

JSP

1. EL
2. JSTL
3. JSTL문법

* EL(Expression Language)

-EL이란 JSP에서 저장객체를 출력할때 **스크립팅을 전혀 쓰지 않을 수 있는 기술**입니다.

-EL은 일종의 스크립트 언어로 자료 타입, 수치 연산자, 논리 연산자, 비교 연산자 등을 제공하며 **표현식을 대체**할 수 있습니다.

- EL의 사용법

ex) 표현식 `<%= value %>` → **EL `${value}`**

- EL 내부에 사용하는 연산자

1. 산술: +, -, *, /, %
2. 관계: ==, !=, <, <=, >, >=
3. 조건: `a ? b : c` (a조건식이 참이면 b를 실행, 거짓이면 c를 실행)
4. 논리: `&&`, `||`

* 액션 태그와 EL

- 액션태그 `<jsp:getProperty name="member" property="name"/>`

- EL -> **`${member.name}`**

- EL의 내장객체

`${내장객체.name}`

내장 객체	의미
1. pageScope	JSP의 page 객체를 참조하는 객체
2. requestScope	JSP의 request 객체를 참조하는 객체
3. sessionScope	JSP의 session 객체를 참조하는 객체
4. applicationScope	JSP의 application 객체를 참조하는 객체
5. param	요청 파라미터를 참조하는 객체
6. paramValues	요청 파라미터(배열)를 참조하는 객체
7. initParam	서블릿컨텍스트 초기화 파라미터를 참조하는 객체
8. cookie	쿠키 객체를 참조하는 객체.

* JSTL (JSP Standard Tag Library)

- JSP의 경우 HTML태그와 같이 사용되어 전체적인 코드의 가독성이 떨어집니다.
- 그래서 이런 단점을 보완하고 만들어진 것이 JSTL입니다.
- JSTL을 사용하면 **자바의 제어문을 HTML 태그화** 시킬 수 있습니다.
- JSTL의 경우 우리가 사용하는 Tomcat 기본 컨테이너에 포함되어 있지 않으므로, 별도의 라이브러리를 설치하고 사용합니다.
- **설치 방법:**
<http://jakarta.apache.org> 접속
(jstl.jar, standard.jar 파일 설치 된 톰캣의 lib 파일 안으로 복사)

종류	URI	prefix
Core	http://java.sun.com/jsp/jstl/core	c
XML processing	http://java.sun.com/jsp/jstl/xml	x
Formatting	http://java.sun.com/jsp/jstl/fmt	fmt
Database access	http://java.sun.com/jsp/jstl/sql	sql
Functions	http://java.sun.com/jsp/jstl/functions	fn

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

태그	설 명
c:catch	예외처리에 사용
c:out	JspWriter 에 내용 출력
c:set	JSP 에서 사용될 변수 설정 <c:set value="value" target="targetObjectName" property="propertyName" />
c:remove	설정한 변수 제거
c:if	조건처리
c:choose	다중 조건 처리
c:forEach	컬렉션이나 Map 의 각 항목 처리
c:forTokens	구분자로 분리된 각각의 토큰을 처리할 때 사용
c:when	조건에 맞을 때
c:otherwise	맞는 조건이 없을 경우
c:import	URL 을 사용하여 다른 자원의 결과를 삽입
c:redirect	URL 을 재작성
c:param	파라미터 설정

* JSTL Core Library 사용 방법

- Core 라이브러리는 기본적인 라이브러리로 출력, 제어문, 반복문 같은 기능이 포함되어 있습니다.
- 코어 라이브러리를 사용하기 위해서 반드시 JSP 파일 내의 지시자 태그로

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

1. 출력 태그

```
<c:out value="출력값">
```

2. 변수 선언 태그

- 변수 태그는 JSTL이 지원하는 태그에서 사용할 수 있는 변수를 설정하기 위해 사용합니다. EL 변수를 생성합니다.

```
<c:set var="변수명" value="설정값" scope="범위">
```

- **var**: 값을 저장할 EL 변수의 이름을 지정함.
- **value**: 변수의 값을 지정함. 표현식, EL, 텍스트를 사용하여 지정가능.
- **scope**: 변수를 저장할 영역을 지정함. page, request, session, application 중 하나를 지정해야 하며 지정하지 않을 경우 page를 기본 값으로 사용.

3. 변수 제거 태그

- remove 태그를 사용할 때 주의점은 삭제할 변수의 scope를 지정하지 않으면 동일한 이름으로 저장된 모든 영역의 변수를 삭제합니다.

```
<c:remove var="변수명" scope="범위">
```

4. 조건문(if) 태그

- if 태그는 자바 언어의 if 블록과 비슷한 기능을 제공합니다. 중첩된 if~else 문과 같은 효과를 낼 수는 없지만 단순한 if 블록을 쉽게 대체할 수 있기 때문에 많이 사용합니다.

```
<c:if test="조건식" var="조건 처리 변수명">
```

5. 조건문(choose) 태그

- choose 태그는 자바의 switch 구문과 if~else 구문을 혼합한 형태로 다수의 조건문을 하나의 블록에서 수행할 때 사용합니다.

```
<c:choose>
```

```
    <c:when test="조건식"> 처리 내용 </c:when>
```

```
    <c:when test="조건식"> 처리 내용 </c:when>
```

```
    <c:otherwise> 처리 내용 </c:otherwise>
```

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

6. 반복문 태그

- `<c:forEach items="객체명" begin="시작값" end="끝 값" step="증감식" var="변수명">`
- forEach 태그는 배열, 컬렉션 또는 맵에 저장되어 있는 값들을 순차적으로 처리할 때 사용합니다.
- 배열의 경우는 객체의 배열뿐만 아니라 기본 데이터 타입의 배열도 알맞게 처리를 하며 기본 데이터 타입은 래퍼 클래스를 통해 처리합니다.