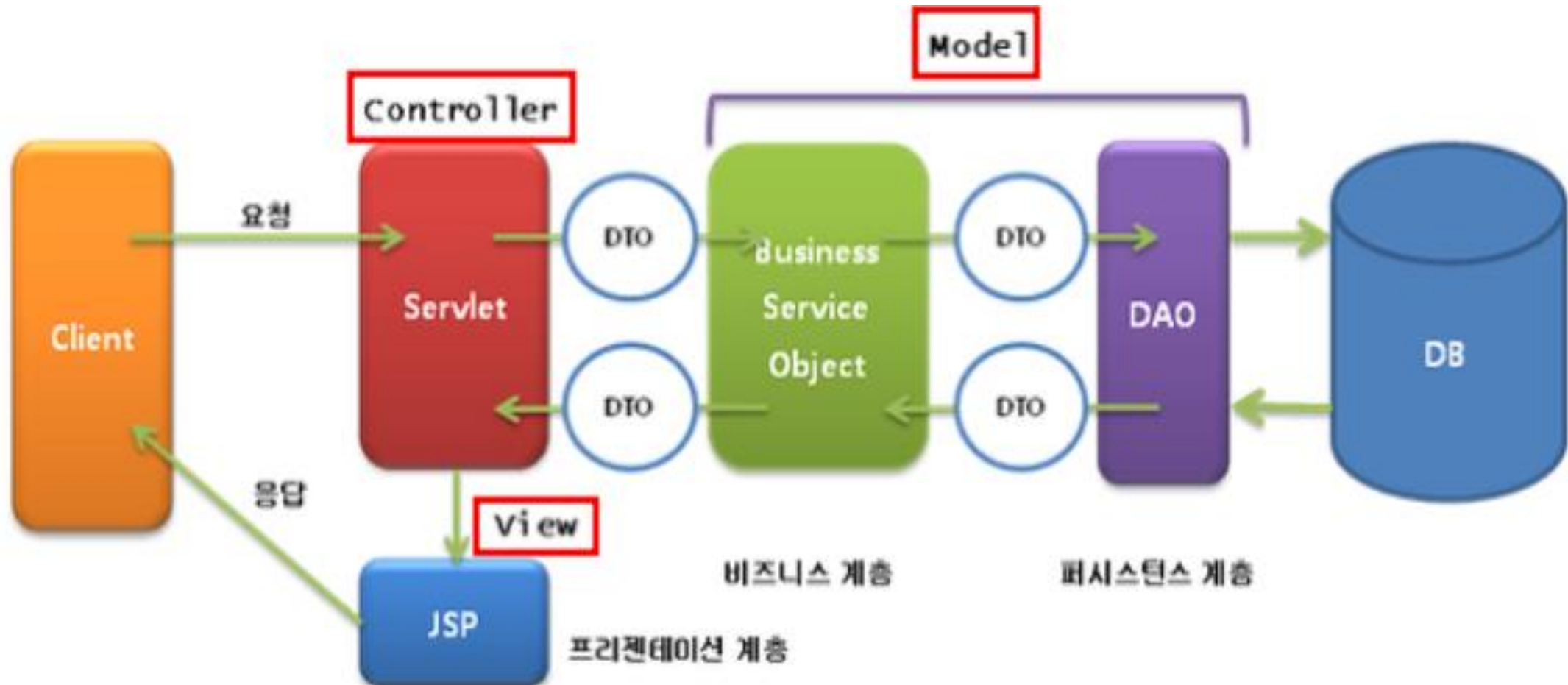


# 표현언어(Expression Language)

안화수

# 모델2 설계방식

## ❖ MVC Architecture



# 모델2 설계방식

## ❖ MVC 패턴

MVC는 Model / View / Controller 의 약자로 애플리케이션을 세 역할로 나누어서 개발하는 개발 방법론이다.

### ➤ Model

- 애플리케이션의 데이터 처리를 담당함
- Service 클래스 + DAO 클래스로 구현함

### ➤ View

- 사용자 인터페이스를 처리함
- JSP 를 이용해서 구현함 : EL(Expression Language, 표현언어) + JSTL(JSP Standard Tag Library)

### ➤ Controller

- 클라이언트의 요청을 받아 Model과 View사이에서 흐름을 조정한다.
- Java Servlet으로 구현함

# 표현언어

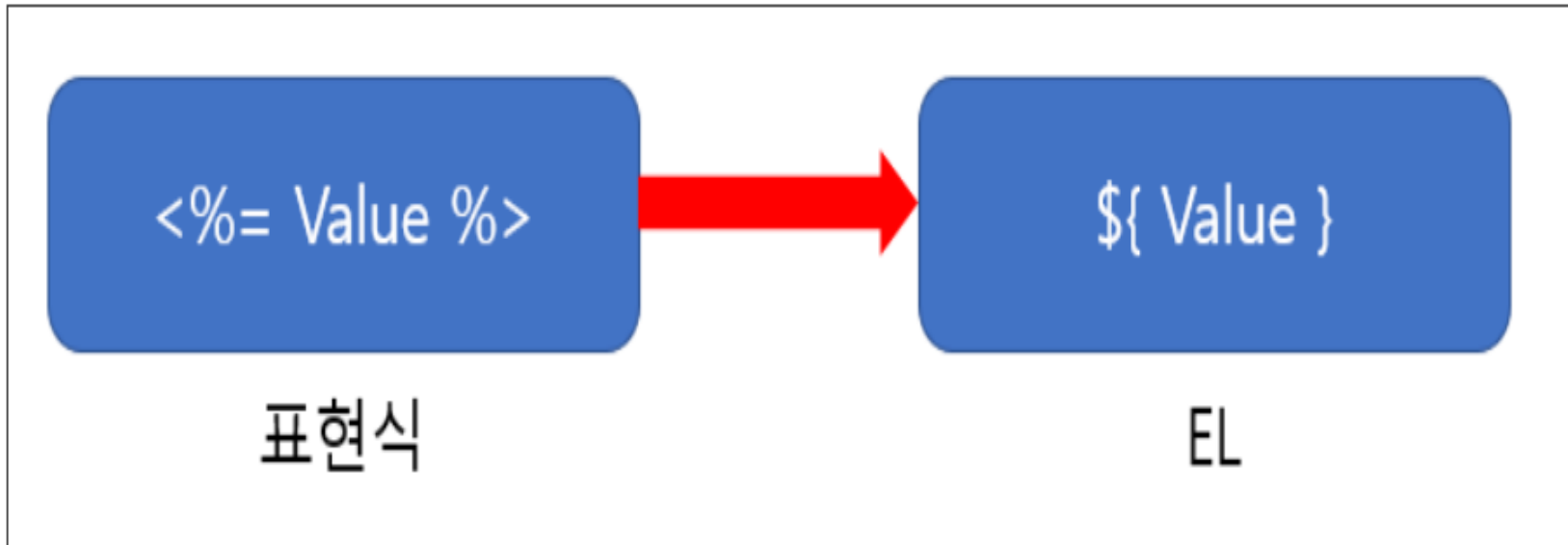
## ❖ 표현언어 (Expression Language)

- JSP의 표현식 태그와 비슷한 역할을 수행한다.
- 표현언어는 `${ }` 기호를 사용한다.
- 표현언어에 JSP의 네가지 영역 객체로 공유한 값을 구해올 수 있다.  
ex) `${ sessionScope.id }`
- 표현언어에 산술연산자(+,-,\*,/,%), 비교연산자(>,>=,<,<=,==,!=), 논리연산자(&&, ||, !), 조건연산자, empty 연산자 등을 사용할 수 있다.  
ex) `${ 10 + 20 }`
- 표현언어에 자바빈 클래스의 getter 메소드를 호출할 수 있다.  
ex) `${ article.id }`

# 표현언어

## ❖ 표현언어 (Expression Language)

- EL(Expression Language)은 자바 빈의 프로퍼티 값을 JSP의 표현식 `<%= %>` 이나 액션 태그 `<jsp:useBean>`를 사용하는 것 보다 쉽고 간결하게 사용할 수 있게 하는 기술이다.



# 연산자

## ❖ 산술 연산자

표현식	실행 결과
$\${10+20}$	30
$\${10-20}$	-10
$\${10*20}$	200
$\${10/20}$	0.5
$\${10 \div 20}$	0.5
$\${10 \% 20}$	10
$\${10 \bmod 20}$	10

# 연산자

## ❖ 비교 연산자

표현식	연산자
<code>\${10 == 11}</code>	false
<code>\${10 eq 11}</code>	false
<code>\${10 != 11}</code>	true
<code>\${10 ne 11}</code>	true
<code>\${10 &lt; 11}</code>	true
<code>\${10 lt 11}</code>	true
<code>\${10 &gt; 11}</code>	false
<code>\${10 gt 11}</code>	false
<code>\${10 &lt;= 11}</code>	true
<code>\${10 le 11}</code>	true
<code>\${10 &gt;= 11}</code>	false
<code>\${10 ge 11}</code>	false

# 연산자

## ❖ 논리 연산자

표현식	실행 결과
<code>\${true &amp;&amp; false}</code>	false
<code>\${true and false}</code>	false
<code>\${false    true}</code>	true
<code>\${false or true}</code>	true
<code>\${not true}</code>	false
<code>\${!true}</code>	false



# 연산자

## ❖ empty 연산자

- 값이 null이면 true
- 문자열, 배열, Map, Collection 객체의 크기가 0이면 true
- 그 외에는 false

```
<% // 공유 설정  
    request.setAttribute("title", "EL 연산자");  
%>
```

```
${empty title } <br>                                <!-- false -->  
${empty title2 } <br>                <!-- true -->
```

# EL의 기본객체

기본 객체	설명
pageContext	JSP의 page 기본 객체와 동일하다.
pageScope	<u>pageContext</u> 기본 객체에 저장된 속성의 <속성, 값> 매핑을 저장한 Map 객체.
<u>requestScope</u> \${requestScope.page}	request 기본 객체에 저장된 속성의 <속성, 값> 매핑을 저장한 Map 객체. <u>&lt;%=request.getAttribute("page") %&gt;</u>
<u>sessionScope</u> \${sessionScope.id}	session 기본 객체에 저장된 속성의 <속성, 값> 매핑을 저장한 Map 객체. <u>&lt;%=session.getAttribute("id") %&gt;</u>
applicationScope	application 기본 객체에 저장된 속성의 <속성, 값> 매핑을 저장한 Map 객체.
<u>param</u> \${param.name}	요청 <u>파라미터</u> 의 <파라미터이름, 값> 매핑을 저장한 Map 객체. <u>&lt;%=request.getParameter("name") %&gt;</u>
paramValues	요청 <u>파라미터</u> 의 <파라미터이름, 값배열> 매핑을 저장한 Map 객체.
header	요청 정보의 <헤더이름, 값> 매핑을 저장한 Map 객체.
headerValues	요청 정보의 <헤더이름, 값 배열> 매핑을 저장한 Map 객체.
cookie	<쿠키 이름, Cookie> 매핑을 저장한 Map 객체.
initParam	초기화 <u>파라미터</u> 의 <이름, 값> 매핑을 저장한 Map 객체.