

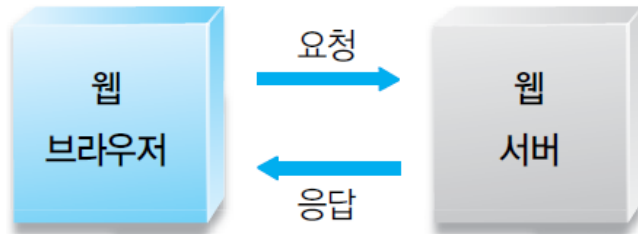
## 7. 쿠키와 세션

1. 쿠키
2. 세션

# 1. 쿠키(Cookie)

## □ 쿠키 개요

- HTTP 프로토콜은 웹 브라우저(클라이언트)의 요청에 대한 응답을 하고 나면 해당 클라이언트와의 연결을 지속하지 않음



클라이언트의 요청에 응답 후

◀ HTTP 프로토콜의 비연결성

# 1. 쿠키(Cookie)

## □ 쿠키 개요

- ▣ 상태가 없는 프로토콜을 위해 상태를 지속시키기 위한 방법
- ▣ 정보를 웹 브라우저에 저장

쿠키의 동작 방식 ▶



# 1. 쿠키(Cookie)

## □ 쿠키 사용 순서

- ① 먼저 쿠키를 생성한다.
- ② 쿠키에 필요한 설정을 한다. 쿠키의 유효 시간, 쿠키에 대한 설명 등을 적용하고, 도메인, 패스, 보안등을 한다.
- ③ 웹 브라우저에 생성된 쿠키를 전송한다.

## □ 웹 브라우저에 저장된 쿠키를 사용하는 절차

- ① 웹 브라우저의 요청에서 쿠키를 얻어온다.
- ② 쿠키는 이름, 값의 쌍으로 된 배열형태로 리턴 된다. 리턴 된 쿠키의 배열에서 쿠키 이름을 가져온다.
- ③ 쿠키 이름을 통해서 해당 쿠키에 설정된 값을 추출한다.

# 1. 쿠키(Cookie)

## □ 쿠키의 사용

- 이름, 값, 유효기간, 도메인, 경로 등으로 이루어짐
- 쿠키의 이름은 알파벳과 숫자로 이루어짐
  - 쿠키 값은 공백, 괄호, 등호, 콤마, 콜론, 세미콜론을 포함하려면 인코딩 필요
- 쿠키 생성
  - `Cookie cookie = new Cookie(String name, String value);`
- 쿠키를 `response`객체에 추가
  - `response.addCookie(cookie);`
- `request`객체에 실려 온 쿠키를 읽어 올 때
  - `Cookie[] cookies = request.getCookies();`

## □ 쿠키 생성-예

```
// 1. Cookie 객체 생성
Cookie c = new Cookie("id", "pinksung");
// 2. 유효기간 설정
c.setMaxAge(365 * 24 * 60 * 60);
// 3. 클라이언트에 쿠키 전송
response.addCookie(c);
//4. 쿠키를 생성하여 클라이언트에 전송
response.addCookie(new Cookie("pwd", "test1234"););
response.addCookie(new Cookie("age", "20"););
```

## □ 쿠키 얻기

- ▣ 쿠키 객체 얻어오기
- ▣ 쿠키 객체에 설정된 값 알아내기

```
Cookie[] cookies=request.getCookies();  
for(Cookie c:cookies){  
    out.println(c.getName() + " : "+c.getValue() + "<br>");  
}
```

## □ 쿠키 값 삭제

- ▣ 유효기만 만료 시킴
- ▣ `setMaxAge(0)` 으로 설정

```
Cookie cookie = new Cookie("id", "");  
cookie.setMaxAge(0); // 쿠키의 유효기간을 만료시킴  
response.addCookie(cookie);
```

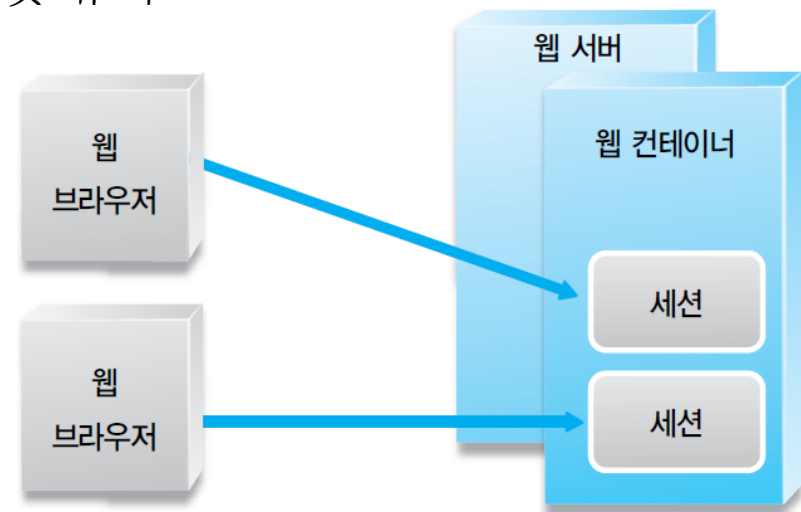


## 2. 세션(Session)

### □ 세션 개요

- 웹 서버 쪽의 웹 컨테이너에 상태를 유지하기 위한 정보를 저장
  - 웹 브라우저 당 1개씩 생성되어 웹 컨테이너에 저장
- 웹 브라우저와 웹 서버의 상태 유지가 훨씬 안정적이고 보안상의 문제도 해결
- 웹 서버는 각각의 웹 브라우저로부터 발생한 요청에 대해서 특정한 식별자를 부여
  - 이것을 사용해서 세션을 구분 및 유지

세션과 웹 브라우저의 관계 ▶



# 세션(Session)

## □ 세션 사용 방법

### ▣ 세션의 속성 설정

- session객체의 `setAttribute()` 메소드를 사용
  - `session.setAttribute(String name, Object value);`
  - `session.setAttribute( "id","aaaa@king.com");`

### ▣ 설정된 세션 정보 얻기

- session객체의 `getAttribute()` 메소드를 사용
  - `Object getAttribute(String name)`
  - `String id = (String)session.getAttribute("id");`

## □ 세션 사용 방법

### ▣ 모든 세션 정보 얻기

- Enumeration `getAttributeNames()`
- `hasMoreElements()` : Enumeration 에 데이터가 있는지 검사
- `nextElement()`: 다음 아이템이 존재 하다는 가정하에 아이템 하나씩 가져옴

```
Enumeration<String> names = session.getAttributeNames();  
while (names.hasMoreElements()) {  
    String name = names.nextElement().toString();  
    String value = session.getAttribute(name).toString();  
    out.println(name + " : " + value + "<br>");  
}
```

## □ 세션사용

### ▣ 세션관련 메소드

- getId() : 세션 아이디 반환
- getCreationTime() : 세션생성 시간 반환
- getLastAccessedTime() : 마지막 접속 시간 반환, 마지막으로 request 를 보내 시간을 long형으로 반환
- isNew() : 해당 세션에 처음 생성이면 true, 이전 생성이면 false
- getMaxInactiveInterval() 세션 유효 시간 반환

```
String id_str = session.getId();  
long lasttime = session.getLastAccessedTime();  
long createdtime = session.getCreationTime();  
long time_used = (lasttime - createdtime) / 60000;  
int inactive = session.getMaxInactiveInterval() / 60;  
boolean b_new = session.isNew();
```

## □ 세션 사용방법

### ▣ 세션의 속성 삭제

- session객체의 removeAttribute() 메소드를 사용

```
session.removeAttribute("name");  
session.removeAttribute("id");
```

### ▣ 세션 무효화 - 모든 세션 속성 제거

- session객체의 invalidate() 메소드를 사용

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

## 2. 세션(Session)

### □ 세션을 사용한 로그인 처리

