JSP의 내장객체

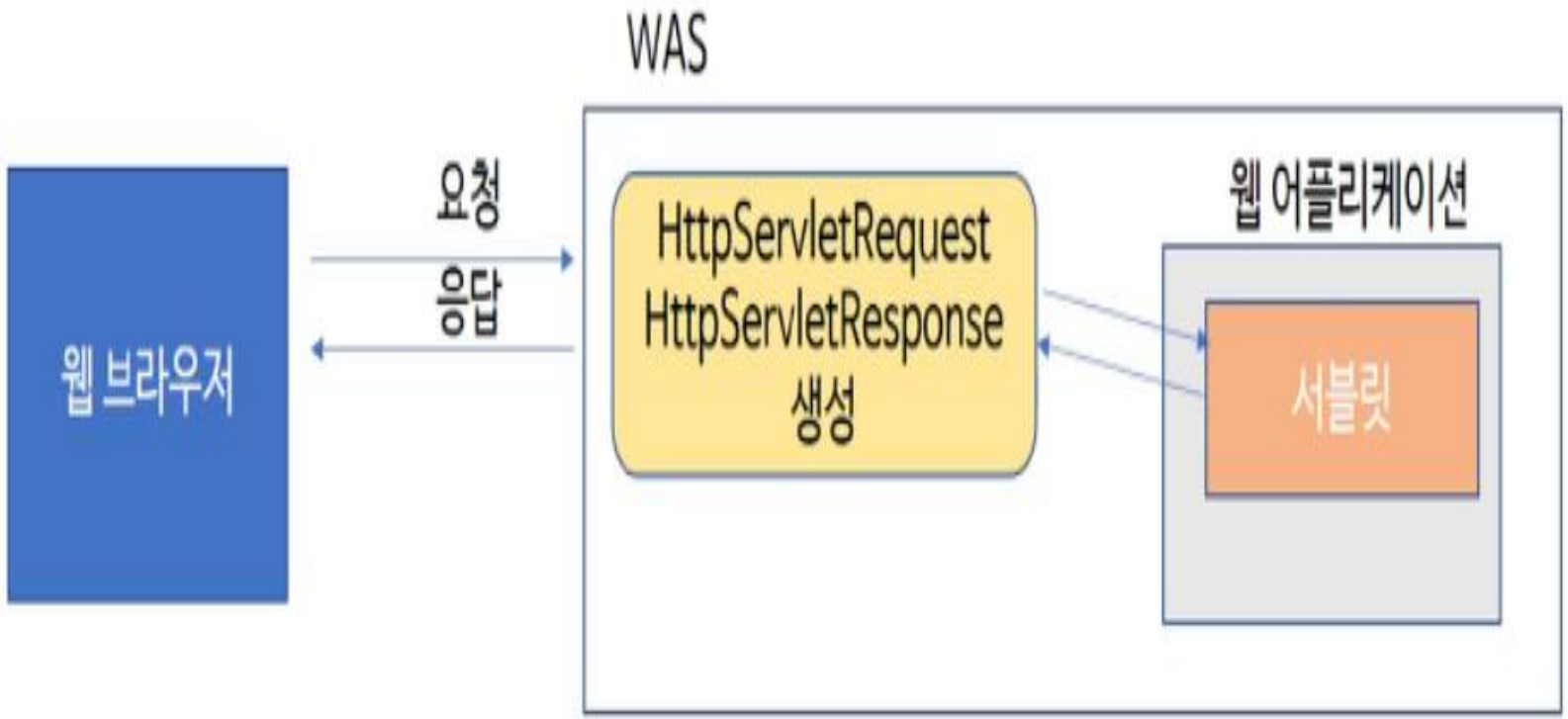
안 화 수

JSP의 내장객체

기본 객체	실제 타입	설명
request	javax.servlet.http.HttpServletRequest 또는 javax.servlet.ServletRequest	**클라이언트의 요청 정보를 저장한다.**
response	javax.servlet.http.HttpServletResponse 또는 javax.servlet.ServletResponse	**응답 정보를 저장한다.**
pageContext	javax.servlet.jsp.PageContext	JSP 페이지에 대한 정보를 저장한다.
session	javax.servlet.http.HttpSession	**HTTP 세션 정보를 저장한다.**
application	javax.servlet.ServletContext	웹 어플리케이션에 대한 정보를 저장한다.
out	javax.servlet.jsp.JspWriter	**JSP 페이지가 생성하는 결과를 출력할 때 사용되는 출력 스트림이다.**
config	javax.servlet.ServletConfig	JSP 페이지에 대한 설정 정보를 저장한다.
page	java.lang.Object	JSP 페이지를 구현한 자바 클래스 인스턴스이다.
exception	java.lang.Throwable	예외 객체. 에러 페이지에서만 사용된다.

request 객체

- **request 객체**
  클라이언트의 요청 정보를 처리해주는 객체



request 객체

request 객체의 주요 메소드

void setCharacterEncoding(String env) : 한글 인코딩 처리

String getParameter(String name) : name에 해당하는 파라미터 값을 구함

String[] getParameterValues(String name) : checkbox 같이 여러 개의 파라미터 값을 구함

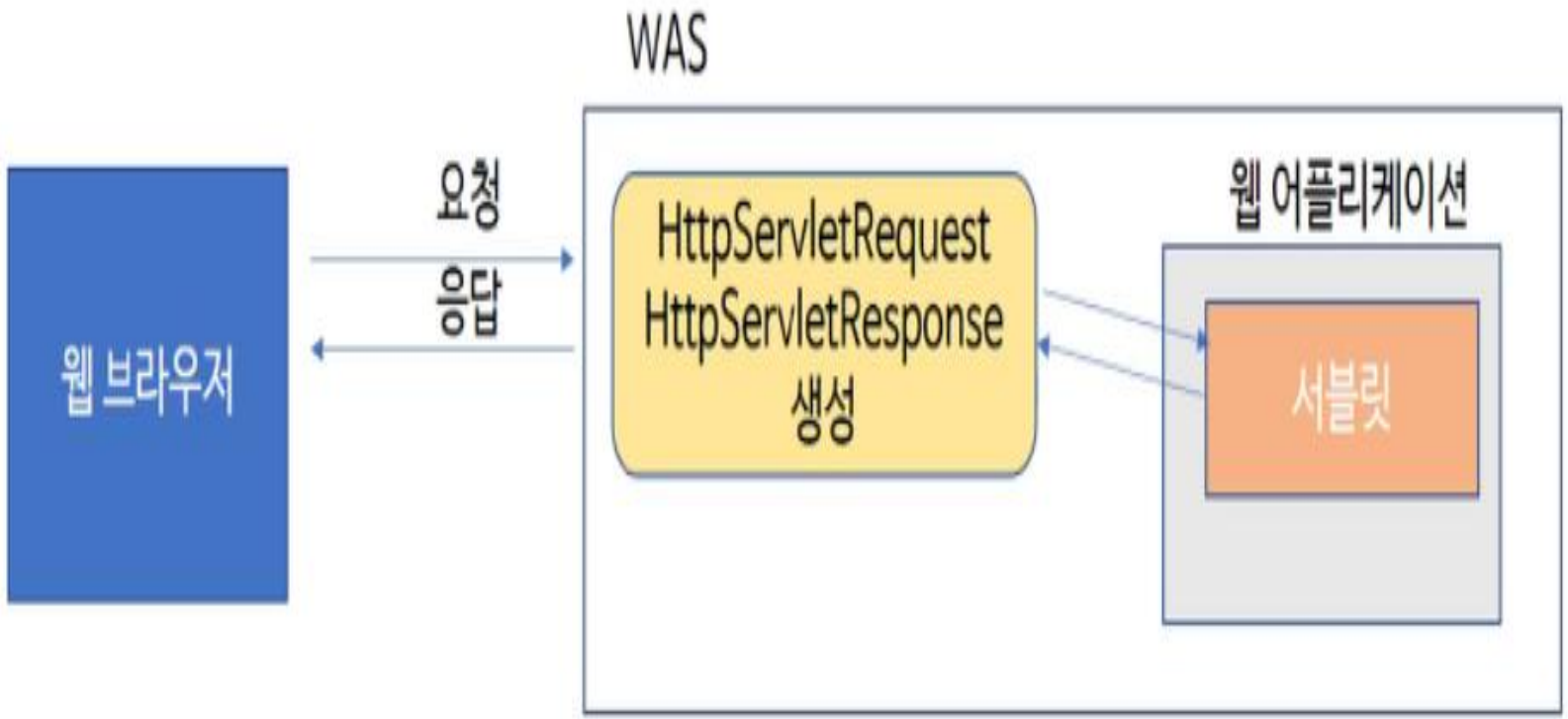
String getRemoteAddr() : client의 IP주소를 구함

String getRequestURI() : 요청 URI를 구함

String getContextPath() : 컨텍스트 패스(project명)를 구함

response 객체

- **response 객체**
  클라이언트의 응답 정보를 처리해주는 객체



response 객체

response 객체의 주요 메소드

String getCharacterEncoding() : 한글 인코딩 형태를 구함

void addCookie(Cookie cookie) : 쿠키를 발행함

Void sendRedirect(String location) : 지정한 URL로 이동한다.

response 객체

각 언어별 페이지 이동 방법

- HTML

```
<meta  http-equiv="refresh"  content="3;  url=http://www.naver.com">
```

- JavaScript

```
<script>
   alert("페이지 이동");
   location.href = "http://www.naver.com";
</script>
```

- JSP

```
<%
        response.sendRedirect("http://www.naver.com");
%>
```

out 객체

- **out 객체**
  JSP가 생성하는 데이터를 출력하기 위한 출력 스트림을 만들어 주는 객체

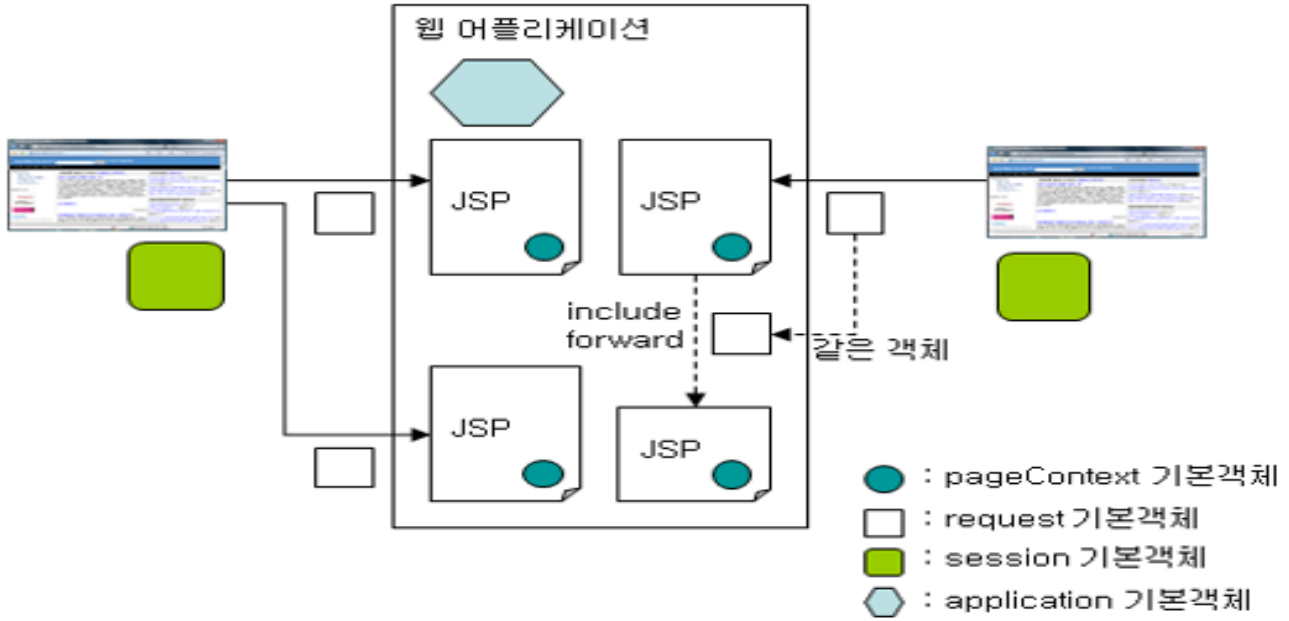
- **out객체의 주요 메소드**

  print() : 출력 스트림을 브라우저에 출력

  println() : 출력 스트림을 브라우저에 출력

기본객체와 영역

- 네 영역
  - PAGE 영역 - 하나의 JSP 페이지를 처리할 때 사용되는 영역
  - REQUEST 영역 - 하나의 HTTP 요청을 처리할 때 사용되는 영역
  - SESSION 영역 - 하나의 웹 브라우저와 관련된 영역
  - APPLICATION 영역 - 하나의 웹 어플리케이션과 관련된 영역



기본객체와 영역

기본 객체	영역	쓰임새
pageContext	PAGE	(한번의 요청을 처리하는) 하나의 JSP 페이지 내에서 공유될 값을 저장한다.
request	REQUEST	한번의 요청을 처리하는 데 사용되는 모든 JSP 페이지에서 공유될 값을 저장한다.
session	SESSION	한 사용자와 관련된 정보를 JSP 들이 공유하기 위해서 사용된다.
application	APPLICATION	모든 사용자와 관련해서 공유할 정보를 저장한다.

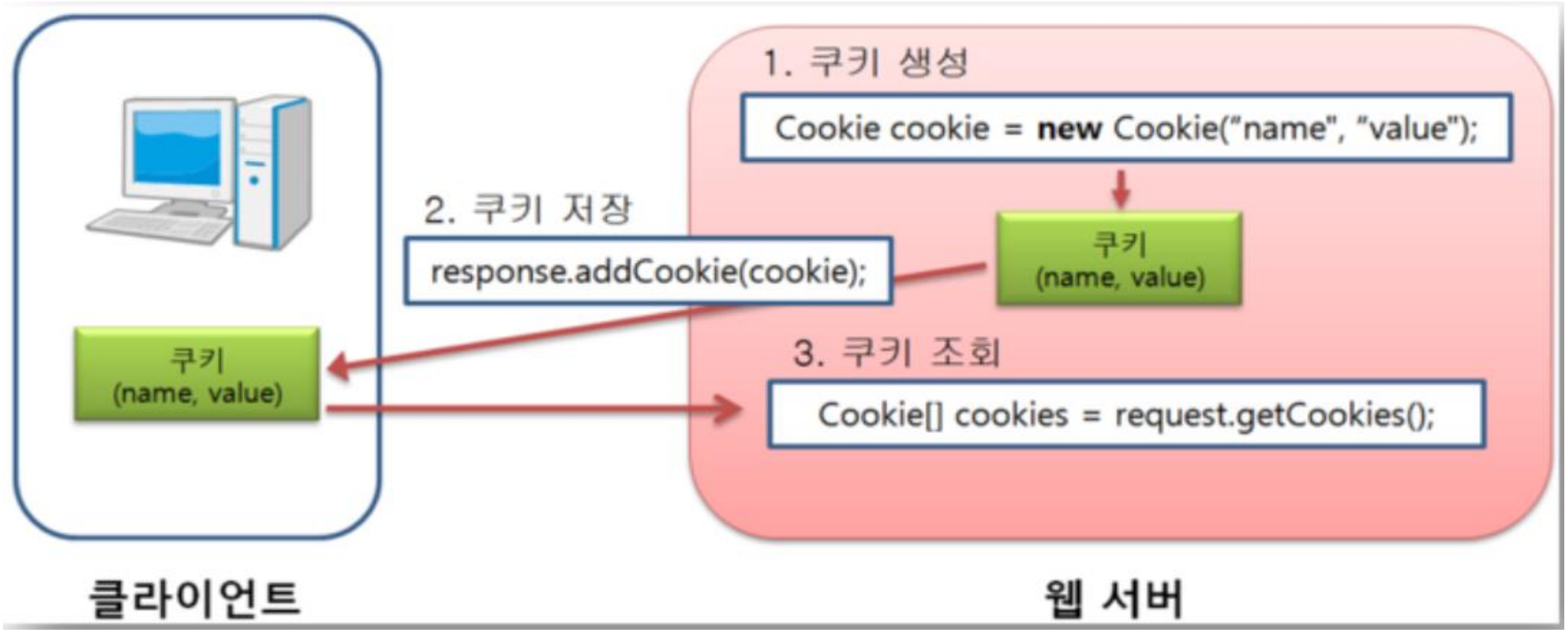
기본객체와 영역

- 영역에 관련된 기본 객체
  - pageContext, request, session, application
- 메서드

메서드	리턴타입	설명
setAttribute(String name, Object value)	void	이름이 name인 속성의 값을 value로 지정한다.
getAttribute(String name)	Object	이름이 name인 속성의 값을 구한다. 지정한 이름의 속성이 존재하지 않을 경우 null을 리턴한다.
removeAttribute(String name)	void	이름이 name인 속성을 삭제한다.
getAttributeNames()	Enumeration	속성의 이름 목록을 구한다. (pageContext 기본 객체는 이 메서드를 제공하지 않는다.)

cookie

❖ 쿠키 처리 과정



cookie

❖ 쿠키의 동작 순서

1. 클라이언트가 서버의 웹페이지를 요청한다.
2. 웹 서버에서 쿠키를 생성한다. (보통은 로그인에 성공 했을때 서버에서 쿠키를 발행한다.)
3. 생성한 쿠키에 정보를 담아 HTTP 화면을 돌려줄 때, 쿠키정보를 클라이언트에게 돌려준다.
4. 발행된 쿠키는 클라이언트의 로컬 PC에 저장 가지고 있다가, 서버에서 쿠키정보를 요청할 때 발행된 쿠키정보를 전송한다.
5. 동일 사이트 재방문시 클라이언트의 PC에 해당 쿠키가 있는 경우, 요청 페이지와 함께 쿠키를 전송한다.

cookie

❖ 쿠키의 기능

- cookie는 서버와 클라이언트가 연결 상태(로그인)를 유지하기 위해서 사용되는 개념이다.
- 보통 로그인에 성공 했을 경우에 서버측에서 쿠키를 발행하고, 발행된 쿠키는 클라이언트의 임시 폴더에 쿠키 파일이 생성된다.
- 서버측에서는 페이지를 이동하면서 클라이언트의 임시 폴더에 발행된 쿠키 정보를 참조하면서 서버와 클라이언트의 연결 상태(로그인)를 유지한다.
- cookie를 사용하면, 페이지 이동을 하면서 회원 정보를 가지고 가지 않아도 된다.

cookie

❖ 쿠키의 특징

1. 이름, 값, 만료일(저장 기간 설정), 경로 정보로 구성되어 있다.

2. 클라이언트에 총 300개의 쿠키를 저장할 수 있다.

3. 하나의 도메인 당 20개의 쿠키를 가질 수 있다.

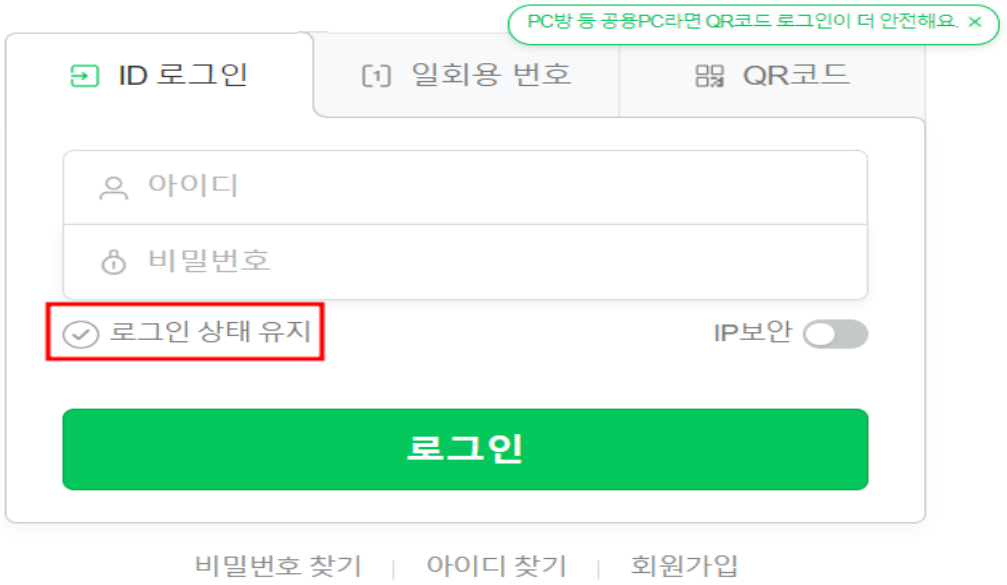
4. 하나의 쿠키는 4KB(=4096byte)까지 저장 가능하다.

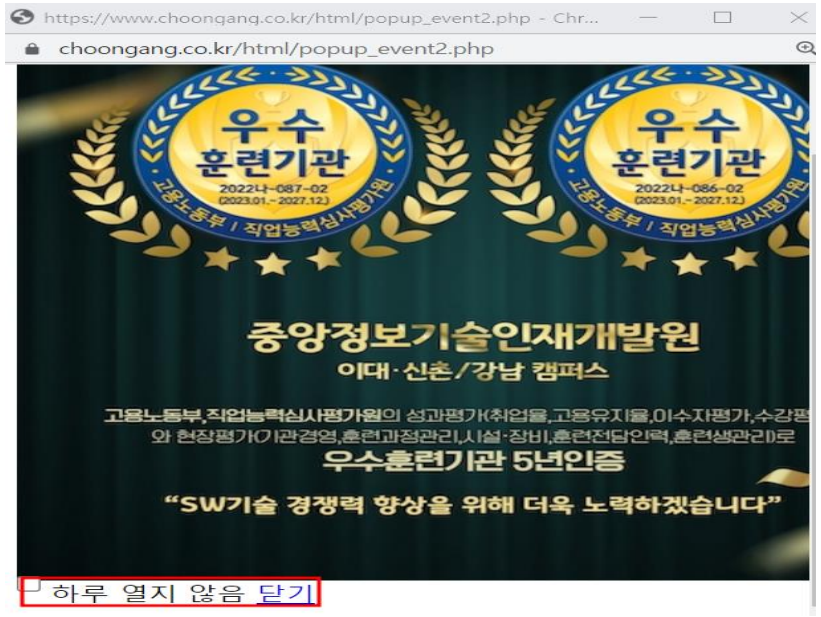
cookie

❖ 쿠키를 사용하는 경우

1. 로그인 할때 로그인상태 유지나 자동 로그인을 처리할때 쿠키를 사용한다.

2. 팝업창을 통해 "오늘 이 창을 다시 보지 않기" 체크할때 쿠키를 사용한다.





cookie

1. 쿠키 생성

```
Cookie cook =  new Cookie ( String name, String value );
Cookie cook =  new Cookie (  "id", "test" );
```

2. 쿠키 추가

쿠키를 생성한 후에는 반드시 response 객체의 addCookie() 메소드를 사용해서 쿠키를 추가해 주어야 한다.

```
response.addCookie( cook );
```

3. 쿠키 값 변경

쿠키 생성후 쿠키의 새로운 이름에 대응하는 값을 새롭게 지정할 때 setValue() 메소드를 사용한다.

```
cook.setValue( newValue );
```

cookie

4. 쿠키의 수명(지속지간) 설정

```
cook.setMaxAge ( int expiry );      // 단위: 초
```

만약, 쿠키의 유효시간을 1시간으로 설정 한다면....

```
cook.setMaxAge ( 60*60 );
```

5. 쿠키 읽기

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

쿠키의 이름 : cook.getName()
쿠키의 값 : cook.getValue()

session

session

- session은 서버와 클라이언트가 연결 상태(로그인)를 유지하기 위해서 사용되는 개념이다.
- 보통 로그인에 성공 했을 경우에 서버측에서 session으로 클라이언트와 연결된다.
  이때 session을 연결한 클라이언트의 웹브라우저를 통해서 회원정보를 공유하게 된다.
- session은 cookie와 다르게, 클라이언트의 컴퓨터에 회원정보를 저장하지 않는다.
- session을 사용하면, 페이지 이동을 하면서 회원 정보를 가지고 가지 않아도 된다.

session

1. 세션 설정

name으로 지정한 이름에 value값을 할당한다.

session.setAttribute (java.lang.String name, java.lang.Object value);

예) session.setAttribute (“id“, “guardian23“);

2. 세션 사용

name이란 이름에 해당되는 속성값을 Object타입으로 반환한다.

session.getAttribute (java.lang.String name)

예) String id = (String) session.getAttribute (“id“);

session

3. 세션의 유지시간 설정

세션의 최대 유지시간을 초 단위로 설정한다.

```
session.setMaxInactiveInterval();
```

4. 세션 삭제

name으로 지정한 속성 값을 제거한다.

```
session.removeAttribute ( java.lang.String  name );
```

예) session.removeAttribute (“id“);

5. 모든 세션 삭제

현재 생성된 세션을 무효화 시킨다.

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

session

❖ 쿠키와 세션의 차이점

	쿠키(Cookie)	세션(Session)
저장 위치	클라이언트(=접속자 PC)	웹 서버
저장 형식	text	Object
만료 시점	쿠키 저장시 설정 (브라우저가 종료되도, 만료시점이 지나지 않으면 자동삭제되지 않음)	브라우저 종료시 삭제 (기간 지정 가능)
사용하는 자원(리소스)	클라이언트 리소스	웹 서버 리소스
용량 제한	총 300개 하나의 도메인 당 20개 하나의 쿠키 당 4KB(=4096byte)	서버가 허용하는 한 용량제한 없음.
속도	세션보다 빠름	쿠키보다 느림
보안	세션보다 안좋음	쿠키보다 좋음