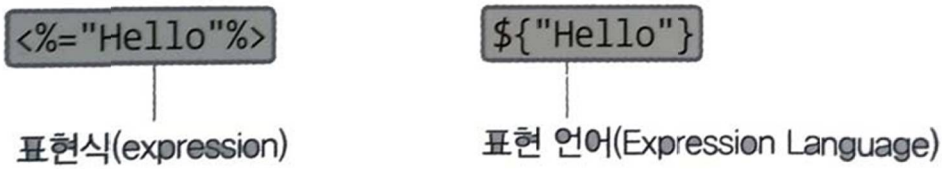
목차

1. EL(Exression Language)의 표기법

- Expression Language
- JSP 출력에 대한 부분을 쉽게 하기 위해 개발한 Tag 스크립트 언어
- EL 은 JSTL 태그와 결합하면 간단하고 편리한 표기를 사용하여 복잡한 작동이 표현될 수 있다.
- EL 익스프레션은 달러 표시($)와 중괄호 ({}) 를 앞에 붙여 사용하여 범위를 정한다.

표현식	표현 언어
<%=expr%>	${expr}
<%="hello"%>	${"hello"}

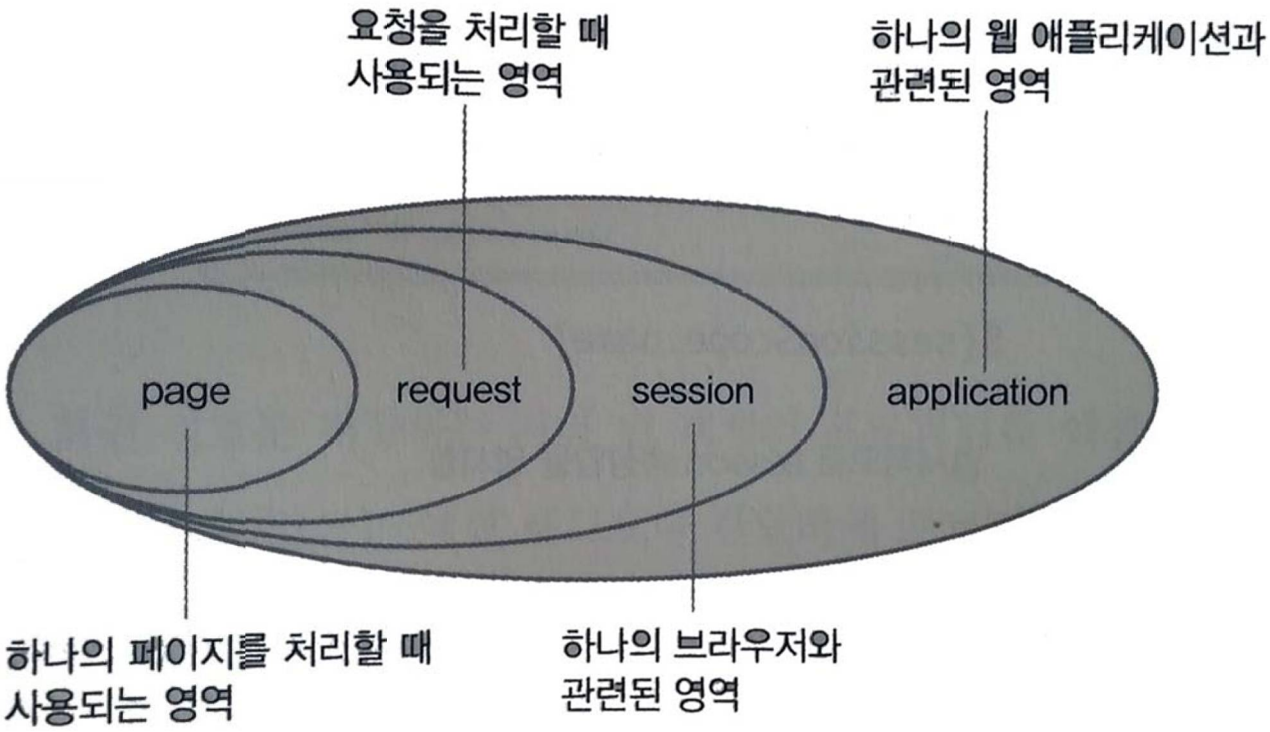


1.1 EL 의 내장 객체

구분	식별자	설명
JSP	pageContext	현재 페이지의 프로세싱과 상응하는 PageContext 인스턴스
요청 매개변수	param	요청 매개변수의 기본값을 저장하는 Map
	paramValues	요청 매개변수의 모든 값들을 String 배열로 저 하는 Map

1.2 EL 의 Scope 내장 객체

구분	식별자	설명
Scope	pageScope	page scope 의 변수들 request scope 의 변수들
	requestScope	
	sessionScope	session scope 의 변수들
	applicationScope	application scope 의 변수들



이 밖에 header 정보, cookie 정보와 관련된 내장 객체들도 있으나 아직 활용 전이므로 나중에 다시 살펴보도록 하자.내장 객체 뿐 아니라 Request Scope 도 JSP 와 동일하며 연산자도 지원한다.

1.3 EL 의 연산자

종류	연산자
관계	+, -, *, /, div, %, mod
산술	<, lt, >, gt, =, ge, ==, eq, !=, ne
논리	&&, and, ||, or, !, not
조건	a ? b : c
null 검사	empty

연산자를 이용하여 EL 을 표기 할 때는 아래와 같이 쓸 수 있다.

${item.price * (1 + taxRate[user.address.zipcode])}

관계형 연산자 및 논리적 연산자를 사용하는 EL 표기법은 아래와 같다.

${(x >= min) && (x <= max)}

EL 에서의 리터럴은 숫자, 캐릭터 스트링, 부울, null 등이 존재한다. 캐릭터 스트링은 싱글 쿼테이션 또는 더블 쿼테이션으로 지정된다. 부울 값은 true 와 false 로 계산된다

1.4 EL 로 표현 단순화 하기

보통 DB 를 통해 가져온 데이터를 표현하기 위해서는 Java 에서 지정한 타입명으로 표현을 할 수 있다.
자바에서 setAttribute("name", name); 과 같이 쓴 경우

${name}

자바에서 setter, getter 로 생성한 Object 를 쓴 경우 setAttribute("empBean" , empBean) 와 같이 담을 수 있는데 이럴 경우 JSP 에서는 EL 로 아래와 같이표기 할 수 있다.

${empBean.empNo}

JSP 에서는 setter, getter 를 붙여줘야 할 필요가 없이

인스턴스명.필드명 으로 사용해야 한다.

만약, 저장객체가 리스트형일 경우 아래와 같이 표현할 수 있다.

```
List list = (List)request.getAttribute("list"); list.get(0);

${list["0"]}
```

attribute 저장된 list 를 가져온다
[]안에 값은 list 의 키값 or 프로퍼티명 or 인덱스이다
[]의 왼쪽은 Map, Beans, 배열, List 타입이 올 수 있다.

1.5 EL 연습

1.5.1 http://localhost/el/el01

```
@Controller
public class JstlController {

    @RequestMapping(value = "/el/el01", method = RequestMethod.GET)
    public String jstl01(Model model) {
        logger.info("el01");

        return "el/el01";
    }
}
```

✓ views/el/el01.jsp

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

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt"%>

<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>EL 코어 라이브러리 : set</title>
</head>
<body>

<!--표현 언어(EL, Expression Language : 이렇게 값을 가져오면  편하다-->
${"Hello"} <br>
<%="Hello"%> <br> <!--스크립트릿-->
<% out.println("Hello"); %> <br> <!--표현식( Expression)-->

<hr />

정수형 : ${10} <br> 실수형 : ${5.6} <br>
문자열형: ${"성윤정"} <br> 논리형: ${true} <br>
null : ${null} <br>

<hr />
```

```
\${5+2} : ${5+2} <br>
\${5/2} : ${5/2} <br>
\${5 mod 2} : ${5 mod 2}<br>
\${5 > 2} : ${5 > 2}<br>
\${2 gt 10} : ${2 gt 10}<br>
\${(5 > 2) ? 5 : 2} : ${(5 > 2) ? 5 : 2}<br>
\${(5 > 2) || (2 < 10)} : ${(5 > 2) || (2 < 10)}<br>
<%
String input=null;
%>
\${empty  input} : ${empty  input}<br>

</body>
</html>
```

Hello
Hello
Hello

정수형 : 10
실수형 : 5.6
문자열형: 성윤정
논리형: true
null :

${5+2} : 7
${5/2} : 2.5
${5 mod 2} : 1
${5 > 2} : true
${2 gt 10} : false
${(5 > 2) ? 5 : 2} : 5
${(5 > 2) || (2 < 10)} : true
${empty input} : true

1.5.2 http://localhost/el/el02

```
@Controller
public class JstlController {

    @RequestMapping(value = "/el/el02", method = RequestMethod.GET)
    public String jstl02(Model model) {
        logger.info("el02");

        return "el/el02";
    }
}
```

✓ views/el/el02.jsp

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

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt"%>

<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Insert title here</title>
</head>
<body>
    <form method="get" action="/el/el02sub" >
        <label for="userid"> 아이디 : </label>
        <input type="text" name="id" id="userid"><br>
        <label for="userpwd"> 암   호 : </label>
        <input type ="password" name="pwd" id="userpwd"><br>
        <input type="submit" value="로그인">
    </form>
</body>
</html>
```

http://localhost/el/el02sub

```
@Controller
public class JstlController {

    @RequestMapping(value = "/el/el02sub", method = RequestMethod.POST)
    public String jstl02sub(Model model) {
        logger.info("el02sub");

        return "el/el02sub";
    }
}
```

✓ views/el/el02sub.jsp

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

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt"%>

<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
    당신이 입력한 정보입니다(표현식 방식).
    <hr>
    아이디 :
    <%=request.getParameter("id")%>
    <br> 비밀번호 :
    <%=request.getParameter("pwd")%>
    <br>

    <br> 당신이 입력한 정보입니다(EL 방식)
    <hr>
    아이디 : ${param.id}
    <br>
    비밀번호 : ${param["pwd"]}
</body>
</html>
```

당신이 입력한 정보입니다(표현식 방식).

아이디 : DSFAD
비밀번호 : FADFA

당신이 입력한 정보입니다(EL 방식).

아이디 : DSFAD
비밀번호 : FADFA

1.5.3 http://localhost/el/el03

```
@Controller
public class JstlController {

    @RequestMapping(value = "/el/el03", method = RequestMethod.GET)
    public String jstl03(Model model) {
        logger.info("el03");

        return "el/el03";
    }
}
```

✓ views/el/el03.jsp

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

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
EL 식
<hr>

== 연산자 사용 결과 : ${param.id=="pinksung"}
<br>
</body>
</html>
```

EL 식

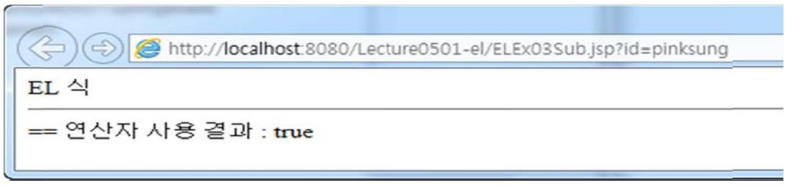
== 연산자 사용 결과 : true

1.5.4 http://localhost/el/el04

```
@Controller
public class JstlController {

    @RequestMapping(value = "/el/el04", method = RequestMethod.GET)
    public String jstl04(Model model) {
        logger.info("el04");

        return "el/el04";
    }
}
```

✓ views/el/el04.jsp

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

<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
<form action="/el/el04sub" method="post" enctype="application/x-www-form-urlencoded">

    숫자 1 :
    <br> 숫자 2 :
    <br>

    <input type="text" name="num1">
    <input type="text" name="num2">

    <input type="submit" value="전송">
</form>
</body>
</html>
```



http://localhost/el/el04sub

```
@Controller
public class JstlController {

    @RequestMapping(value = "/el/el04sub", method = RequestMethod.POST)
    public String jstl04sub(Model model) {
        logger.info("el04sub");

        return "el/el04sub";
    }
}
```

✓ views/el/el04sub.jsp

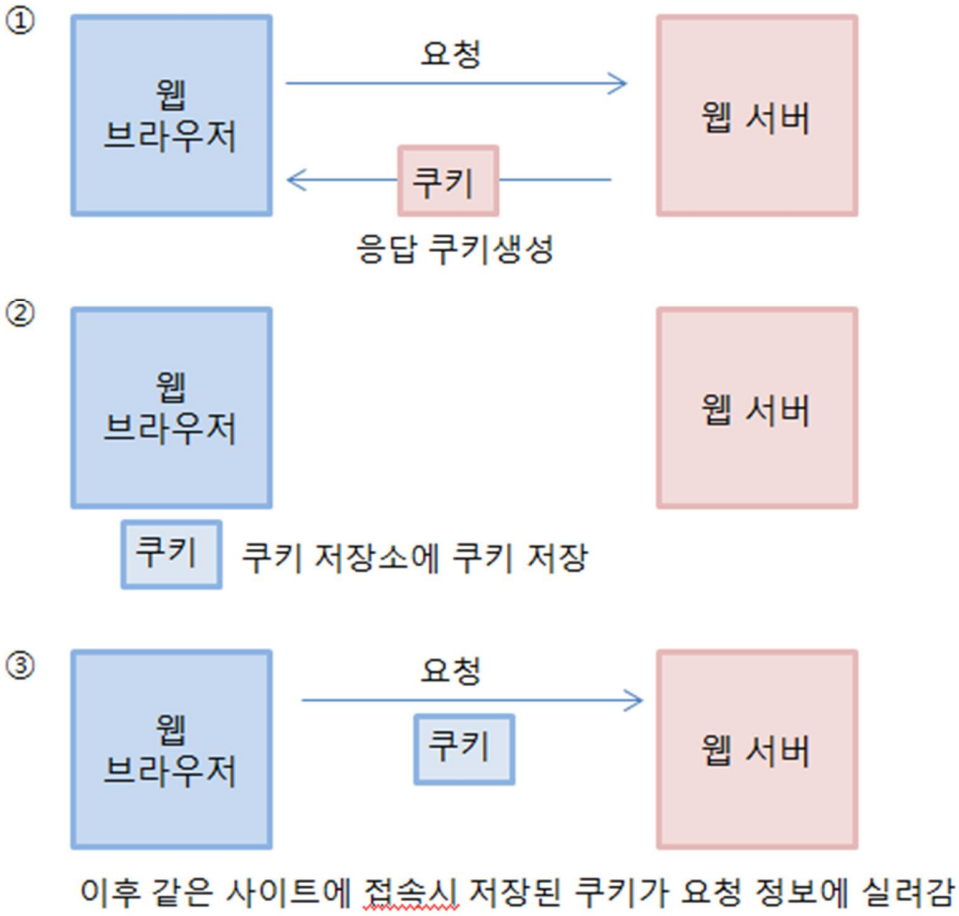
```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>


<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
<form action="/el/el04sub" method="post" enctype="application/x-www-form-urlencoded">
    숫자 1 : <input type="text" name="num1">
    <br>
    숫자 2 : <input type="text" name="num2">
    <br>

    <input type="submit" value="전송">
</form>
</body>
</html>
```

2. 쿠키와 세션의 차이점

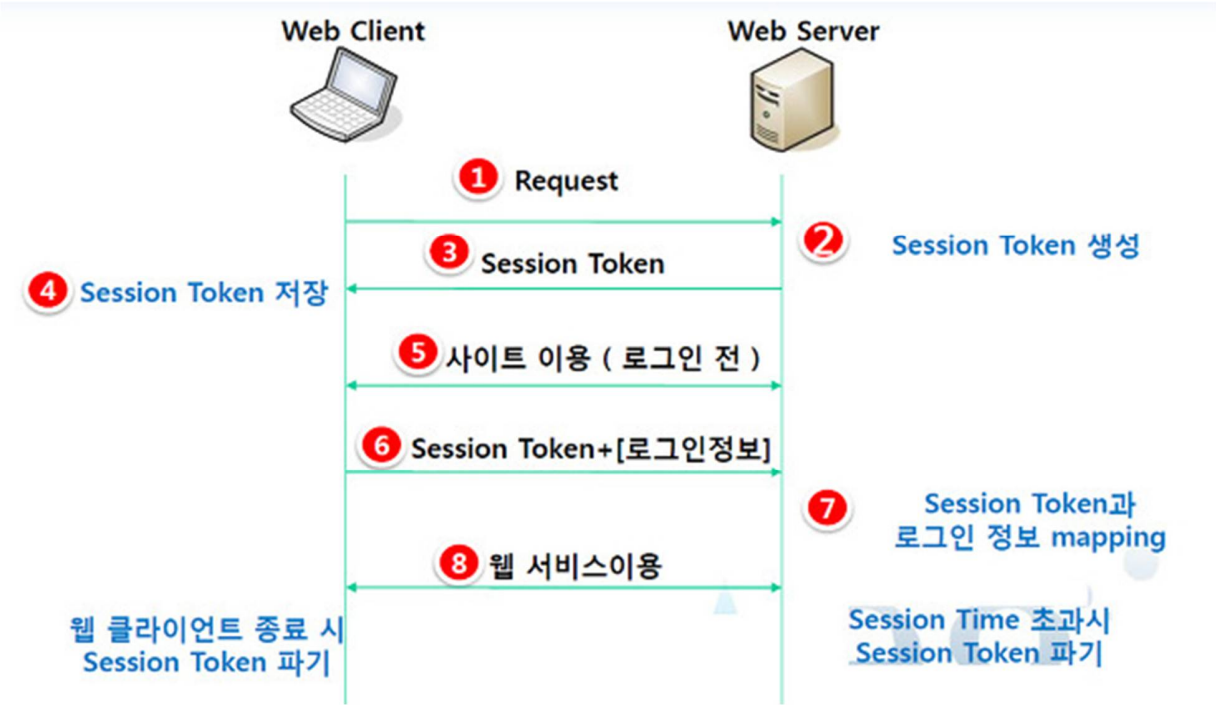
- 쿠키란?
    - 쿠키란 서버측에서 클라이언트측에 상태 정보를 저장하고 추출할 수 있는 메커니즘
    - 클라이언트의 요청마다 웹 브라우저로부터 서버로 전송되는 정보의 일종
    - HTTP 에서 클라이언트의 상태 정보를 클라이언트의 하드 디스크에 저장하였다가 필요 시 정보를 참조하거나 재사용할 수 있음.
    - 사용 예
        - 방문했던 사이트에 다시 방문 하였을 때 아이디와 비밀번호 자동 입력
        - 팝업에서 "오늘 이 창을 다시 보지 않음" 체크
    - 쿠키의 제약조건
        - 클라이언트에 총 300 개까지 쿠키를 저장할 수 있다
        - 하나의 도메인 당 20 개의 값만을 가질 수 있다
        - 하나의 쿠키 값은 4096Byte 까지 저장 가능하다
- 세션이란?
    - 세션이란 클라이언트와 웹서버 간에 네트워크 연결이 지속적으로 유지되고 있는 상태를 말함
    - 클라이언트가 서버에 처음 접속하면 클라이언트에 대하여 유일한 ID 를 부여하게 되는데, 이 ID 를 세션이라 부른다
    - 세션 ID 를 이용하여 페이지 이동 시나 클라이언트가 재 접속시 클라이언트를 구분할 수 있는 유일한 수단이 된다
    - 세션의 장점
        - 각각의 클라이언트마다 고유의 ID 부여
        - 세션 객체마다 저장해 둔 데이터를 이용하여 서로 다른 클라이언트의 요구에 맞게 서비스 제공
        - 클라이언트 자신만의 고유한 페이지를 열어놓아서 생길 수 있는 보안상의 문제 해결 용이

구분	쿠키	세션
저장 위치	클라이언트	서버
저장 형식	Text 로 저장	Object 로 저장
종료 시점	expire data 가 지났거나 expire data 설정하지 않았으면 브라우저 종료 시나	브라우저를 닫거나, 서버에 의해 지워지는 경우
자 원	클라이언트의 자원을 사용	서버의 자원을 사용
용도	사이트 재 방문시 사용자 정보를 기억하기 위해 사용 (ID, PW, 팝업창 제한등)	서버를 이용하는 동안에 사용자 정보를 유지하기 위해 사용
용량 제한	한 도메인 당 20 개, 쿠키 하나 당 4KB, 총 300 개	서버가 허용하는 한 용량에 제한이 없음

2.1 쿠키 생성과 사용, 소멸



2.2 세션의 생성과 사용, 소멸



3. page 지시어(Directive)

3.1 page 지시어란?

JSP 페이지의 타입, 스크립팅 언어, import 할 클래스, 세션 및 버퍼의 사용 여부, 버퍼의 크기 등 JSP 페이지에서 필요한 정보를 지정할 때 사용되는 속성들이다.

3.2 page 디렉티브의 속성

속성명	사용법	속성 설명
info	info="설명.."	페이지를 설명해 주는 문자열을 지정하는 속성
language	language="java"	JSP 페이지의 스크립트 요소에서 사용할 언어를 지정하는 속성
contentType	contentType="text/html;charset=utf-8"	JSP 페이지가 생성할 문서의 타입을 지정하는 속성
extends	extends="system.MasterClass"	자신이 상속 받을 클래스를 지정할 때 사용하는 속성
import	import="java.util.Vector" import="java.util.*"	다른 패키지에 있는 클래스를 가져다 쓸 때 사용하는 속성
session	session="true"	HttpSession 을 사용할지 여부를 지정하는 속성
buffer	buffer="10kb"	buffer="none" JSP 페이지의 출력 버퍼의 크기를 지정하는 속성
autoFlush	autoFlush="false"	출력 버퍼가 다 찰 경우에 저장되어 있는 내용의 처리를 설정 하는 속성
isThreadSafe	isThreadSafe="true"	현 페이지에 다중쓰레드를 허용할지 여부를 설정하는 속성
errorPage	errorPage="error/fail.jsp"	에러 발생 시 에러를 처리할 페이지를 지정하는 속성
isErrorPage	isErrorPage="false"	해당 페이지를 에러 페이지로 지정하는 속성
pageEncoding	pageEncoding="UTF-8"	해당 페이지의 문자 인코딩을 지정하는 속성
isELignored	isELIgnored="true"	표현 언어(EL)에 대한 지원 여부를 설정하는 속성

3.3 page 지시어 사용 예제

3.3.1 JSP 페이지에서 세션을 사용하기 위한 설정

```
<%@ page session="true" %>
```

3.3.2 JSP 페이지를 "utf-8"로 인코딩하기 위한 설정

```
<%@ page contentType = "text/html;charset=utf-8" pageEncoding="utf-8" %>
```