

1. EL ?
 - 1) JSTL(JSP Standard Tag Library) 1.0 규약에 소개된 내용
 - 2) JSP 2.0 에 새롭게 추가된 기능
 - 3) ECMAScript(JavaScript) 와 XPath 의 개념을 이용해서 설계
 - 4) EL에서는 '.' 과 '['] 는 동일하게 처리 --> `expr.a == expr["a"]`
 - 5) Null값에 대해 관대하고, 자동 형변환처리가 가능
2. 제공하는 기능
 - 1) JSP 의 네가지 기본객체(`page`, `request`, `session`, `application`)가 제공하는 `scope` 의 속성 사용
 - 2) 집합 객체에 대한 접근 방법 제공
 - 3) 수치 연산, 관계 연산, 논리 연산자 제공
 - 4) Java Class Method 호출 기능 제공
 - 5) 표현 언어만의 기본 객체 제공
3. 표현 방법


```

      ${expression}
      ${7} ==> 7 ==> <%= 7 %>
      ${10 + 20 + 30} ==> 60
      ${ 스코프변수.속성명 } 또는 ${스코프변수["속성명"]}
      <%= %>, <%= %>, <%! %>에서는 사용할 수 없다.
      왜냐하면 expression, scriptlet, declaration 대신 사용하려고 만들었기 때문.
      ${name} --> <%=name %>와 같다.
      
```
4. [] 와 .연산자
 - 1) EL 표현식의 반환 값이 속성을 가진 객체라면 [] 나 . 연산자를 이용해서 속성에 접근가능
 - 2) [] 와 . 연산자의 동작은 비슷
 - 3) [] 는 일반적인 형태, . 는 좀 더 간략한 형태
 - 4) 객체의 속성에 접근하려면 다음 두 표현식 중 하나를 사용해야 한다.
 - `${object["propertyName"]}`
 - `${object.propertyName}`
 - 5) 사용예
 - `${header["host"]}` or `${header.host}`
 - 6) 하지만, propertyName에 영문자, 숫자, \$나 '_'가 아닌 특수 문자가 포함될 경우에는 `${header.이름}` 사용못함
 - ex: `${header["user-agent"]}` ok, but `${header.user-agent}` not
5. 평가 규칙
 - 1) EL 표현식은 왼쪽에서 오른쪽으로 평가된다.
 - 2) 만일 `expr-a[expr-b]` 의 평가순서는 아래와 같다.
 - `expr-a`를 평가해 `value-a` 를 얻는다.
 - `value-a`가 null 이면 null 을 반환
 - `expr-b`를 평가해 `value-b` 를 얻는다.
 - `value-b`가 null 이면 null을 반환
 - `value-a` 의 타입이 `java.util.Map` 이면 `value-b`가 Map 의 키인지 확인한다.
 - 만일 키가 맞다면 `value-a.get(value-b)`의 결과를 반환한다.
 - 만일 키가 아니라면 null 을 반환한다.
 - `value-a`의 타입이 `java.util.List` 이거나 배열이라면 다음 작업을 수행한다.
 - a . `value-b` 를 int 로 변환한다. 변환에 실패하면 예외를 던진다.
 - b. `value-a.get(value-b)`가 `IndexOutOfBoundsException`을 던지거나 `Array.get(value-a, value-b)` 가 `ArrayIndexOutOfBoundsException` 을 던지면 null 을 반환
 - c. 예외가 발생하지 않을 경우 `value-a`가 `value-a.get(value-b)`를, 배열이라면 `Array.get(value-a, value-b)`를 반환
 - `value-a`가 Map, List, 배열이 아니라면 `value-a`는 자바빈이어야 한다.
 - 자바빈이 맞다면 `value-b` 를 String 으로 변환한다. `value-b`가 `value-a`의 읽기 가능한 속성이라면 속성의 getter 메소드를 호출해 결과를 반환한다. getter메소드가 예외를 던진다면 이 표현식은 유효하지 않은 표현식이다. `value-a` 가 자바빈이 아닐 경우 예도 이 표현식은 유효하지 않다.
6. 기본(내장)객체
 - 1) `pageContext` : JSP 의 `pageContext` 기본객체와 동일
 - 2) `pageScope` : `pageContext` 에 저장된 <속성, 값> 매핑의 Map 객체

```

62 3) requestScope : request 에 저장된 <속성, 값> 매핑의 Map 객체
63 4) sessionScope : session 에 저장된 <속성, 값> 매핑의 Map 객체
64 5) applicationScope : application 에 저장된 <속성, 값> 매핑의 Map 객체
65 6) param : 요청 파라미터의 <파라미터이름, 값> 매핑의 Map 객체. 값의 타입은 String.
66     request.getParameter(이름)의 결과와 동일
67 7) paramValues : 요청 파라미터의 <파라미터이름, 값 배열> 매핑의 Map 객체. 값의 타입은 String [].
68     request.getParameterValues(이름)의 결과와 동일
69 8) header : 요청 정보의 <헤더 이름, 값> 매핑의 Map 객체. request.getHeader(이름)의 결과와 동일
70 9) headerValues : 요청 정보의 <헤더 이름, 값 배열> 매핑의 Map 객체.
71     request.getHeaders(이름)의 결과와 동일
72 10) Cookie : <쿠키 이름, Cookie> 매핑의 Map 객체. request.getCookies() 로 구한 Cookie 배열로부터 생
    성
73 11) initParam : 초기화 파라미터의 <이름, 값> 매핑의 Map 객체.
74     application.getInitParameter(이름)의 결과와 동일
75
76 //////////////////////////////////////////////////eldemo.jsp////////////////////////////////////
77 <body>
78 <%--
79     Value 1 : <%=request.getParameter("num1") %><br />
80     Value 2 : <%=request.getParameter("num2") %><br />
81 --%>
82     Value 1 : ${param.num1} <br />
83     Value 2 : ${param['num2']} <br />
84     ${param.num1} + ${param.num2} = ${param.num1 + param.num2}
85 <%--
86     <%
87         int num1 = Integer.parseInt(request.getParameter("num1"));
88         int num2 = Integer.parseInt(request.getParameter("num2"));
89         %>
90         <%=num1%> + <%=num2%> = <%=num1 + num2 %>
91     --%>
92 </body>
93 //////////////////////////////////////////////////eldemo1.jsp////////////////////////////////////
94 <body>
95 <%--<%=request.getHeader("user-agent") --%>
96 User's Browser 정보 : ${header['user-agent']} <br />
97 --${header.User-Agent} ==> 잘못된 식
98 --헤더 이름에 영문자, 숫자, $나 '_'가 아닌 특수 문자가 포함될 경우에는 ${header.이름} 사용못함
99 Ueer's Host : ${header.Host} <br /> == ${header["Host"]}
100 Accept[0] : ${headerValues.Accept[0]}<br />
101
102 요청 메소드 : ${pageContext.request.method} <br />
103 사용자의 IP : ${pageContext.request.remoteAddr} <br />
104 사용자의 IP : ${pageContext["request"].remoteAddr} <br />
105 사용자의 IP : ${pageContext["request"]["remoteAddr"]} <br />
106 파라미터 정보 : ${empty param.irum ? "unknown" : param.irum} <br />
107 Cookie 정보 : ${cookie["JSESSIONID"].value}
108 </body>
109 //////////////////////////////////////////////////cookieset.jsp////////////////////////////////////
110 <%
111     Cookie cookie = new Cookie("irum", "Sally");
112     response.addCookie(cookie);
113     Cookie cookie1 = new Cookie("gender", "male");
114     response.addCookie(cookie1);
115     Cookie cookie2 = new Cookie("age", "50");
116     response.addCookie(cookie2);
117 %>
118 <html>
119     <head><title>쿠키 데이터 저장 프로그램</title></head>
120     <body>
121         쿠키 값이 설정되었습니다.

```



```

183 1) Boolean : true, false
184 2) Integer : 0 ~ 9의 정수, 음수표현을 위해 '-' 사용
185 3) Float : 0 ~ 9의 실수, 소수점('.') 사용, 지수형도 가능, 0.5E10, 3e10
186 4) String : ' ', " " 의 문자열, 'DOG', '고양이'
187 5) Null : null
188
189 8. Operator
190 1) 산술연산자 : +, -, *, /(div), %(mod) --> ${"10" + 1}, ${"일" + 10}, ${null + 1}
191 2) 비교연산자 : ==(eq), !=(ne), <(lt), >(gt), <=(le), >=(ge)
192 3) 논리연산자 : &&(and), ||(or), !(not)
193 4) empty 연산자 : 값의 널 여부검사, empty <값> e.g. ${empty param["name"]}
194 5) 비교 선택 연산자 : <수식> ? <값1> : <값2>
195 5) 연산자 우선순위
196     - [], .
197     - ()
198     - -(단항), not, !, empty
199     - *, /, div, %, mod
200     - +, -
201     - <, >, <=, >=, lt, gt, le, ge
202     - ==, !=, eq, ne
203     - &&, and
204     - ||, or
205     - ? :
206
207 //////////////////////////////////eldemo5.jsp////////////////////////////////////
208 \${"10" + 1 } : ${"10" + 1 }<br />
209 \${5 < 10} : ${5 < 10 }<br />
210 \${5 lt 10} : ${5 lt 10}<br />
211 //////////////////////////////////eldemo6.jsp////////////////////////////////////
212 <body>
213     X = ${param.num1}, Y = ${param.num2} <br><br>
214     X + Y = ${param.num1 + param.num2} <br>
215     X - Y = ${param.num1 - param.num2} <br>
216     X * Y = ${param.num1 * param.num2} <br>
217     X / Y = ${param.num1 / param.num2} <br>
218     X % Y = ${param.num1 % param.num2} <br><br>
219     X가 더 큼니까? ${param.num1 - param.num2 > 0} <br>
220     Y가 더 큼니까? ${param.num1 - param.num2 < 0} <br><br>
221     X와 Y가 모두 양수입니까? ${param.num1 > 0} && (param.num2 > 0)} <br><br>
222     X와 Y가 같습니까? ${param.num1 == param.num2 ? "예" : "아니오"} <br> <br>
223 </body>
224 //////////////////////////////////eldemo7.jsp////////////////////////////////////
225 <body>
226     ${param.num1}을 ${param.num2}로 나눈 몫은? ${param.num1 div param.num2} <br>
227     나머지는? ${param.num1 mod param.num2} <br><br>
228     둘 다 양수입니까? ${param.num1 > 0} and (param.num2 > 0)} <br>
229     둘 다 음수입니까? ${param.num1 < 0} and (param.num2 < 0)} <br>
230 </body>
231
232 9. 배열 아이템 출력
233 //////////////////////////////////eldemo8.jsp////////////////////////////////////
234 <%
235     String array[] = new String[3];
236     array[0] = "조성모";
237     array[1] = "설운도";
238     array[2] = "나훈아";
239     pageContext.setAttribute("WINNERS", array);
240 %>
241 <html>
242     <body>
243         <h3>명단</h3>

```

```

244     1등. ${WINNERS[0]} <br>
245     2등. ${WINNERS[1]} <br>
246     3등. ${WINNERS[2]} <br>
247 </body>
248 </html>
249
250 10. List 계열 아이템 출력
251 //////////////////////////////////////////////////eldemo9.jsp////////////////////////////////////
252 <jsp:directive.page import="java.util.ArrayList" />
253 <%
254     ArrayList<String> items = new ArrayList<String>();
255     items.add("딸기");
256     items.add("오렌지");
257     items.add("복숭아");
258     pageContext.setAttribute("FRUITS", items);
259 %>
260 <body>
261     <h3>이달에 가장 많이 팔린 과일입니다.</h3>
262     1위. ${FRUITS[0]} <br>
263     2위. ${FRUITS[1]} <br>
264     3위. ${FRUITS[2]} <br>
265 </body>
266
267 11. Map 계열 아이템 출력
268 //////////////////////////////////////////////////eldemo10.jsp////////////////////////////////////
269 <jsp:directive.page import="java.util.HashMap" />
270 <%
271     HashMap<String, String> map = new HashMap<String, String>();
272     map.put("Edgar", "보스턴");
273     map.put("Thomas", "오하이오");
274     map.put("John", "워싱턴");
275     pageContext.setAttribute("ADDRESS", map);
276 %>
277 <body>
278     ${param.NAME}의 주소는? ${ADDRESS[param.NAME]}
279 </body>
280 http://localhost.../eldemo10.jsp?NAME=John
281
282 12. JavaBean 호출하기
283     ${beanName["propertyName"]} or ${beanName.propertyName}
284 //////////////////////////////////////////////////Product.java////////////////////////////////////
285 package com.example.libs;
286
287 public class Product {
288     private String name;
289     private int price;
290     public String getName() {
291         return name;
292     }
293     public void setName(String name) {
294         this.name = name;
295     }
296     public int getPrice() {
297         return price;
298     }
299     public void setPrice(int price) {
300         this.price = price;
301     }
302 }
303 //////////////////////////////////////////////////eldemo11.jsp////////////////////////////////////
304 <jsp:directive.page import="com.example.libs.Product" />

```

```

305 <jsp:useBean id="info" class="com.example.libs.Product" />
306 <jsp:setProperty name="info" property="name" value="Pencil" />
307 <jsp:setProperty name="info" property="price" value="10000" />
308 <%
309     session.setAttribute("PRODUCT", info);
310 %>
311
312 <body>
313     <h2>상품정보</h2>
314     이름 : ${sessionScope.PRODUCT.name} <br />
315     가격 : ${sessionScope["PRODUCT"]["price"]} }
316 </body>
317

```

11.EL에서 클래스 static 메소드 호출하는 방법

1) 사용할 클래스 작성한다.

```

320 *****
321 *****

```

```

321 package com.example.libs;
322

```

```

323 import java.util.Date;
324 import java.text.SimpleDateFormat;
325

```

```

326 public class DateEL {
327     private static SimpleDateFormat formatter = new SimpleDateFormat("yyyy-MM-dd");
328     public static String getFormat(Date date){
329         return formatter.format(date);
330     }
331 }
332

```

```

332 *****
333 *****

```

2) 함수를 정의할 TLD 파일을 작성한다.

```

334 *****
335 *****

```

```

335 <taglib xmlns="http://java.sun.com/xml/ns/javaee" version="2.1">
336 <description>EL에서 함수 실행</description>
337 <tlib-version>1.0</tlib-version>
338 <short-name>elfunctions</short-name>
339
340 <function>
341     <description>Date 객체 포매팅</description>
342     <name>dateFormat</name>
343     <function-class>com.example.libs.DateEL</function-class>
344     <function-signature>java.lang.String getFormat(java.util.Date)</function-signature>
345 </function>
346 </taglib>
347

```

```

347 *****
348 *****

```

3) JSP 페이지에서 함수 사용하기

```

349 *****
350 *****

```

```

350 <%@ page language="java" contentType="text/html; charset=UTF-8"
351     pageEncoding="UTF-8"%>
352 <%@ taglib prefix="eltest" uri="/WEB-INF/tlds/el-functions.tld" %>
353 <%
354     java.util.Date today = new java.util.Date();
355     request.setAttribute("today", today);
356 %>
357 <html>
358 <body>
359     오늘은 <b>${eltest:dateFormat(today)}</b> 입니다.
360 </body>

```

```
361 </html>
362
363 12. EL 비 활성화 방법
364 1) web.xml 파일에 비활성화 옵션 지정하기
365 <jsp-config>
366     <jsp-property-group>
367         <url-pattern>*.jsp</url-pattern>
368         <el-ignored>true</el-ignored>
369     </jsp-property-group>
370 </jsp-config>
371
372 2) <%@ page isELIgnored="true" %>
```