**[ 8 ] EL(Expression Language)** ; 표현식을 대신해서 값을 표현

el표기법, 내장객체, getter를 대신하는 el

1. EL (Expression Language)란?
   * EL(Expression Language)란, 표현식(<%=~ %>) 를 조금 더 간략하게 대신해서 값을 표현하는 언어. 액션 태그를 조금 다르게(간략하게) 표현한 것.
   * 표현식 (<%= value %> ) → EL 표기법(${ value })

<body> 코드를 보면 물론 HTML은 다르다

<h1>${10 }</h1>

<h1>${99.99}</h1>

<h1>${"ABC" }</h1>

<h1>${**true**}</h1>

</body>

* + EL 연산자 (EL 표기법 안에서 아래의 연산자를 사용할 수 있다)
    1. 산술 : + - \* / %
    2. 관계 : ==(eq) !=(ne) <(lt) >(gt) <= >=
    3. 조건: a ? b : c
    4. 논리 && ||
    5. ${ empty 값} 공백이거나 null일 때 true
    6. ${not empty 값} 공백이거나 null이 아닐 때 ture

<body>

<h1>표현식 태그 </h1>

산술연산 : 10+1 = <%=10+1 %> <br>

산술연산 : 10 / 0.0 = <%=10/0.0 %><br>

관계연산 : 1 > 2 = <%= 1>2 %> <br>

관계연산 : 1 &lt; 2 = <%= 1<2 %> <br>

관계연산 : 1 == 2 = <%= 1 == 2 %> <br>

논리연산 : (1>2) || (1&lt;2) = <%=(1>2) || (1<2) %><br>

삼항연산 : <%= ( 1 > 2) ? "1>2입니다" : "1>2가 아닙니다" %><br>

파라미터 : <%=request.getParameter("pageNum")!=**null**? "pageNum 넘어롬" : "pageNum 안넘어옴"%><br>

<hr>

<h1>EL 표기법</h1>

산술연산 : 10+1 = ${10+1 } <br>

산술연산 : 10/0 = ${10/0 } <br>

관계연산 : 1 > 2 = ${ 1 **gt** 2 }<br>

관계연산 : 1 > 2 = ${ 1 > 2 }<br>

관계연산 : 1 &lt; 2 = ${ 1 **lt** 2 }<br>

관계연산 : 1 &lt; 2 = ${ 1 < 2 }<br>

관계연산 : 1 == 2 = ${ 1 **eq** 2 } <br>

관계연산 : 1 == 2 = ${ 1 == 2 } <br>

논리연산 : (1>2) || (1&lt;2) = ${(1 **gt** 2) || (1 **lt** 2) }<br>

삼항연산 : ${( 1 > 2) ? "1>2입니다" : "1>2가 아닙니다" }<br>

파라미터 : ${**not empty** param.pageNum ? "pageNum 넘어옴" : "pageNum 안넘어옴"}<br>

</body>

1. 내장 객체
   * Scope에 관한 객체
     1. requestScope : request객체의 attribute를 참조하는 객체(rquest attribute의 집합) Map 타입
     2. sessionScope : session객체의 attribute를 참조하는 객체(session attribute의 집합) Map 타입
   * 그 외 유용한 객체
     1. **param** : 요청 파라미터를 참조하는 객제(웹 브라우저로부터 입력된 데이터의 집합) Map 타입
     2. paramValues : 요청 파라미터(배열)를 참조하는 객제(웹 브라우저로부터 입력된 데이터의 집합-똑 같은 이름의 데이터가 여럿일 때 사용) Map 타입
   * input.jsp

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

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

</head>

<body>

<form action=*"inputView.jsp"* method=*"get"*>

<p>이름 <input type=*"text"* name=*"name"*></p>

<p>아이디 <input type=*"text"* name=*"id"*></p>

<p>비밀번호 <input type=*"text"* name=*"pw"*></p>

<p><input type=*"submit"*></p>

</form>

<%

request.setAttribute("requestName","requestValue");

session.setAttribute("sessionName", "sessionValue");

// inputView.jsp에게 request객체 그대로 전달하며 위임

//RequestDispatcher dispatcher = request.getRequestDispatcher("inputView.jsp?name=홍길동&id=hong&pw=111");

//dispatcher.forward(request, response);

%>

<p>페이지객체 내 어트리뷰트 <%=request.getAttribute("requestName") %></p>

<p>페이지객체 내 어트리뷰트 <%=session.getAttribute("sessionName") %></p>

<hr>

<p>페이지객체 내 어트리뷰트 ${requestScope.requestName }</p>

<p>페이지객체 내 어트리뷰트 ${sessionScope.sessionName }</p>

<hr>

<p>페이지객체 내 어트리뷰트 ${requestName }</p>

<p>페이지객체 내 어트리뷰트 ${sessionName }</p>

</body>

</html>

*inputView.jsp*

<body>

<%

String name = request.getParameter("name");

String id = request.getParameter("id");

String pw = request.getParameter("pw");

%>

<h2>이름 : <%=name %></h2>

<h2>I D : <%=id %></h2>

<h2>P W : <%=pw %></h2>

<hr>

<%

<h2>이름 : ${param.name }</h2>

<h2>이름 : ${param.id }</h2>

<h2>이름 : ${param.pw }</h2>

<hr>

<h2>request객체내 어트리뷰트 :${request\_name }</h2>

<h2>세션객체 내 어트리뷰트 : ${session\_name }</h2>

</body>

**petInput.html**

<body>

<form action=*"petsResult.jsp"*>

ID : <input type=*"text"* name=*"id"*><br><br>

다음 중 회원님이 키우고 있는 애완동물을 선택하세요<br><br>

개 <input type=*"checkbox"* name=*"pets"* value=*"개"*>

고양이 <input type=*"checkbox"* name=*"pets"* value=*"고양이"*>

금붕어 <input type=*"checkbox"* name=*"pets"* value=*"금붕어"*><br><br>

<input type=*"submit"* value=*"확인"*><input type=*"reset"* value=*"취소"*>

</form>

</body>

petsResult.jsp

<body>

<%

String id = request.getParameter("id");

String[] pets = request.getParameterValues("pets");

out.print("ID : "+id+"<br>");

out.print("선택한 동물 : ");

**if**(pets!=**null**){

**for**(String p : pets){

out.println(p +" ");

}

}**else**{

out.println("키우는 동물이 없네요");

}

%>

<hr>

ID ${param.id }<br>

선택한 동물 : ${paramValues.pets[0] } ${paramValues.pets[1] } ${paramValues.pets[2] }

${**empty** paramValues.pets ? "키우는 동물이 없네요":"" }

</body>

1. Getter를 대신하는 EL
   * <jsp:getProperty name="member" property="name"/>을 ${ member.name }으로 표현 가능

**DTO(자바빈)**

**package** com.tj.ex;

**public** **class** MemberInfo {

**private** String name;

**private** String id;

**private** String pw;

**public** String getName() {**return** name;}

**public** **void** setName(String name) {**this**.name = name;}

**public** String getId() {**return** id;}

**public** **void** setId(String id) {**this**.id = id;}

**public** String getPw() {**return** pw;}

**public** **void** setPw(String pw) {**this**.pw = pw;}

}

<body>

<%

Member member = **new** Member("aaa","111","김길동");

request.setAttribute("member", member);

%>

<h1>회원정보(el표기법)</h1>

<h3>member : ${member }</h3>

<h3>ID : ${member.id }</h3>

<h3>PW : ${member.pw }</h3>

<h3>name : ${member.name }</h3>

</body>