

# 상태정보 유지기술

쿠키, 세션

# 쿠키(Cookie)의 특성

정보저장 위치 : 사용자 컴퓨터

서버의 부하가 분산된다.

보안상의 문제가 발생할 수 있다.

정보저장 유효기간을 사용자가 설정한다. (브라우저가 닫힌 이후에도 유지)

한 사이트에서 저장할 수 있는 쿠키 객체는 한정되어 있음(20개)

하나의 쿠키에 저장할 수 있는 데이터는 브라우저별 한계가 있음(4k)

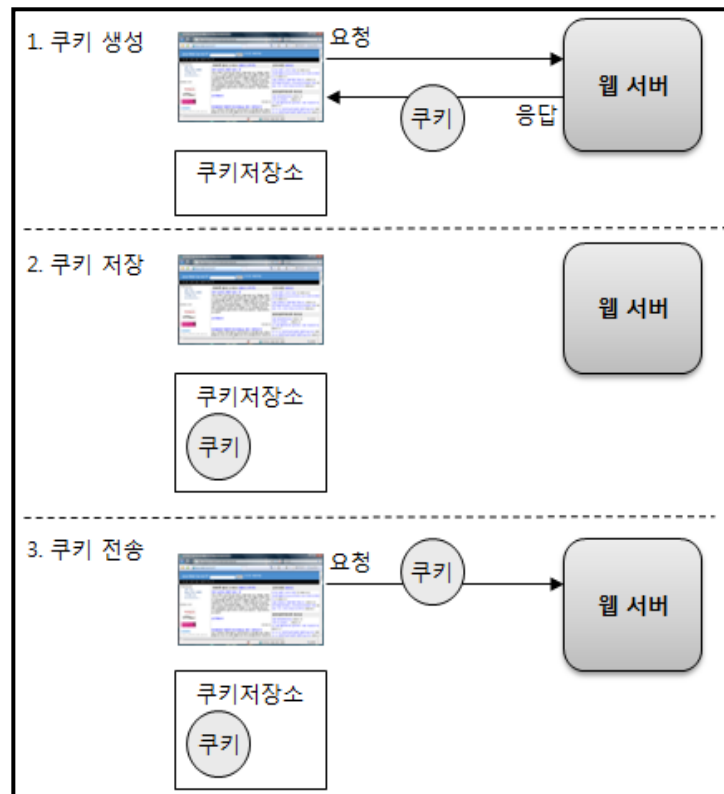
컴퓨터에 쿠키 한계치(300개)가 오게 되면 랜덤하게 저장 되어 있는 쿠키가 삭제 된다.

쿠키의 정보저장은 문자열만 가능

한글 저장 시에는 인코딩을 반드시 해야 가능

# 쿠키(Cookie)

- '쿠키(cookie)'는 웹 브라우저가 보관하고 있는 데이터로서 웹 서버에 요청을 보낼 때 함께 전송
- 동작방식



- 쿠키 생성 및 전송하기

```
Cookie cookie = new Cookie("cookieName", "cookieValue");  
response.addCookie(cookie);
```

- 쿠키 객체 얻어오기

```
Cookie[] cookies = request.getCookies();
```

- Cookie 객체 중요 API

메서드	리턴타입	설명
getName()	String	쿠키의 이름을 추출한다.
getValue()	String	쿠키의 값을 추출한다.

# 쿠키의 구성

## • 구성요소

명칭	설명
이름(name)	각각의 쿠키를 구별하는 데 사용.
값(value)	쿠키의 이름과 관련된 값.
유효시간(maxAge)	쿠키의 유지 시간
도메인(domain)	쿠키를 전송할 도메인
경로(path)	쿠키를 전송할 요청 경로

## • 쿠키 이름 제약

- 아스키 코드의 알파벳과 숫자만을 포함할 수 있다.
- 콤마(,), 세미콜론(;), 공백(' ') 등의 문자는 포함 할 수 없다.
- '\$'로 시작할 수 없다.

# 쿠키의 인코딩 / 디코딩

- 쿠키는 값으로 한글과 같은 문자를 가질 수 없음
- 쿠키 값의 처리

설정하고자 하는 값의 인코딩 :

```
- URLEncoder.encode("값", "euc-kr")
- String cName = "한글이름";
  cName = URLEncoder.encode(cName, "euc-kr");
Cookie c = new Cookie("name", cName);
```

인코딩 되어있는 값의 디코딩 :

```
- URLDecoder.decode("값", "euc-kr")
- Cookie cookie = ....;
  String gName = cookie.getValue();
  gName = URLDecoder.decode(gName, "euc-kr");
```



# 쿠키 유효시간

- 쿠키의 유효 시간 설정

- **Cookie.setMaxAge(int maxAge)** : 초 단위 설정
- 유효 시간 미 지정 시, 웹 브라우저 닫을 때 쿠키도 함께 삭제
  - `setMaxAge(0);` // 쿠키 삭제
  - `setMaxAge(-1);` // 브라우저 닫을 때 쿠키 삭제
  - `setMaxAge(60 * 60 * 24);` // 1일 설정



# 세션(Session)의 특성

정보저장 위치 : 서버

서버의 부하가 가중된다.

보안상의 문제 발생 소지를 줄인다.

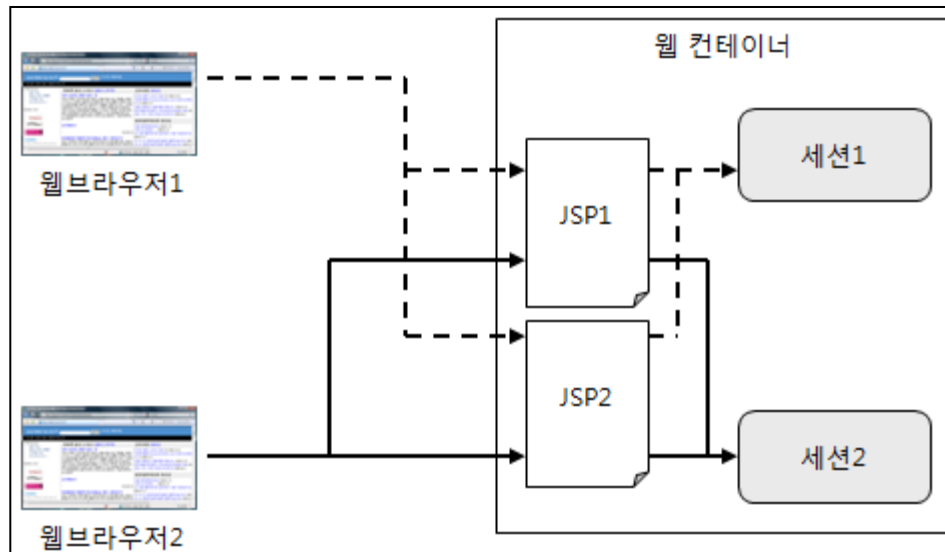
정보저장 최대 유효기간이 브라우저가 닫힐 때 까지 이다.

정보 저장 시 객체타입이 가능하다.



# 세션(session)

- 웹 컨테이너에서 클라이언트의 정보를 보관할 때 사용
- 오직 서버에서만 생성
- 클라이언트마다 세션이 생성



# 세션(Session)

- 세션이 존재하지 않을 경우 세션이 생성되고, 세션이 존재할 경우 이미 생성된 세션을 사용

## JSP

- `<% session.setAttribute("userInfo", userInfo); %>`

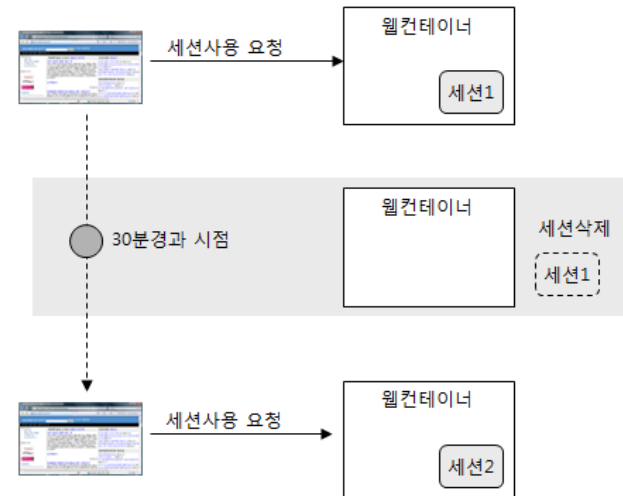
## Servlet

- `HttpSession session = request.getSession( );`  
: 세션이 없을 경우 새롭게 생성, 존재할 경우 기존 세션을 반환
- `HttpSession session = request.getSession( true/false);`  
: **false**로 설정할 경우에는 생성된 세션이 없을 경우 **null**을 반환  
하고 생성된 세션이 존재할 경우 기존 세션을 반환

# 세션(Session) 종료 / 타임 아웃

- 세션 타임아웃 시간 설정
  - web.xml에 설정(분 단위)

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



- 프로그램에서 설정(API 활용 - 초단위)  
`session.setMaxInactiveInterval(int interval);`

# 세션(Session) 주요 API 메소드

- 세션에 값 설정

- session.setAttribute(String name, Object value);
- 값 저장 시 유지보수를 위하여 단순 문자열보다는 클래스 객체를 선호

- 문자열 이용 시

```
session.setAttribute("id", "sbc");  
session.setAttribute("type", "S");
```

- 객체 이용 시

```
UserInfo user = new UserInfo();  
user.setId("sbc");  
user.setType("S");  
session.setAttribute("userInfo", user);
```

# 세션(Session) 주요 API 메소드

- 세션에 설정된 값 추출

- Object obj = session.getAttribute(String name);

```
String id = (String )session.getAttribute("id");  
UserInfo user = (UserInfo)session.getAttribute("userInfo");
```

설정된 객체 정보

```
session.setAttribute("id", "sbc");  
UserInfo user = new UserInfo();  
user.setId("sbc");  
user.setType("S");  
session.setAttribute("userInfo", user);
```



# 세션(Session) 주요 API 메소드

- 세션 속성 삭제

- session.removeAttribute(String name);

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

- 세션 삭제

- session.invalidate( );