

1 1. EL ?  
2 1)JSTL(JSP Standard Tag Library) 1.0 규약에 소개된 내용  
3 2)JSP 2.0 에 새롭게 추가된 기능  
4 3)ECMAScript(JavaScript) 와 XPath 의 개념을 이용해서 설계  
5 4)EL에서는 '.' 과 '[' ']' 는 동일하게 처리 --> `expr.a == expr["a"]`  
6 5)Null값에 대해 관대하고, 자동 형변환처리가능  
7  
8  
9 2. 제공하는 기능  
10 1)JSP의 네가지 기본객체(page, request, session, application)가 제공하는 scope의 속성 사용  
11 2)집합 객체에 대한 접근 방법 제공  
12 3)수치 연산, 관계 연산, 논리 연산자 제공  
13 4)Java Class Method 호출 기능 제공  
14 5)표현 언어만의 기본 객체 제공  
15  
16  
17 3. 표현 방법  
18 `-${expression}`  
19 `-${7} ==> 7 == <%= 7 %>`  
20 `-${10 + 20 + 30} ==> 60`  
21 `-${Scope변수.속성명}` 또는 `-${Scope변수["속성명"]}`  
22 `<% %>, <%= %>, <%! %>`에서는 사용할 수 없다.  
23 -왜냐하면 expression, scriptlet, declaration 대신 사용하려고 만들었기 때문.  
24 `-${name} --> <%=name %>`와 같다.  
25  
26  
27 4. [] 와 .연산자  
28 1)EL 표현식의 반환 값이 속성을 가진 객체라면 [] 나 . 연산자를 이용해서 속성에 접근가능  
29 2>[] 와 . 연산자의 동작은 비슷  
30 3>[] 는 일반적인 형태, . 는 좀 더 간략한 형태  
31  
32 4)객체의 속성에 접근하려면 다음 두 표현식 중 하나를 사용해야 한다.  
33 `-${object["propertyName"]}`  
34 `-${object.propertyName}`  
35  
36 5)사용예  
37 `-${header["host"]}` or `-${header.host}`  
38  
39 6)하지만, propertyName에 영문자, 숫자, \$나 '\_'가 아닌 특수 문자가 포함될 경우에는 `-${header.이름}` 사용못함  
40 -ex: `-${header["user-agent"]}` ok, but `-${header.user-agent}` not  
41  
42  
43 5. 평가 규칙  
44 1)EL 표현식은 왼쪽에서 오른쪽으로 평가된다.  
45 2)만일 `expr-a[expr-b]`의 평가순서는 아래와 같다.  
46 -`expr-a`를 평가해 `value-a`를 얻는다.  
47 -`value-a`가 null 이면 null을 반환  
48 -`expr-b`를 평가해 `value-b`를 얻는다.  
49 -`value-b`가 null 이면 null을 반환  
50 -`value-a`의 타입이 `java.util.Map` 이면 `value-b`가 Map의 키인지 확인한다.  
51 -만일 키가 맞다면 `value-a.get(value-b)`의 결과를 반환한다.  
52 -만일 키가 아니라면 null을 반환한다.  
53 -`value-a`의 타입이 `java.util.List` 이거나 배열이라면 다음 작업을 수행한다.  
54 a. `value-b`를 int 로 변환한다.  
55 b. 변환에 실패하면 예외를 던진다.  
56 c. `value-a.get(value-b)`가 `IndexOutOfBoundsException`을 던지거나 `Array.get(value-a, value-b)`  
57 가 `ArrayIndexOutOfBoundsException` 을 던지면 null 을 반환  
58 d. 예외가 발생하지 않을 경우 `value-a`가 `value-a.get(value-b)`를, 배열이라면 `Array.get(value-a, value-b)`를 반환  
59 -`value-a`가 Map, List, 배열이 아니라면 `value-a`는 Java Bean이어야 한다.  
60 -Java Bean이 맞다면 `value-b`를 String으로 변환한다.  
61 -`value-b`가 `value-a`의 읽기 가능한 속성이라면 속성의 getter 메소드를 호출해 결과를 반환한다.  
62 -getter method가 예외를 던진다면 이 표현식은 유효하지 않은 표현식이다.  
63 -`value-a`가 Java Bean자바빈이 아닐 경우에도 이 표현식은 유효하지 않다.  
64  
65 6. 기본(내장)객체

- 1)pageContext  
-JSP의 pageContext 기본객체와 동일
- 2)pageScope  
-pageContext에 저장된 <속성, 값> mapping의 Map 객체
- 3)requestScope  
-request에 저장된 <속성, 값> mapping의 Map 객체
- 4)sessionScope  
-session에 저장된 <속성, 값> mapping의 Map 객체
- 5)applicationScope  
-application에 저장된 <속성, 값> mapping의 Map 객체
- 6)param  
-요청 parameter의 <parameter이름, 값> mapping의 Map 객체.  
-값의 type은 String.  
-request.getParameter(이름)의 결과와 동일

- 7)paramValues  
-요청 parameter의 <parameter이름, 값 배열> mapping의 Map 객체.  
-값의 type은 String [].  
-request.getParameterValues(이름)의 결과와 동일

- 8)header  
-요청 정보의 <header 이름, 값> mapping의 Map 객체.  
-request.getHeader(이름)의 결과와 동일

- 9)headerValues  
-요청 정보의 <header 이름, 값 배열> mapping의 Map 객체.  
-request.getHeaders(이름)의 결과와 동일

- 10)Cookie  
-<cookie 이름, Cookie> mapping의 Map 객체.  
-request.getCookies()로 구한 Cookie 배열로부터 생성

- 11)initParam  
-초기화 parameter의 <이름, 값> mapping의 Map 객체.  
-application.getInitParameter(이름)의 결과와 동일

## 7. Lab

### 1)eldemo.jsp

```
<body>
<%--
    Value 1 : <%=request.getParameter("num1") %><br />
    Value 2 : <%=request.getParameter("num2") %><br />
--%>
    Value 1 : ${param.num1} <br />
    Value 2 : ${param['num2']} <br />
    ${param.num1} + ${param.num2} = ${param.num1 + param.num2}
<%--
    <%
        int num1 = Integer.parseInt(request.getParameter("num1"));
        int num2 = Integer.parseInt(request.getParameter("num2"));
    %>
    <%=num1%> + <%=num2%> = <%=num1 + num2 %>
--%>
</body>
```

### 2)eldemo1.jsp

```
<body>
<%--<%=request.getHeader("user-agent") --%>
User's Browser 정보 : ${header['user-agent']} <br />
--${header.User-Agent} ==> 잘못된 식
--헤더 이름에 영문자, 숫자, $나 '_'가 아닌 특수 문자가 포함될 경우에는 ${header.이름} 사용못함
```



```

200     <param-value>scott</param-value>
201 </context-param>
202 <context-param>
203     <param-name>DBPWD</param-name>
204     <param-value>tiger</param-value>
205 </context-param>
206

```

#### 8)eldemo3.jsp

```

207 <body>
208     <h1>초기 환경설정 정보 알아내기</h1>
209     DBUSER : ${initParam.DBUSER}<br />
210     DBPWD : ${initParam.DBPWD }
211 </body>
212

```

#### 9)eldemo4.jsp

```

213
214 <%
215     session.setAttribute("username", "한지민");
216 %>
217 <body>
218     세션변수의 값 : <%= (String)session.getAttribute("username") %><br />
219     세션변수의 값 : ${sessionScope.username }<br />
220     세션변수의 값 : ${sessionScope["username"]} <br />
221 </body>
222

```

### 7. Data Types

- 1) Boolean : true, false
- 2) Integer : 0 ~ 9의 정수, 음수표현을 위해 '-' 사용
- 3) Float : 0 ~ 9의 실수, 소수점('.') 사용, 지수형도 가능, 0.5E10, 3e10
- 4) String : ' ', " " 의 문자열, 'DOG', '고양이'
- 5) Null : null

### 8. Operator

- 1) 산술연산자 : +, -, \*, /(div), %(mod) --> \${"10" + 1}, \${"일" + 10}, \${null + 1}
- 2) 비교연산자 : ==(eq), !=(ne), <(lt), >(gt), <=(le), >=(ge)
- 3) 논리연산자 : &&(and), ||(or), !(not)
- 4) empty 연산자 : 값의 null 여부검사, empty <값> e.g. \${empty param["name"]}
- 5) 비교 선택 연산자 : <수식> ? <값1> : <값2>
- 6) 연산자 우선순위
  - [], .
  - ()
  - -(단항), not, !, empty
  - \*, / , div, %, mod
  - +, -
  - <, >, <=, >=, lt, gt, le, ge
  - ==, !=, eq, ne
  - &&, and
  - ||, or
  - ? :

### 9. Lab

#### 1)eldemo5.jsp

```

253 \${"10" + 1 } : ${"10" + 1 }<br />
254 \${5 < 10} : ${5 < 10 } <br />
255 \${5 lt 10} : ${5 lt 10}<br />
256

```

#### 2)eldemo6.jsp

```

257 <body>
258     X = ${param.num1}, Y = ${param.num2} <br><br>
259     X + Y = ${param.num1 + param.num2} <br>
260     X - Y = ${param.num1 - param.num2} <br>
261     X * Y = ${param.num1 * param.num2} <br>
262     X / Y = ${param.num1 / param.num2} <br>
263     X % Y = ${param.num1 % param.num2} <br><br>
264     X가 더 큼니까? ${param.num1 - param.num2 > 0} <br>
265

```

```

267     Y가 더 큼니까? ${param.num1 - param.num2 < 0} <br><br>
268     X와 Y가 모두 양수입니까? ${ (param.num1 > 0) && (param.num2 > 0)} <br><br>
269     X와 Y가 같습니까? ${param.num1 == param.num2 ? "예" : "아니오"} <br> <br>
270 </body>
271
272 3)eldemo7.jsp
273 <body>
274     ${param.num1}을 ${param.num2}로 나눈 몫은? ${param.num1 div param.num2} <br>
275     나머지는? ${param.num1 mod param.num2} <br><br>
276     둘 다 양수입니까? ${ (param.num1 > 0) and (param.num2 > 0)} <br>
277     둘 다 음수입니까? ${ (param.num1 < 0) and (param.num2 < 0)} <br>
278 </body>
279
280
281 10. 배열 Item 출력
282 -eldemo8.jsp
283 <%
284     String array[] = new String[3];
285     array[0] = "조성모";
286     array[1] = "설운도";
287     array[2] = "나훈아";
288     pageContext.setAttribute("WINNERS", array);
289 %>
290 <html>
291     <body>
292         <h3>명단</h3>
293         1등. ${WINNERS[0]} <br>
294         2등. ${WINNERS[1]} <br>
295         3등. ${WINNERS[2]} <br>
296     </body>
297 </html>
298
299
300 11. List 계열 item 출력
301 -eldemo9.jsp
302 <jsp:directive.page import="java.util.ArrayList" />
303 <%
304     ArrayList<String> items = new ArrayList<String>();
305     items.add("딸기");
306     items.add("오렌지");
307     items.add("복숭아");
308     pageContext.setAttribute("FRUITS", items);
309 %>
310 <body>
311     <h3>이달에 가장 많이 팔린 과일입니다.</h3>
312     1위. ${FRUITS[0]} <br>
313     2위. ${FRUITS[1]} <br>
314     3위. ${FRUITS[2]} <br>
315 </body>
316
317
318 12. Map 계열 item 출력
319 -eldemo10.jsp
320 <jsp:directive.page import="java.util.HashMap" />
321 <%
322     HashMap<String, String> map = new HashMap<String, String>();
323     map.put("Edgar", "보스턴");
324     map.put("Thomas", "오하이오");
325     map.put("John", "워싱턴");
326     pageContext.setAttribute("ADDRESS", map);
327 %>
328 <body>
329     ${param.NAME}의 주소는? ${ADDRESS[param.NAME]}
330 </body>
331 http://localhost... /eldemo10.jsp?NAME=John
332
333

```

### 334 13. JavaBean 호출하기

#### 335 1)형식

336 `${beanName["propertyName"]} or ${beanName.propertyName}`

337

#### 338 2)Product.java

339 `package com.example.libs;`

340

341 `public class Product {`

342  `private String name;`

343  `private int price;`

344  `public String getName() {`

345  `return name;`

346  `}`

347  `public void setName(String name) {`

348  `this.name = name;`

349  `}`

350  `public int getPrice() {`

351  `return price;`

352  `}`

353  `public void setPrice(int price) {`

354  `this.price = price;`

355  `}`

356 `}`

357

#### 358 3)eldemo11.jsp

359 `<jsp:directive.page import="com.example.libs.Product" />`

360 `<jsp:useBean id="info" class="com.example.libs.Product" />`

361 `<jsp:setProperty name="info" property="name" value="Pencil" />`

362 `<jsp:setProperty name="info" property="price" value="10000" />`

363 `<%`

364  `session.setAttribute("PRODUCT", info);`

365 `%>`

366

367 `<body>`

368  `<h2>상품정보</h2>`

369  `이름 : ${sessionScope.PRODUCT.name} <br />`

370  `가격 : ${sessionScope["PRODUCT"]["price"]} </body>`

371 `</body>`

372

373

### 374 14.EL에서 class의 static method 호출하는 방법

#### 375 1)사용할 class 작성한다.

376 `package com.example.libs;`

377

378 `import java.util.Date;`

379 `import java.text.SimpleDateFormat;`

380

381 `public class DateEL {`

382  `private static SimpleDateFormat formatter = new SimpleDateFormat("yyyy-MM-dd");`

383  `public static String getFormat(Date date){`

384  `return formatter.format(date);`

385  `}`

386 `}`

387

#### 388 2)함수를 정의할 TLD file을 작성한다.

389 `<taglib xmlns="http://java.sun.com/xml/ns/javaee" version="2.1">`

390  `<description>EL에서 함수 실행</description>`

391  `<tlib-version>1.0</tlib-version>`

392  `<short-name>elfunctions</short-name>`

393

394  `<function>`

395  `<description>Date 객체 포매팅</description>`

396  `<name>dateFormat</name>`

397  `<function-class>com.example.libs.DateEL</function-class>`

398  `<function-signature>java.lang.String getFormat(java.util.Date)</function-signature>`

399  `</function>`

400 `</taglib>`

3)JSP page에서 함수 사용하기

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib prefix="eltest" uri="/WEB-INF/tlds/el-functions.tld" %>
<%
    java.util.Date today = new java.util.Date();
    request.setAttribute("today", today);
%>
<html>
<body>
    오늘은 <strong>${eltest:dateFormat(today)}</strong> 입니다.
</body>
</html>
```

15. EL 비 활성화 방법

1)web.xml 파일에 비활성화 옵션 지정하기

```
<jsp-config>
    <jsp-property-group>
        <url-pattern>*.jsp</url-pattern>
        <el-ignored>true</el-ignored>
    </jsp-property-group>
</jsp-config>
```

2)<%@ page isELIgnored="true" %>