

chapter06

EL
JSTL

1. EL(Expression Language)
2. JSTL(JSP Standard Tag Library)

■표현언어(Expression Language)

- 웹 페이지에 값을 표시(표현),하는 데 사용되는 태그
- JSP의 출력 문법을 보완하는 역할
- `${}` 를 사용한다.
- 표현언어의 식을 계산해서 출력한다.
- `null` 는 공백으로 출력한다.

`${member.name}`

■ 데이터 타입

- 정수형, 실수형, 문자열형, 논리형, null

■ [실습1] eljstl/01 에서

1. 정수형 값의 출력을 확인해 봅니다.
2. 실수형 데이터를 출력해 보세요.
3. 문자열형 데이터를 출력해 봅니다.
4. boolean 형 데이터를 출력해 봅니다.
5. null 도 확인해 보세요

■ 연산자

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

■ [실습2] eljstl/02 에서

1. 산술연산을 해봅니다.
2. 관계연산을 해 봅니다.
3. 조건연산을 해 봅니다.
4. 논리연산을 해봅니다.
5. null 검사를 해봅니다.

■스크립틀릿 과 비교

[스크립틀릿]

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

[EL 표현식]

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

- 스크립틀릿에 비해 가독성이 좋다.
- JSTL과 함께 사용할 수 있으며 코드가 단순해지고 자바코드를 사용하지 않아도 된다.

■ 요청파라미터 처리하기

[스クリプト릿]

```
<%  
    String id= request.getParameter("id");  
%>  
id = <%=id %> <br/>
```

```
<%=request.getParameter("id") %>
```

[EL 표현식]

```
${param.id } <br/>
```

■ [실습3] eljstl/03 에서

EL로 요청파라미터를 출력해 봅니다.

■ EL로 객체 접근 하기 객체→bean

```
request.setAttribute("userVo", userVo);
```

```
${requestScope.userVo.name }  
${requestScope.userVo.email }
```

■ [실습4] eljstl/04

Servlet에서 request범위에 담은 객체(빈:bean)를 JSP에서 접근해 보는 연습을 합니다.

■EL로 내장객체 접근 하기

pageScope	페이지 범위에 빈을 저장
requestScope	요청 범위에서 빈을 저장
sessionScope	세션 범위에서 빈을 저장
applicationScope	어플리케이션 범위에서 빈을 저장

■[실습5]

mysite에서 /include/header.jsp 에서 session 범위에 있는 authUser 객체를 EL로 접근해 봅니다.

■EL로 내장객체 접근 하기

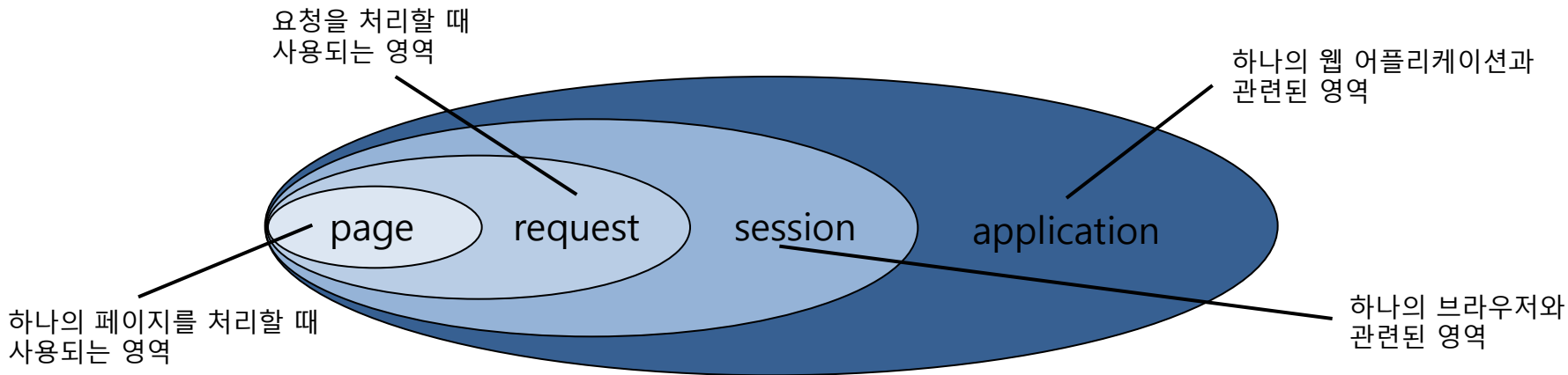
- 스크립트릿 로 내장객체에 접근하고 저장된 빈(객체)에 접근하는 코드는 자바코드로 길어지며 지저분해 보인다.
- EL로 간결하게 표현한다.
- 내장객체 접근시 내장객체의 표현은 생략할 수 있다.

■[실습6]

앞의 [실습05]에 사용했던 내장객체 표현을 생략하고 바로 객체이름으로 표현해 봅니다.

■ 내장객체 생략시 같은 이름의 객체에 대한 우선 순위

• pageScope -> **requestScope** -> **sessionScope** -> applicationScope



■ [실습7] eljstl/07 에서

두 객체가 같은 이름으로 requestScope, sessionScope 에 있을때 내장객체 생략시 접근 우선순위를 확인해 봅니다.

chapter06

EL
JSTL

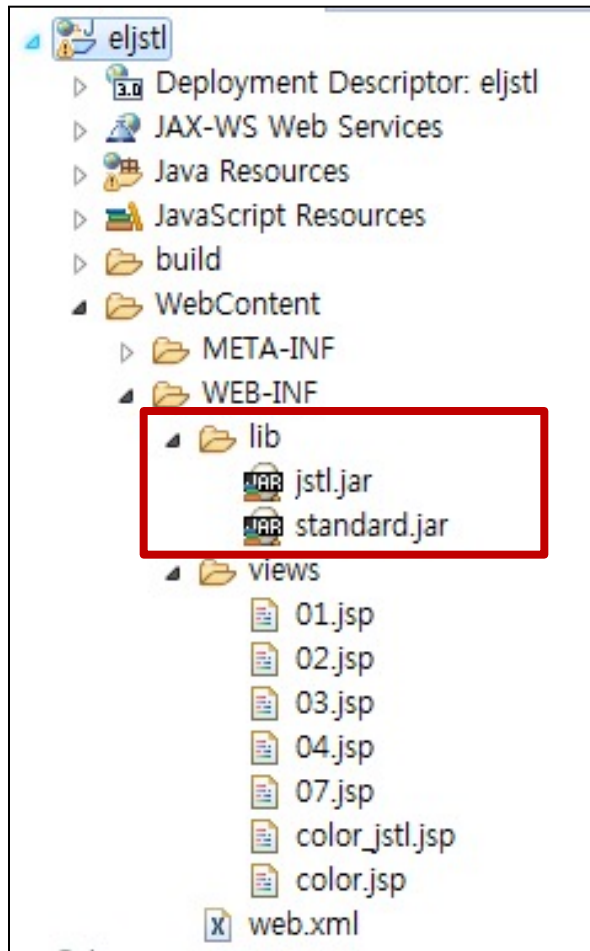
1. EL(Expression Language)
2. JSTL(JSP Standard Tag Library)

■ JSTL(JSP Standard Tag Library)

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

■ [실습8]

jar 파일 다운로드 받아 프로젝트에 적용시키기



■[실습9]

태그 라이브러리 지시자 붙이기

•기본기능 (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"%>
```

•함수처리(function) :

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

■ <c:if> * if ~ else 가 없다

```
<c:if test="조건식">
```

조건이 참일 경우 실행할 문장

```
</c:if>
```

```
<c:if test="\${param.color == 1}">
```

빨강

```
</c:if>
```

■[실습10]

color.jsp 는 스크립트릿으로 작성되어 있습니다.

JSTL 를 사용하여 10_color_jstl.jsp 로 변경해 보고 두 JSP 코드를 비교해 봅니다.

■ <c:choose> if~else, if~else 문과 유사

·

```
<c:choose>
```

```
    <c:when test="조건1"> 조건1일때 </c:when>
```

```
    <c:when test="조건2"> 조건2일때 </c:when>
```

```
    <c:otherwise> 조건에 만족하지 않을때 </c:otherwise>
```

```
</c:choose>
```

■ [실습11]

11_color_jstl.jsp 를 <c:choose> 문으로 수정합니다.

■ [실습12]

mysite 에서 /include/header.jsp 를 <c:choose>로 처리해 봅니다.

■ <c:foreach>

- 배열 또는 List, Map 등의 컬렉션 객체에 저장되어 있는 요소들을 순차적으로 처리할때 사용한다.

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

[실습13]

mysite, emaillist2, guestbook2 에서 for문을 변경해 봅니다.

[실습 14]

`${status.index }` , `${status.count }` 도 값을 확인해 봅니다.

■ <c:import>

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

```
<c:import url="/WEB-INF/views/includes/header.jsp"></c:import>
```

[실습15]

mysite에서 <jsp:include>를 <c:import> 로 변경해 봅니다.