

# Chapter 14

## 표현 언어(Expression Language)

이번 Chapter에서는 JSP 페이지에서 스크립트릿과 같은 자바 코드를 대체하는 표현 언어(EL : Expression Language)의 기본 사용 방법과 제공하는 연산자 및 내장 객체에 대해 학습한다.



## 표현 언어의 개요

기존 방식과 표현 언어를 사용한 방식  
비교하기

표현 언어의 특징 및 기능

표현 언어의 작성 방법



## 표현 언어의 연산자와 내장 객체

표현 언어의 연산자

표현 언어의 내장 객체

- 파일 JSP페이지에 사용되는 선언문 (<%!%>), 스크립트릿(<%%>), 표현식 (<%= %>)과 같은 자바 코드를 대신
- 표현 언어는 좀 더 쉽고 자연스러운 형태로 태그(주로 액션태그)의 속성 값을 지정하고 객체의 메소드에 지정하는 방법을 제공



# 표현 언어의 개요

## 기존 방식과 표현 언어를 사용한 방식 비교하기

### ■ 유동적인 속성 값 지정

#### ■ 기존 코드

- `<someTags:aTag attribute="<%=aName%>">`

#### ■ 표현 언어

- `<someTags:aTag attribute="${aName}">`



# 표현 언어의 개요

## 기존 방식과 표현 언어를 사용한 방식 비교하기

### ■ 객체의 getter 메소드에 접근

#### ■ 기존 코드

- `<%=aCustomer.getAddress()%>`

#### ■ 표현 언어

- `${aCustomer.address}`





# 표현 언어의 개요

## C 표현 언어의 특징 및 기능

### ■ 특징

- 파라미터 값이 null 이어도 상관없음
- 파라미터 값의 파싱을 신경 쓰지 않아도 됨



# 표현 언어의 개요

## 표현 언어의 특징 및 기능

### ■ 기능

- 변수와 연산자를 포함하고 함수를 호출할 수 있음
- JSP의 영역(page, request, session, application)에 저장된 어떤 속성 및 자바 빈이라도 표현 언어 (EL)의 변수로서 사용
- 내장 객체도 지원



## 표현 언어의 작성 방법

- ▣ 표현 언어의 표현식은 숫자, 문자열, boolean 값 및 null같은 상수 값(리터럴)들도 포함할 수 있음
- ▣ 표현 언어는 \$와 표현식 그리고 {}를 사용해서 표현
- ▣ 작성 방법
  - 표현 언어는 항상 \${ 로 시작해서 }로 끝남
    - ▣ \${num}

## 표현 언어의 작성 방법

### ■ 작성 방법

- 표현식 안에 연산식도 쓸 수 있음
  - `${num + 1}`
- 표현식에는 브라켓 연산자(bracket ([[]] operator)를 사용할 수 있음
  - `${article["num"] + 1}`
- 동적으로 값을 받도록 JSTL이나 커스텀 태그의 JSP 액션의 속성에 값을 지정할 때도 사용할 수 있음
  - `<c:out value="${article.num + 1}"/>`

## C 표현 언어의 연산자

연산자	설 명
.	빈의 프로퍼티나 맵(Map)의 엔트리에 접근
[]	배열이나 리스트(List)의 엘리먼트에 접근
()	괄호. 표현식의 연산 순서를 바꿔서 연산 할 때 사용
a?b:c	조건 테스트 , 조건(a) ? true일때 리턴값(b) : false일 때 리턴값 (c)
+	더하기
-	빼기
*	곱하기
/ 또는 div	나누기

## C 표현 언어의 연산자

연산자	설 명
% 또는 mod	나머지
== 또는 =	같다
!= 또는 !=	같지 않다
< 또는 lt	보다 작다
> 또는 gt	보다 크다
<= 또는 le	작거나 같다
>= 또는 ge	크거나 같다
&& 또는 and	논리 AND

## C 표현 언어의 연산자

연산자	설 명
(' '가 연속으로 두개) 또는 or	논리 OR
! 또는 not	논리 not (true를 false로, false를 true로 변환)
empty	빈 변수 값 체크. null, 빈 문자열, 빈 배열, 엔트리가 없는 맵(Map)이나 컬렉션(Collection) 인가를 묻는 조건식에 사용
func(args)	함수(클래스에서 정의한 메소드) 호출. func는 임의의 함수 이름이고 args는 매개변수 리스트로 없을 수도 있음 $\{ns:func(args1, args2,...)\}$ 과 같이 사용된다. 여기서 ns는 func() 메소드가 속한 클래스의 prefix



## 표현 언어의 내장 객체

- 내장 객체는 `${sessionScope.id}`와 같이 표현식에서 사용
- 종류
  - `sessionScope`
    - session 영역 객체
  - `applicationScope`
    - application 영역 객체
  - `param`
    - 요청 파라미터 객체. 파라미터 값을 얻어낼 때 사용하는 것으로 `request.getParameter()`와 같은 역할을 수행



## C 표현 언어의 내장 객체

### □ 종류

- pageScope
  - page 영역 객체
- requestScope
  - request 영역 객체
- paramValues
  - 요청 파라미터 컬렉션. 복수의 값을 갖는 파라미터로부터 값을 얻어낼 때 사용하는 것으로 `request.getParameterValues()`와 같은 역할을 수행

## 표현 언어의 내장 객체

### 종류

#### ■ header

- HTTP 요청 헤더 객체. 헤더 값을 얻어낼 때 사용하는 것으로 `request.getHeader()`와 같은 역할을 수행

#### ■ headerValues

- HTTP 요청 헤더 객체 컬렉션. 복수의 값을 갖는 헤더로부터 값을 얻어낼 때 사용하는 것으로 `request.getHeaders()`와 같은 역할을 수행

#### ■ cookie

- 모든 쿠키 값 컬렉션. 요청객체로부터 모든 쿠키값을 얻어낼 때 사용하는 것으로 `request.getCookies()`와 같은 역할을 수행

## C 표현 언어의 내장 객체

### ▣ 종류

#### ■ initParam

- 모든 애플리케이션의 초기화 파라미터 이름을 얻어내는 컬렉션. `config.getInitParamer()`와 같다.

#### ■ pageContext

- 현재 JSP 페이지의 컨텍스트(Context). 주로 다른 내장 객체 `servletContext`, `session`, `request`, `response`를 구할 때 사용