

커스텀 태그란?

- 커스텀 태그(Custom Tag)

- 사용자 정의 태그를 의미함
- 스크립트릿 사용을 줄이고 태그와 같은 형태로 프로그램 코드를 대체하거나 재활용 가능한 구조를 통해 태그 라이브러리로 활용하고자 개발된 규격임
- 외형적인 형태는 XML(HTML) 태그 구조이지만 서블릿 형태로 변환될 때 자바 코드로 변경되어 통합되는 방식
- 커스텀 태그를 사용하기 위해서는 taglib 지시어를 사용하여 커스텀 태그가 어디에 정의되어 있는지를 먼저 선언해야 하며 태그에 사용할 접두어를 지정해야 함
- 커스텀 태그 자체가 서버에서 해석되는 구조이며, 프로젝트가 특정 커스텀 태그에 종속될 수 있다는 문제 때문에 커스텀 태그를 직접 만드는 방식은 점차 줄어들고 있음
 - 대신 커스텀 태그 기술로 만들어진 JSTL(JSP Standard Tag Library)이 자바 웹 개발에 꼭 필요한 요소가 됨

커스텀 태그란?

● 커스텀 태그의 예시

- 특정 상품 코드를 전달하면 해당 상품에 대한 세부 정보를 출력하기

```
<%@ taglib tagdir="/WEB-INF/tags" prefix="m" %>
```

```
<m:printData pid="87459989" />
```

- 태그 파일로 정의된 커스텀 태그를 사용하며 'WEB-INF/tags/printData.tag' 파일로부터 태그 정의를 가지고 옴
- m은 태그 앞에 붙일 접두어로 태그 파일명이 태그 이름이 됨

EL이란?

- 표현 언어(Expression Language, EL)
 - 주로 EL이라 불림
 - 현재 페이지의 자바 객체 혹은 scope object에 저장된 자바 빈 객체를 손쉽게 접근하고 사용할 수 있게 함
 - 데이터를 표현하기 위한 용도로 설계되었지만, 제한된 객체 참조가 가능하며 해당 객체의 메서드 호출도 가능함
 - EL은 단순한 출력 외에도 사칙연산, 비교연산, 논리연산, 3항 연산 등을 지원함
 - 연산 기능은 핵심 로직의 구현보다는 상황에 따라 출력값을 변경하는 정도의 용도로 사용하는 것이 좋음

EL이란?

● EL의 장점

- 간단한 구문으로 손쉽게 변수/객체를 참조할 수 있음
- 데이터가 없거나 null 객체를 참조할 때 에러가 발생하지 않음

● 자바 빈 접근

- EL을 통해 scope object에 저장된 자바 빈 객체를 참조하는 방법

```
${저장이름.변수명}
```

- 앞의 자바 빈 설명에서 만든 Member 클래스의 멤버 정보에 접근하기
 - 컨트롤러에 의해 session에 저장하는 과정이 있었다고 가정함

```
<h2>멤버 정보< /h2>  
이름: ${m.name} <br>
```

EL이란?

- EL을 사용하지 않을 경우

- 다음과 같이 표현식을 사용하거나 <jsp:getProperty> 액션으로 출력할 수도 있음

이름: <%= m.name %>
 // 표현식 사용

이름: <jsp:getProperty name="m" property="name" />
 // 액션 사용

- 표현을 위한 3가지 방법 : 예제 01_el
- 데이터 타입에 따른 EL : 예제 02_el.jsp

표현 언어로 표현을 단순화 하기

- 표현 언어에서 사용 가능한 데이터 타입으로 문자열, 정수, 실수, 논리형, null이 있다
- 다만 null은 공백으로 출력한다. : 예제 02_el.jsp
- 표현언어 내부에서 사용가능 한 연산자

종류	연산자
산술	+, -, *, /(div) ,% (mod)
관계형	==(eq), !=(ne), <(lt), >(gt), <=(le), >=(ge)
조건형	a ? b : c
논리형	&& (and), (or), ! (not)
null 검사	empty

비교 표현식

eq - equal (=)

ne - not equal (<>)

lt - little (<)

le - little or equal (<=)

gt - greater (>)

ge - greater or equal (>=)

구분	표현식	비교
크다	a gt b	a > b
작다	a lt b	a < b
크거나 같다	a ge b	a >= b
작거나 같다	a le b	a <= b
같다	a eq b	a == b
같지 않다	a ne b	a != b

표현 언어로 표현을 단순화 하기

- 연산자는 기호와 텍스트 둘 다 사용 가능하다

`${3==3}`

`${3 eq 3}`

- `empty` 는 객체가 비었는지 확인할 때 사용=> 비어있다면 `true` 반환

`${empty input}`

- 예제 03_el.jsp
- 후에 나올 `jstl`와 함께 사용하면 보다 가독성 높은 코드를 작성할 수 있다

EL이란?

● EL 연산

- 기본적인 사칙연산, 비교연산, 논리연산, 3항 연산 등이 가능함

```
${10 + 20}           // 사칙연산, 30  
${10 * 20}           // 사칙연산, 200  
${true && false}     // 논리연산, false  
${10 >= 20}          // 논리연산, false  
${user.name == "홍길동"? "교수" : "학생"} // 3항 연산, 이름이 홍길동이면 교수 출력
```

● 배열, 맵 데이터 연동

- 참조하는 객체가 배열이나 맵 형태인 경우 다음과 같이 사용함

```
${myList[0]}        // 배열인 경우  
${myMap["name"]}     // 맵인 경우
```

표현 언어로 요청 파라미터 처리하기

- 요청 처리는 `request.getParameter()`를 사용한다.
- 다만 표현언어에서는 `param`객체를 사용한다.

내장 객체	설명
<code>param</code>	JSP의 내장 객체인 <code>request</code> 의 <code>getParameter()</code> 와 동일한 역할인 파라미터의 값을 알려 준다
<code>paramValues</code>	동일한 이름으로 전달되는 파라미터 값들을 배열 형태로 얻어오는 데 사용하는 <code>request</code> 의 <code>getParameterValues()</code> 와 동일한 역할을 한다.

- `param`객체는 `.`또는 `[]`로 사용자의 입력값을 가져온다.
- 예제 04_login.jsp, 04_testLogin.jsp

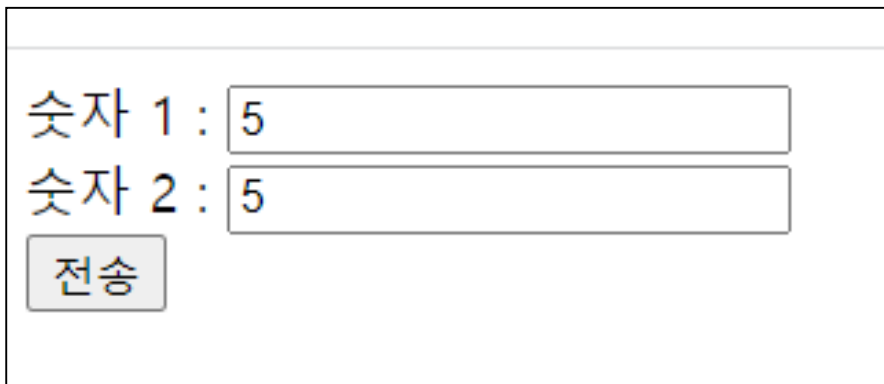
EL로 요청 파라미터 처리하기

- 표현 언어에서 null처리는 공백으로 사용된다.
- 또한 기존 Java에서 객체 비교를 위해선 equals메소드를 사용해야 하는 반면 표현언어로는 ==으로 객체에 저장된 값을 비교할 수 있다.
- 그리고 표현언어는 문자열을 숫자로 변환할 필요도 없다.

- 연습문제01 : EL로 형변환 없이 두 수를 입력 받아 합을 구하기

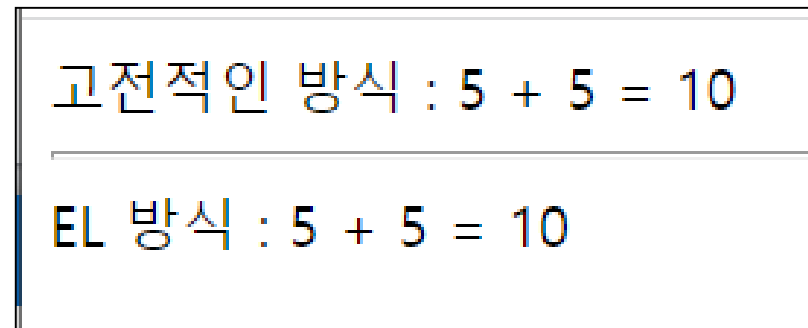
• addForm.jsp

addition.jsp



숫자 1 :

숫자 2 :



고전적인 방식 : 5 + 5 = 10

EL 방식 : 5 + 5 = 10

EL이란?

- Scope Object 접근

- EL은 기본적으로 모든 scope에서 자바 빈 객체를 찾음
- 만일 특정 scope만을 대상으로 참조하려면 '내장객체명Scope.속성이름'으로 사용
 - 예) session과 request 모두에 'm'이라는 이름으로 저장된 객체가 있다고 할 때, request scope에 있는 객체를 참조하려면 다음과 같이 사용할 수 있음

이름: `${requestScope.m.name}`

표현 언어로 내장 객체에 접근 하기

- JSP에서 웹 애플리케이션을 구현하는 데 필요한 정보를 JSP의 내장 객체에 속성값으로 저장해서 사용했다.
- 각 속성에 저장된 값을 표현언어에서는 다음과 같은 형태로 사용할 수 있다.

카테고리	내장 객체	설명
범위	pageScope	page 기본 객체에 저장된 속성의<속성,값> 매핑을 저장한 Map객체
	requestScope	request기본 객체에 저장된 속성의<속성,값> 매핑을 저장한 Map객체
	sessionScope	session 기본 객체에 저장된 속성의<속성,값> 매핑을 저장한 Map객체
	applicationScope	application 기본 객체에 저장된 속성의<속성,값> 매핑을 저장한 Map객체

<%=name%>

자바의 변수이름

\${name}

속성 이름

표현 언어로 내장 객체에 접근 하기

- JSP 내장 객체에 정보를 주고 받기 위해서는 다음 메소드를 사용한다.

메소드	설명
setAttribute(name,value)	주어진 이름(name)에 값(value)을 설정한다
getAttribute(name)	주어진 이름(name)에 설정된 값을 얻어온다
getAttributeNames()	현재 객체에 관련된 모든 속성의 이름을 얻어온다
removeAttribute(name)	주어진 이름(name)에 설정된 값(value)을 제거한다

- JSP 만으로 웹 프로그래밍을 구현하는 방법을 모델1 이라고 한다.
- 비즈니스 로직은 서블릿에서 전담하고 JSP는 결과 출력에 집중하는 방식을 모델 2라고 한다.

표현 언어로 내장 객체에 접근 하기

- JSP 각 내장 객체를 표현언어에서는 각각 어떻게 접근하는 지 표를 통해 알아보자

속성	JSP 내장객체	표현 언어의 내장객체	서블릿 클래스
page 속성	pageContext	pageScope	javax.servlet.jsp.jspContext 클래스
request 속성	request	requestScope	javax.servlet.ServletRequest인터페이스
session 속성	session	sessionScope	javax.servlet.http.HttpSession인터페이스
application 속성	application	applicationScope	javax.servlet.ServletContext인터페이스

표현 언어로 내장 객체에 접근 하기

- page 객체에 저장된 값을 얻어오는 법

자바 코드	표현 언어
pageContext.getAttribute("num1");	<code>\${pageScope.num1}</code>

- request 객체에 저장된 값을 얻어오는 법

자바 코드	표현 언어
request.getAttribute("num1");	<code>\${requestScope.num1}</code>

- session 객체에 저장된 값을 얻어오는 법

자바 코드	표현 언어
session.getAttribute("num1");	<code>\${sessionScope.num1}</code>

- application 객체에 저장된 값을 얻어오는 법

자바 코드	표현 언어
application.getAttribute("num1");	<code>\${applicationScope.num1}</code>

표현 언어로 내장 객체에 접근 하기

- 다만 표현언어에서 내장 객체에 데이터를 접근할 때 어느 객체에 접근할지 생략할 수 있다.

`${num1}` 형태로 사용 가능하다.

- 이때 num1 속성의 값을 가져올 때 어느 내장 객체인지 알 수 없으므로 표현언어에서 사용할 때는 다음 순서로 자동으로 검색해서 해당 속성이 있을 때 가져온다.

pageScope -> requestScope -> sessionScope -> applicationScope

- 예제 07_scope.jsp