Expression Language (EL)

Expression Language

- 표현 언어 (Expression Language)
  - 웹 페이지에 값을 표현(Expression)하는데 사용하는 태그
  - JSP의 출력 문법을 보완하는 역할
  - ${} 를 사용
  - 표현 언어는 단순 값의 출력은 물론 식을 계산하여 출력할 수 있음
  - null은 공백으로 출력

Expression Language

- 데이터 타입
  - 정수형, 실수형, 문자열형, 논리형, null

- [실습] /ELJSTL/01
  1. 정수형 값의 출력을 확인해 봅니다.
  2. 실수형 데이터를 출력해 봅시다.
  3. 몬자열형 데이터를 출력해 봅시다.
  4. Boolean 형 데이터를 출력해 봅시다.
  5. null 값은 어떻게 출력되는지 확인해 봅시다.

Expression Language

- 연산자

종류	연산자
산술연산	+, -, *, /(div), %(mod)
관계연산	== (eq), != (ne), < (lt), > (gt), <= (le), >= (ge)
논리연산	&& (and), || (or), ! (not)
삼항연산	a ? b : c
null 검사	empty

- [실습2] /ELJSTL/02
  1. 산술 연산을 해 봅니다
  2. 관계연산을 해 봅니다
  3. 논리 연산을 해 봅니다
  4. null 검사를 해 봅니다

Expression Language

- EL vs Scriptlet

```jsp
<!-- Scriptlet -->
<%
    if (null == session.getAttribute("authUser")) {
%>
    <p>로그인을 하지 않았습니다.
<%
    }
%>
```

```jsp
- EL -->
<c:if test="${ empty authUser }">
<p>로그인을 하지 않았습니다.</p>
</c:if>
```

- EL은 Scriptlet에 비해 가독성이 좋다
- EL은 JSTL과 함께 사용할 수 있음
- 코드가 단순해지고 자바 코드를 JSP 내에 사용하지 않아도 됨

Expression Language

- EL을 이용 요청 파라미터 처리

```
<!-- Scriptlet -->
<%= request.getParameter("result") %>
```

```
<!-- EL -->
${ param.result }
```

- [실습 03]
  - /ELJSTL/03: EL로 요청 파라미터를 연습합니다.

Expression Language

- EL을 이용 Bean(객체) 접근하기

```
<!-- Scriptlet -->
<%
    request.setAttribute("memberInfo", memberVo);
%>
```

```
<!-- EL -->
${ requestScope.memberInfo.name }
$( requestScope.memberInfo.email )
```

- [실습 04]
  - /ELJSTL/04: Servlet에서 request 범위에 담은 Bean 객체를 JSP에서 EL로 접근해 보는 연습을 합니다.

Expression Language

- EL을 이용한 내장 객체 접근

내장 객체	범위
pageScope	페이지 범위의 Bean에 접근
requestScope	요청(Request) 범위의 Bean에 접근
sessionScope	세션(Session) 범위의 Bean에 접근
applicationScope	응용프로그램(Application) 범위의 Bean에 접근

- [실습 05]
  - MyHome 프로젝트: /includes/header.jsp에서 session 범위에 있는 authUser 객체를 EL로 접근해 봅니다.

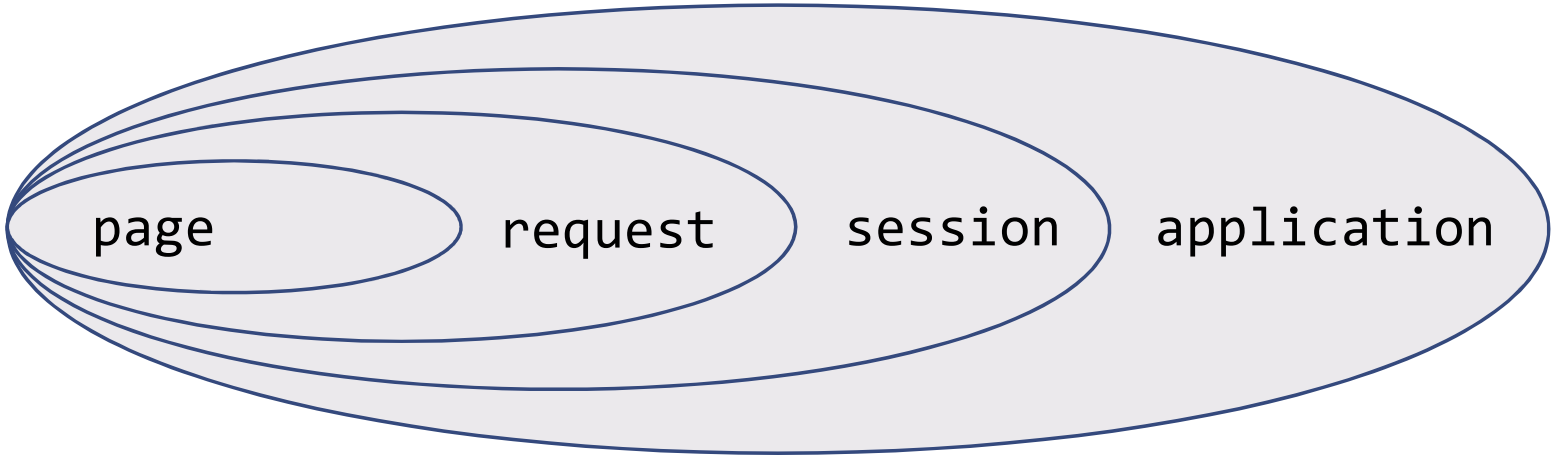
Expression Language

- EL을 이용한 내장 객체 접근
  - 스크립틀릿으로 내장 객체에 접근하고 저장된 빈(객체)에 접근하는 크드는 자바코드이고 길어지며 가독성이 매우 떨어진다
  - EL을 이용, 간결하게 표현한다
  - 내장 객체 접근시 내장 객체의 표현은 생략할 수 있다.

- [실습 06]
  - 이전 [실습 05]에 사용했던 내장 객체 표현을 생략하고 바로 객체 이름으로 표현해 봅니다.

Expression Language

- 내장 객체 생략시 동일 이름의 객체에 대한 우선 순위
  - pageScope -> requestScope -> sessionScope -> applicationScope



- [실습 07]
  - /ELJSTL/07 : 두 객체가 동일 이름으로 requestScope와 sessionScope에 있을 때 내장 객체를 생략시 접근 우선 순위를 확인해 봅니다

JSTL

JSP Standard Tag Library

JSTL
: JSP Standard Tag Library

- JSP에서 사용 가능한 표준 태그 라이브러리
- JSP 코드가 깔끔해지고 가독성이 좋아진다
- JSTL 라이브러리
  - 기본 기능 (core)
  - 형식화 (format)
  - 데이터베이스 (sql)
  - XML 처리 (xml)
  - 함수 처리 (function)

- [실습 08]
  - http://tomcat.apache.org/ 에서 태그 라이브러리를 다운로드 받아 프로젝트에 적용해 봅니다

JSTL
: JSP Standard Tag Library

- [실습 09]
  - 태그 라이브러리 지시자를 jsp 페이지 상단에 붙여 봅니다

  - 기본 기능 (core)

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
```

  - 형식화 (format)

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt"%>
```

  - 함수 처리 (function)

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn"%>
```

JSTL
: JSP Standard Tag Library

- JSTL을 사용하는 이유
  - [실습 10] /ELJSTL/color.jsp
    - color.jsp는 Scriptlet으로 작성되어 있습니다.
    - JSTL을 사용하여 colorJSTL.jsp로 변경해 보고 두 코드를 비교해 봅니다.

JSTL
: JSP Standard Tag Library

- <c:if>
  - Java의 if 문과 비슷한 기능 제공
  - else 처리문은 없음
  - test 속성에 지정한 조건을 평가

```
<c:if test="${ param.color  == 1 }">
    <span style="color:red;">빨강</span>
</c:if>
```

JSTL
: JSP Standard Tag Library

- <c:choose>
  - Java의 if ~ else if ~ else 문과 같은 기능 제공
  - 서브 태그로 <c:when), <c:otherwise> 가 있음

  - [실습 11]
    - [실습 10]에서 작성한 colorJSTL.jsp를 <c:choose>문을 이용하여 수정합니다.

  - [실습 12]
    - MyHome 프로젝트의 /includes/header.jsp를 <c:choose>로 처리해 봅니다

JSTL
: JSP Standard Tag Library

- <c:forEach>
  - 배열 또는 List, Map 등 컬렉션 객체에 저장되어 있는 요소들을 순차적으로 처리할 때 사용

```
<c:forEach items="${ list }" var="vo" varStatus="status">
    <h1>${ vo.name }</h1>
    <h3>${ vo.email }</h3>
</c:forEach>
```

  - [실습 13]
    - 앞서 작성한 emaillist 등 프로젝트에서 for 문을 <c:forEach> 로 대체해 봅니다.

  - [실습 14]
    - ${status.index}, ${status.count} 도 값을 확인해 봅니다.

JSTL
: JSP Standard Tag Library

- <c:forEach>
  - 컬렉션 객체 순회 기능 외에 특정 횟수를 지정하여 원하는 만큼 반복 가능

```
<c:forEach begin="시작 값" end="끝값" var="변수명" step="증가값">
...
</c:forEach>
```

```
<c:forEach begin="1" end="9" var="i" step="2">
    ${ i }
</c:forEach>
```

  - [실습 15]
    - table.jsp를 복사하여 EL&JSTL 구문으로 변경한 tableJSTL.jsp 를 작성해 봅니다.

JSTL
: JSP Standard Tag Library

- <c:set>
  - 변수에 값을 설정

```
<c:set var="count" value="10" scope="page" />
```

```
<c:set var="count">10</c:set>
```

  - 특정 Bean(객체)의 속성을 설정

```
<c:set target="${ memberInfo }" property="name" value="홍길동" />
```

JSTL
: JSP Standard Tag Library

- <c:import>
  - <jsp:include> 처럼 다른 페이지를 동적으로 포함시킬 때 사용할 수 있는 태그

```
<c:import url="url">
<c:param name="파라미터명" value="파라미터값" />
</c:import>
```

  - [실습 16]
    - MyHome에서 <jsp:include>를 <c:import>로 변경해 봅니다.