

표현 언어(EXPRESSION LANGUAGE)



표현 언어(Expression Language)

- ❖ 값(데이터)을 웹 페이지에 표시(표현)하는데 사용되는 태그이다.
- ❖ JSP 출력에 대한 부분을 쉽게 하기 위해 개발한 태그이다.
- ❖ 표현 언어는 표현식보다 사용방법이 간단하고 문법체계가 직관적으로 쉬운 편이다.
표현 언어는 “\${ }”를 사용하여 값을 표현한다.

■ 예 : 표현식

```
<%=expr%>
```

```
<%="Hello"%>
```

표현 언어

■ 예 : 표

```
${expr}
```

```
${"Hello"}
```

실습 – 표현 언어

<파일 이름: EL_01.jsp>

<body>

`${"표현언어 EL로 출력"}
`

`<%= "스크립트릿 출력" %>
`

`<% out.print("표현식으로 출력"); %>`

</body>

표현언어의 데이터 타입과 연산자

- ❖ 정수형 : `${10}`

- ❖ 실수형 : `${5.6}`

- ❖ 문자열형 : `${"홍길동"}`

- ❖ 논리형 : `${true}`

- ❖ null값 : `${null}`
 // 결과 화면을 공백으로 처리
- ❖ `${5+2}` // 연산자도 표현 가능

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

표현 언어의 연산자

<h1>연산자들</h1>

$\mathbb{W}\{5 + 2\} : \{5 + 2\}$

$\mathbb{W}\{5 - 2\} : \{5 - 2\}$

$\mathbb{W}\{5 * 2\} : \{5 * 2\}$

$\mathbb{W}\{5 / 2\} : \{5 / 2\}$

$\mathbb{W}\{5 \text{ div } 2\} : \{5 \text{ div } 2\}$

$\mathbb{W}\{5 \% 2\} : \{5 \% 2\}$

$\mathbb{W}\{5 \text{ mod } 2\} : \{5 \text{ mod } 2\}$

표현 언어의 연산자

<h1>연산자들</h1>

$\mathbb{W}\{5 == 2\} : \{5 == 2\} \Rightarrow \mathbb{W}\{5 \text{ eq } 5\} : \{5 \text{ eq } 5\}$

$\mathbb{W}\{5 \neq 2\} : \{5 \neq 2\} \Rightarrow \mathbb{W}\{5 \text{ ne } 5\} : \{5 \text{ ne } 5\}$

$\mathbb{W}\{5 < 2\} : \{5 < 2\} \Rightarrow \mathbb{W}\{5 \text{ lt } 2\} : \{5 \text{ lt } 2\}$

$\mathbb{W}\{5 > 2\} : \{5 > 2\} \Rightarrow \mathbb{W}\{5 \text{ gt } 2\} : \{5 \text{ gt } 2\}$

$\mathbb{W}\{5 \leq 2\} : \{5 \leq 2\} \Rightarrow \mathbb{W}\{5 \text{ le } 2\} : \{5 \text{ le } 2\}$

$\mathbb{W}\{5 \geq 2\} : \{5 \geq 2\} \Rightarrow \mathbb{W}\{5 \text{ ge } 2\} : \{5 \text{ ge } 2\}$

표현 언어의 연산자

<h1>연산자들</h1>

$\mathbb{W}\{(5>2) \mid\mid (5<2)\} : \mathbb{S}\{(5>2) \mid\mid (5>2)\} \Rightarrow \mathbb{W}\{\text{false or true}\} : \mathbb{S}\{\text{false or true}\}$

$\mathbb{W}\{(5>2) \&\& (5>2)\} : \mathbb{S}\{(5>2) \&\& (5>2)\} \Rightarrow \mathbb{W}\{\text{false and true}\} : \mathbb{S}\{\text{false and true}\}$

$\mathbb{W}\{\!(5>2)\} : \mathbb{S}\{\!(5>2)\} \Rightarrow \mathbb{W}\{\text{not false}\} : \mathbb{S}\{\text{not false}\}$

표현 언어로 요청 파라미터 처리하기

- ❖ 사용자가 폼에 입력한 값을 얻어오기 위해서 JSP 내장 객체인 request의 `getParameter()` 메소드를 사용했었다.
- ❖ 표현 언어에서는 `request.getParameter()` 대신 `param` 객체를 사용한다.
- ❖ `param` 객체는 “.” 또는 “[]”를 사용하여 사용자가 입력한 값을 얻어온다.

내장 객체	설명
<code>param</code>	JSP의 내장 객체인 request의 <code>getParameter()</code> 와 동일한 역할인 파라미터 값을 알려준다.
<code>paramValues</code>	파라미터 값들을 배열 형태로 얻어오는데 사용하는 request의 <code>getParameterValues()</code> 와 동일한 역할을 한다.

실습 – param 객체

<파일 이름: loginParam.jsp>

```
<form method="get" action="testParam.jsp">
```

```
    <input type="text" name="id" placeholder="아이디"><br>
```

```
    <input type="password" name="pwd" placeholder="비밀번호"><br>
```

```
    <input type="submit" value="로그인">
```

```
</form>
```

실습 – param 객체

<파일 이름: testParam.jsp>

<h3>입력한 정보 출력(고전적인 방식)</h3>

아이디 : <%= request.getParameter("id") %>

비밀번호 : <%= request.getParameter("pwd") %>

<h3>입력한 정보 출력(EL 방식)</h3>

아이디 : \${param.id}

비밀번호 : \${param["pwd"]}

EL 방식으로 내장객체 사용하기

- ❖ 어트리뷰트에 저장된 값을 표현 언어에서는 다음과 같은 형태로 접근해서 사용할 수 있다.

내장 객체	설명
pageScope	page 객체에 저장된 속성의 <속성, 값> 매핑을 저장한 객체
requestScope	request 객체에 저장된 속성의 <속성, 값> 매핑을 저장한 객체
sessionScope	session 객체에 저장된 속성의 <속성, 값> 매핑을 저장한 객체
applicationScope	application 객체에 저장된 속성의 <속성, 값> 매핑을 저장한 객체

- ❖ JSP와 EL의 내장 객체 비교

어트리뷰트	JSP 내장 객체	표현 언어의 내장 객체
page 어트리뷰트	pageContext	pageScope
request 어트리뷰트	request	requestScope
session 어트리뷰트	session	sessionScope
application 어트리뷰트	application	applicationScope

실습 - EL 방식의 내장객체

<파일 이름: **elScope.jsp**>

```
<%  
pageContext.setAttribute("name", "page man");  
request.setAttribute("name", "request man");  
session.setAttribute("name", "session man");  
application.setAttribute("name", "application man");  
%>  
name : ${name} <p>  
page 속성 : ${pageScope.name} <p>  
request 속성 : ${requestScope.name} <p>  
session 속성 : ${sessionScope.name} <p>  
application 속성 : ${applicationScope.name} <p>  
<a href="scopeTest.jsp">이동</a>
```

실습 - EL 방식의 내장객체

<파일 이름: scopeTest.jsp>

name : \${name} <p>

page 속성 : \${pageScope.name} <p>

request 속성 : \${requestScope.name} <p>

session 속성 : \${sessionScope.name} <p>

application 속성 : \${applicationScope.name} <p>