팝업창 제어, 로그인 아이디 저장 등의 기능을 구현할 수 있음
- 쇼핑몰에서 오늘 본 상품을 기억하거나, 게시글 조회수를 하루에 1회만 증가시키는 등의 기능에도 응용할 수 있음

■ 핵심요약

- 쿠키는 생성자를 통해서만 생성할 수 있음
- 생성 후 쿠키값은 변경할 수 있으나 쿠키명은 변경할 수 없음
- `setPath()` 메서드로 적용할 경로를 설정
- `setMaxAge()` 메서드로 쿠키의 유지 기간을 설정
- `response` 내장 객체의 `addCookie()` 메서드로 클라이언트에 쿠키를 저장
- 쿠키는 생성 직후 바로 사용할 수는 없고, 클라이언트가 재요청을 했을 때부터 사용할 수 있음

