

# 삼성 청년 SW 아카데미

Web-BackEnd

# EL & JSTL

- EL (Expression Language)
- JSTL (JSP Standard Tag Library)

# EL (Expression Language)

## ✓ EL (Expression Language)

- 표현언어라고 하며 값(data)를 웹 페이지에 표시 하는데 사용되는 태그
- JSP 스크립트의 표현식(<%= %>)을 대신하여 속성값을 쉽게 출력할 수 있도록 고안된 언어

스크립트 표현식 : <%= expr %>      EL : \${expr }

- EL 표현식에서 Map 객체 또는 Java Bean 객체의 경우 도트(.) 연산자를 사용하여 속성을 출력할 수 있다.
- JSP 영역 기본 객체 (page, request, session, application) 속성 사용가능
- 수치, 관계, 논리연산 제공
- 자바 클래스 메서드 호출 기능 제공
- 표현언어 기본 객체 제공 (param, paramValues )

### ✓ EL (Expression Language) 연산자(Operator)

종류	사용 가능 연산자
산술	+, -, *, /(div) , %(mod)
관계	==(eq), !=(neq) , <(lt), >(gt), <=(le), >=(ge)
조건	expr ? val1 : val2
논리	&&(and),   (or), !(not)
null	empty

괄호 안은 동일 의미 연산자

- EL 식은 연산자를 포함할 수 있으며, 산술 연산, 조건 연산, 논리 연산, 관계 연산을 수행할 수 있다.
- 일반적인 연산 이외에 null 연산을 수행할 수 있다.
- empty 연산자는 검사할 객체가 null인지 아닌지 검사하기 위해서 사용된다.  
(null, "", 요소가 없는 list, map, collection은 True)

```
${ empty name }
```

## ✓ EL (Expression Language) 내장 객체

분류	객체	타입	설명
JSP	pageContext	Java Bean	현재 페이지의 page context instance
scope	pageScope	Map	page 기본객체에 저장된 속성을 저장하는 객체
	requestScope	Map	request 기본객체에 저장된 속성을 저장하는 객체
	sessionScope	Map	session 기본객체에 저장된 속성을 저장하는 객체
	applicationScope	Map	application 기본객체에 저장된 속성을 저장하는 객체
요청 파라미터	param	Map	JSP 내장 객체 request의 getParameter(name) 메서드와 동일한 역할을 한다. \${param.name} 또는 \${param["name"]} 의 형태로 사용
	paramValues	Map	JSP 내장 객체 request의 getParameterValues(name) 메서드와 동일한 역할을 한다.

## ✓ EL (Expression Language) 내장 객체

분류	객체	타입	설명
쿠키	cookie	Map	현재 페이지의 page context instance
요청헤더	header	Map	request의 getHeader(name) 메서드와 동일한 역할을 한다.
	headerValues	Map	request의 getHeaders(name) 메서드와 동일한 역할을 한다.

## ✓ EL로 기본 객체 속성 설정하기 / 요청 파라미터 처리하기

### ■ 예시 1) `request.setAttribute("userInfo", "ssafy");`

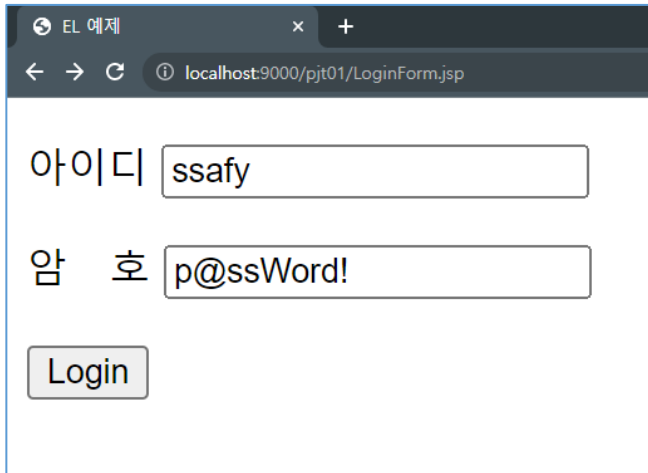
1. `${requestScope.userInfo}`
2. `${pageContext.request.userInfo}`
3. `${userInfo}`     property 이름만 사용 할 경우 pageScope > requestScope > sessionScope > applicationScope 순서대로 객체를 찾음

### ■ 예시 2) `url?name=ssafy&hobby=등산&hobby=수영`

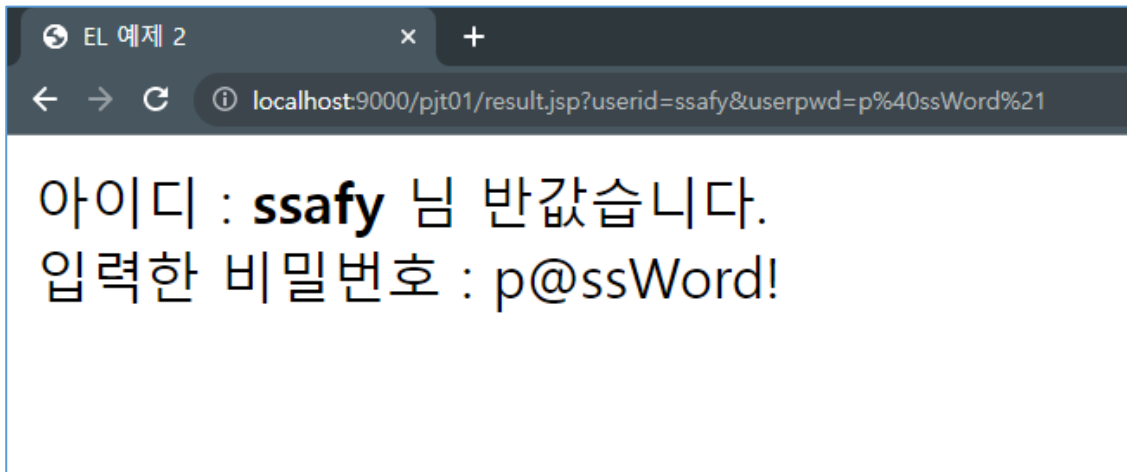
1. `${param.name}`
2. `${paramValues.hobby[0]}` , `${paramValues.hobby[1]}`



## ✓ 로그인 폼 작성하기



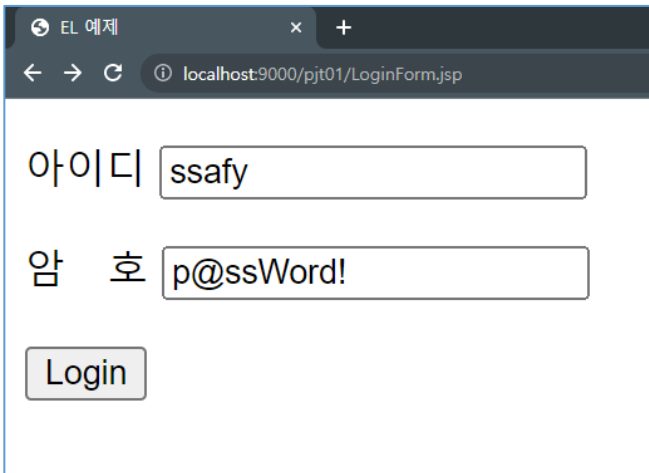
A screenshot of a web browser window showing a login form. The browser's address bar displays 'localhost:9000/pjt01/LoginForm.jsp'. The form contains two input fields: '아이디' (ID) with the value 'ssafy' and '암 호' (Password) with the value 'p@ssWord!'. Below the password field is a 'Login' button.



A screenshot of a web browser window showing the result of a successful login. The browser's address bar displays 'localhost:9000/pjt01/result.jsp?userid=ssafy&userpwd=p%40ssWord%21'. The page content displays the following text:

아이디 : **ssafy** 님 반갑습니다.  
입력한 비밀번호 : p@ssWord!

### ✓ 로그인 폼 작성하기(실습)

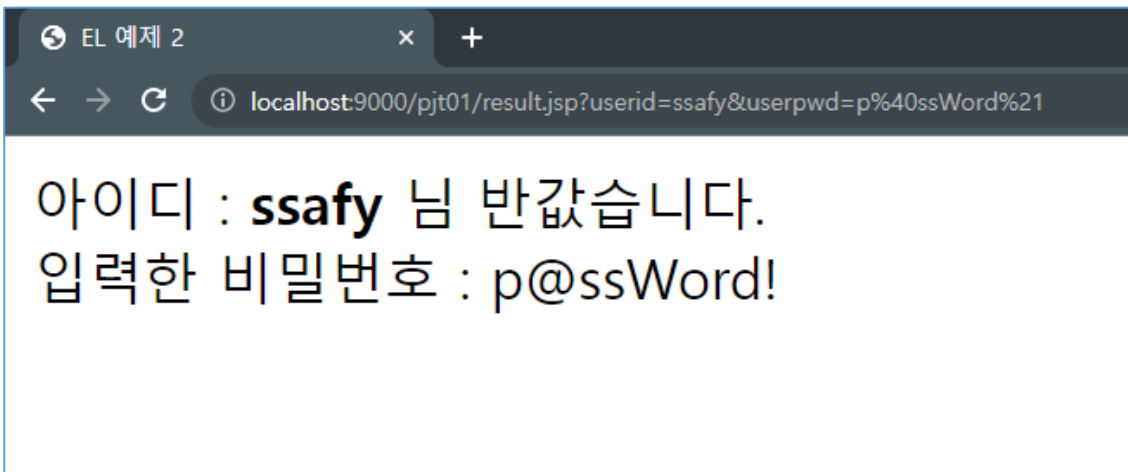


아이디 ssafy

암 호 p@ssWord!

Login

```
<form action="result.jsp" method="GET">
  <p> <label for="userid">아이디</label>
    <input type="text" name="userid" id="userid"></p>
  <p> <label for="userpwd">암 &nbsp;&nbsp; 호</label>
    <input type="text" name="userpwd" id="userpwd"></p>
  <input type="submit" value="Login">
</form>
```



아이디 : **ssafy** 님 반갑습니다.  
입력한 비밀번호 : p@ssWord!

```
<body>
  아이디 : <b>${param.userid}</b> 님 반갑습니다.
  <br>
  입력한 비밀번호 : ${param.userpwd}
</body>
```

# JSTL

## (JSP Standard Tag Library)

## ✓ JSTL (JSP Standard Tag Library)

- JavaEE 기반의 웹 어플리케이션 개발을 위한 컴포넌트 모음
- JSP 스크립트 와 html 코드가 섞여서 복잡한 구조를 가짐.  
이를 간결하게 작성하기 위해서 자바코드를 태그 형태로 작성해 놓은 것
- 유용한 커스텀 태그들을 모아서 표준화 한 것

## ✓ JSTL 기능

- 간단한 프로그램 로직 구현 기능 - 변수선언, if 문장, for문 등
- 데이터 출력 포맷 설정
- DB 입력, 수정, 삭제, 조회 기능
- 문자열 처리 함수
- XML 문서 처리

## ✓ JSTL 사용하기

✓ <https://mvnrepository.com/artifact/javax.servlet/jstl1.2>

← → ↻ [mvnrepository.com/artifact/javax.servlet/jstl/1.2](https://mvnrepository.com/artifact/javax.servlet/jstl/1.2)

**MVNREPOSITORY** Search for groups, artifacts, categories

Indexed Artifacts (26.0M)



Popular Categories

- Aspect Oriented
- Actor Frameworks
- Application Metrics
- Build Tools

Home » javax.servlet » jstl » 1.2

 **JSTL » 1.2**  
JSTL

License	<a href="#">CDDL</a> <a href="#">GPL 2.0</a>
Categories	<a href="#">Java Specifications</a>
Date	(Jun 23, 2011)
Files	<a href="#">pom (356 bytes)</a> <a href="#">jar (404 KB)</a> <a href="#">View All</a>
Repositories	<a href="#">Central</a> <a href="#">EBIPublic</a> <a href="#">OneBusAway Pub</a> <a href="#">Redhat GA</a> <a href="#">WSO2 Dist</a>
Used By	1,841 artifacts

- WebContent
  - META-INF
  - WEB-INF
    - lib
      - jstl-1.2.jar

② WEB-INF/lib 에 복사

① jstl-1.2.jar 다운로드

## ✓ JSTL 사용하기

- taglib 지시자를 이용한 태그 사용 선언

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

- 사용하고자 하는 기능에 따라 어떤 라이브러리를 사용할지 작성한다. (ex: core)
- 사용할 태그를 구분하기 위해서 prefix를 작성해준다.

```
<c:out value="Hello! JSTL"/>
```

- prefix에 작성한 접두사를 적어주고, 기능에 따른 태그를 선택하여 작성한다.

## ✓ JSTL 사용하기

기능	prefix	URI
기본기능, 변수선언, 로직 구현	c	<a href="http://java.sun.com/jsp/jstl/core">http://java.sun.com/jsp/jstl/core</a>
데이터 출력 포맷	fmt	<a href="http://java.sun.com/jsp/jstl/fmt">http://java.sun.com/jsp/jstl/fmt</a>
데이터베이스작업	sql	<a href="http://java.sun.com/jsp/jstl/sql">http://java.sun.com/jsp/jstl/sql</a>
XML 관련 기능	x	<a href="http://java.sun.com/jsp/jstl/xml">http://java.sun.com/jsp/jstl/xml</a>
문자열 처리 함수	fn	<a href="http://java.sun.com/jsp/jstl/fn">http://java.sun.com/jsp/jstl/fn</a>

## ✓ JSTL - Core

태그	설명
c:set	변수에 값 설정
c:remove	변수 제거
c:if	조건에 따른 코드 실행
c:choose	when, otherwise tag와 함께 쓰인다. if-else if - else와 동일하게 사용
c:forEach	배열, collection 의 각 요소에 접근하기 위해 사용
c:import	외부 자원을 사용하기 위해서 사용
c:forTokens	구분자를 이용하여 구분된 토큰을 처리
c:redirect	redirect시 사용
c:url	URL 작성
c:out	출력
c:catch	예외 처리



## ✓ c:set 태그

### 기본 형식

```
<c:set var="변수 이름" value="변수 값" scope="page/request/session/application"/>
```

속성	설명
var	변수이름 지정
value	변수 값 지정
scope	변수의 유효영역 지정, 작성하지 않으면 기본값 page
target	속성값을 지정할 java bean 객체

예시) request영역에 name 속성을 지정하고 'ssafy' 라는 문자열을 저장

```
<c:set var="name" value="ssafy" scope="request"/>
```

위 태그는 아래 자바 코드와 동일하다.

```
request.setAttribute("name", "ssafy")
```

## ✓ c:set 태그

java bean 객체에 속성값 지정

java bean 객체 생성 및 할당

```
<c:set var="user1" value="<%= new com.ssafy.model.User() %>" />
```

java bean 객체 이름

java bean 객체 속성값 지정

```
<c:set target="${user1}" property="name" value="ssafy"></c:set>
```

java bean 객체

java bean 속성 이름

할당 할 값

## ✓ c:if 태그

### 기본 형식

```
<c:if test="조건식">  
    조건이 참일 경우 실행할 문장  
</c:if>
```

‘menu’ 파라미터가 1이라면 한식 출력

```
<c:if test="${param.menu == 1 }">  
    <c:out value="한식"/>  
</c:if>
```

현재 사용자가 작성자라면, 삭제 링크 출력

```
<c:if test="${writerid == userid }">  
    <a href="url">삭제</a>  
</c:if>
```

## ✓ c:forEach 태그

- 컬렉션에 있는 항목들에 대하여 반복 수행
- begin, end, step 속성을 이용하여 반복할 항목을 지정할 수 있다.
- varStatus 속성을 이용해 요소의 순번, 개수 등을 출력할 수 있다.

### 기본 형식

```
<c:forEach [var="변수"] items="배열, Map, Collection 등">  
    동작코드  
</c:forEach>
```

## ✓ c:forEach 태그

### 영화 목록 출력

```
<%  
String[] movies = {"아이언맨", "쇼생크탈출", "겨울왕국", "어벤져스:앤드게임", "범죄도시"};  
request.setAttribute("movies", movies);  
%>  
<br>  
<c:forEach var="movie" items="${movies}"  
           begin="0" end="5" step="1" varStatus="varStatus">  
count : ${varStatus.count}, index : ${varStatus.index}, ${movie }<br>  
</c:forEach>
```

```
count : 1, index : 0, 아이언맨  
count : 2, index : 1, 쇼생크탈출  
count : 3, index : 2, 겨울왕국  
count : 4, index : 3, 어벤져스:앤드게임  
count : 5, index : 4, 범죄도시
```

# 다음 방송에서 만나요!

삼성 청년 SW 아카데미