



Jsp

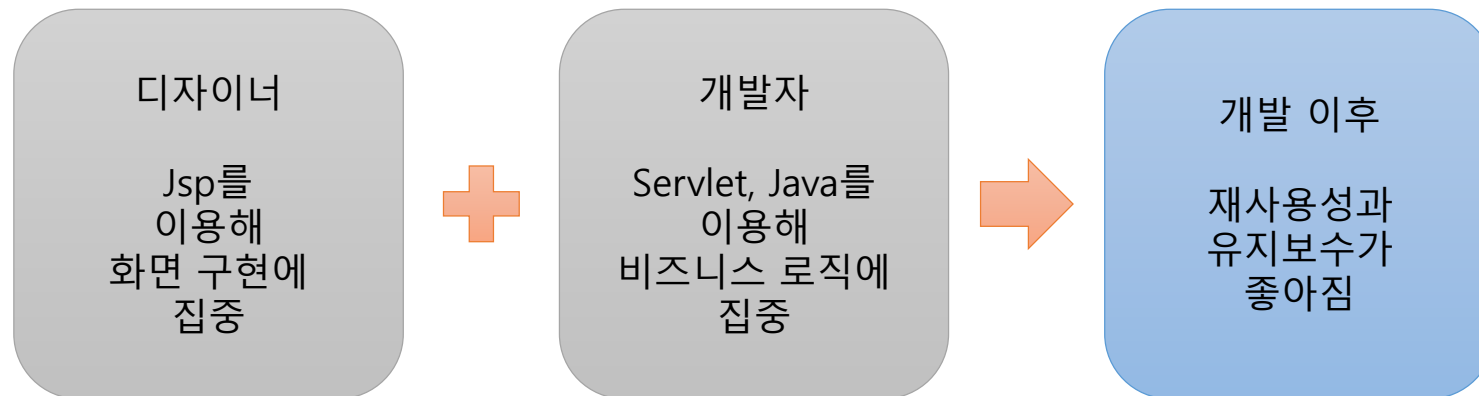
Jsp의 도입

- Servlet의 문제점

- ✓ 자바 코드를 이용해서 웹 화면을 구현하는 것이 바로 Servlet의 원리임
- ✓ Servlet을 활용하면 복잡한 웹 화면을 구현하기가 어려움
- ✓ 자바 코드를 모르면 간단한 웹 화면도 구현할 수 없음
- ✓ 웹 화면을 전문적으로 구현하는 웹 디자이너의 경우 자바 코드를 아는 경우는 매우 드문 케이스임

- Jsp의 도입

- ✓ Servlet : 자바 Base에 HTML을 추가하는 방식
- ✓ Jsp : HTML Base에 자바를 추가하는 방식
- ✓ Jsp는 HTML기반이기 때문에 웹 화면을 보다 쉽게 구현할 수 있음
- ✓ Jsp를 도입하면서 결과 화면은 Jsp로 구현하고 비즈니스 로직은 Servlet으로 구현하는 작업의 분리가 이루어짐

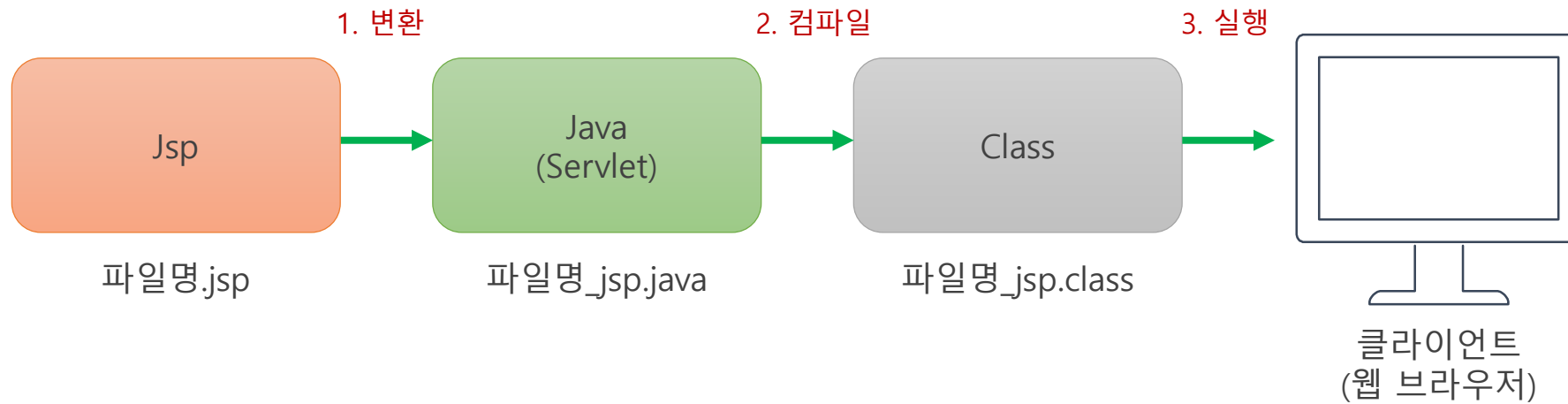


Jsp

- Jsp 구성요소

- ✓ HTML / CSS / JS(JavaScript)
- ✓ Jsp 기본 태그
- ✓ Jsp 액션 태그
- ✓ 커스텀 태그
- ✓ Jsp 스크립트 요소

- Jsp 동작 과정



Jsp 구성 요소

- Jsp 페이지 구성 요소

- ✓ 디렉티브 태그(Directive Tag)
- ✓ 스크립트 요소(Script Element)
 - 주석문(Comment)
 - 표현식(Expression)
 - 선언식(Declaration)
 - 스크립트릿(Scriptlet)
- ✓ 내장 객체
- ✓ 액션 태그(Action Tag)
- ✓ 커스텀 태그(Custom Tag)

Directive

```
<%@page import="java.text.SimpleDateFormat"%>
<%@page import="java.util.Date"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
```

CSS

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
<style type="text/css">
    label {
        display: block;
    }
    label span {
        display: inline-block;
        width: 60px;
    }
</style>
</head>
<body>
```

Scriptlet

```
<%
    Date today = new Date();
    String strToday = new SimpleDateFormat("yyyy-MM-dd").format(today);
    request.setAttribute("today", strToday);
%>
```

내장 객체

HTML

```
<h1>문의 남기기</h1>
<form action="/">
    <label>
        <span>작성자</span>
        <input type="text" name="writer">
    </label>
    <label>
        <span>제목</span>
        <input type="text" name="title">
    </label>
    <textarea rows="5" cols="32" name="content" placeholder="내용"></textarea>
    <br><br>
    <button>문의 남기기</button>
    <input type="reset" value="다시 작성하기">
</form>
```

Directive

```
<%@ include file="footer.jsp" %>
```

```
</body>
</html>
```

디렉티브

- page directive 속성

속성	기본값	작성방법	의미
info	없음	<%@ page info="메인페이지" %>	페이지를 설명하는 문자열
language	java	<%@ page language="java" %>	사용할 언어
contentType	text/html	<%@ page contentType="text/html" %>	페이지 출력 형식
pageEncoding	ISO-8859-1	<%@ page pageEncoding="UTF-8" %>	페이지 문자열 인코딩
import	없음	<%@ page import="java.util.Date" %>	다른 패키지의 클래스 import
session	true	<%@ page session="true" %>	HttpSession 객체 사용 유무
errorPage	없음	<%@ page errorPage="error.jsp" %>	예외 처리 담당 페이지
isErrorPage	false	<%@ page isErrorPage="false" %>	현재 Jsp 페이지의 예외 처리 담당 유무
isELIgnored	true	<%@ page isELIgnored="true" %>	EL 사용 유무
buffer	8KB	<%@ page buffer="8KB" %>	사용할 버퍼 크기
autoFlush	true	<%@ page autoFlush="true" %>	페이지 내용 출력 전에 버퍼가 다 채워질 경우 자동으로 버퍼를 비울 것인지 여부

디렉티브

- include directive

- ✓ 공통으로 사용하는 Jsp 페이지를 다른 Jsp 페이지에 추가할 때 사용함
- ✓ 페이지 상단(헤더 영역)이나 페이지 하단(푸터 영역)의 공통 화면을 만들 때 주로 사용
- ✓ Jsp 페이지의 재사용성이 높고 유지관리가 용이
- ✓ 정적인 페이지를 포함하는 방식으로 동작
- ✓ 형식
`<%@ include file="header.jsp" %>`

- taglib directive

- ✓ 가장 많이 사용되는 커스텀 태그들을 표준화하여 라이브러리로 제공하는 JSTL을 사용할 때 추가함
- ✓ 표현언어(EL), JSTL을 함께 사용할 수 있음
- ✓ 형식
`<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>`

스크립트 요소

- 선언부

- ✓ 전역 변수 선언 및 메소드 정의

```
<%! void method() { } %>
```

- 표현식

- ✓ 변수 및 메소드 호출 결과

```
<%= age %>
```

```
<%= getArea() %>
```

- 주석문

- ✓ 주석 처리
- ✓ 페이지 소스 보기를 통해서 주석의 확인이 불가능함

```
<%-- 주석 --%>
```

- 스크립트릿

- ✓ 일반 자바 코드
- ✓ 지역 변수 선언, 제어문 등 모든 자바 코드

```
<% if(age >= 20) { %>
```

Jsp 내장 객체

- Jsp 내장 객체

- ✓ Jsp가 Servlet으로 변환될 때 컨테이너가 자동으로 생성시키는 서블릿 객체를 의미함
- ✓ Servlet을 사용할 때 자주 사용하던 객체 9개를 컨테이너가 자동으로 생성함
- ✓ 개발자는 Jsp 내장 객체를 일일이 생성할 필요 없이 곧바로 사용할 수 있음

- Jsp 내장 객체 종류

내장 객체	서블릿 타입	의미	저장소
request	javax.servlet.http.HttpServletRequest	클라이언트 요청 정보 저장	○
response	javax.servlet.http.HttpServletResponse	서버 응답 정보	
out	javax.servlet.jsp.JspWriter	Jsp 페이지에 결과 출력	
session	javax.servlet.http.HttpSession	세션 정보 저장	○
application	javax.servlet.ServletContext	컨텍스트 정보 저장	○
pageContext	javax.servlet.jsp.PageContext	Jsp 페이지에 대한 정보 저장	○
page	java.lang.Object	Jsp 페이지의 서블릿 인스턴스 저장	
config	javax.servlet.ServletConfig	Jsp 페이지의 설정 정보 저장	
exception	java.lang.Exception	예외 처리	

Jsp 저장 영역

- Jsp 저장 영역
 - ✓ Jsp 내장 객체 중에서 데이터를 저장할 수 있는 4개 객체를 의미함
 - ✓ 데이터를 속성(Attribute)으로 저장할 수 있음
 - ✓ 종류 : pageContext, request, session, application
- Jsp 저장 영역과 스코프

저장 영역		서블릿 타입	스코프
pageContext	this		동일한 Jsp 내에서 공유
request	HttpServletRequest		한 번의 요청에 대해서 공유
session	HttpSession		같은 브라우저 내에서 공유
application	ServletContext		같은 애플리케이션 내에서 공유

Jsp 액션 태그

- Jsp 액션 태그
 - ✓ 스크립트릿(Scriptlet)에 작성하는 자바 코드 대신 사용할 수 있는 Jsp 태그
 - ✓ 웹 디자이너들의 입장에서 좀 더 쉬운 형태로 자바 기능을 제공하는 개념
 - ✓ prefix로 'jsp'를 사용함
- 주요 Jsp 액션 태그

태그	의미
<jsp:include>	다른 Jsp를 현재 Jsp에 포함시키는 태그
<jsp:param>	<jsp:include> 태그의 하위 태그로 포함할 Jsp에 파라미터를 전달하는 태그
<jsp:forward>	RequestDispatcher 클래스의 forward 기능을 수행하는 태그
<jsp:useBean>	객체 생성(new) 태그
<jsp:setProperty>	Setter
<jsp:getProperty>	Getter

표현 언어

- 표현 언어

- ✓ EL, Expression Language
- ✓ Jsp에 존재하는 각종 스크립트 요소로 인해 웹 디자이너들은 여전히 작업을 어려워 하는 상황이 발생함
- ✓ 스크립트 요소 중 데이터 출력을 담당하던 표현식(<%=값%>)을 대체하기 위해 표현 언어가 등장함
- ✓ 형식

`${값}`

- 표현 언어 주요 특징

- ✓ 기존의 표현식보다 쉬운 형태를 제공
- ✓ 연산자 사용 가능
- ✓ 표현 언어 자체 내장 객체를 제공
- ✓ Jsp 저장 영역에 저장된 속성을 출력할 수 있음

중요

- 표현 언어 자료형

- ✓ 불(true / false)
- ✓ 정수
- ✓ 실수
- ✓ 문자열('hello' / "hello")
- ✓ 널(null)

표현 언어 연산자

- 표현 언어 연산자

종류	연산자	의미
산술 연산자	+ - *	더하기 빼기 곱하기
	/ 또는 div	나누기
	% 또는 mod	나머지
관계 연산자	> 또는 gt	크다
	>= 또는 ge	크거나 같다
	< 또는 lt	작다
	<= 또는 le	작거나 같다
	== 또는 eq	같다
	!= 또는 ne	같지 않다
논리 연산자	&& 또는 and	논리곱
	또는 or	논리합
	! 또는 not	부정
조건 연산자	조건식 ? 값1 : 값2	조건식이 true이면 값1, 조건식이 false이면 값2
empty 연산자	empty 값	값이 null 또는 빈 문자열이면 true

표현 언어 내장 객체

- 표현 언어 내장 객체

종류	내장 객체	의미
스코프	pageScope	pageContext에 저장된 속성을 참조할 때 사용(우선순위 1)
	requestScope	request에 저장된 속성을 참조할 때 사용(우선순위 2)
	sessionScope	session에 저장된 속성을 참조할 때 사용(우선순위 3)
	applicationScope	application에 저장된 속성을 참조할 때 사용(우선순위 4)
요청 파라미터	param	요청 파라미터를 참조할 때 사용(변수) request.getParameter() 대신 사용
	paramValues	요청 파라미터를 참조할 때 사용(배열) request.getParameterValues() 대신 사용
헤더	header	요청 헤더 이름을 단일 값으로 반환
	headerValues	요청 헤더 이름을 여러 값(배열)으로 반환
쿠키	Cookies	쿠키 값을 반환
Jsp 페이지	pageContext	pageContext 객체를 참조할 때 사용

JSTL

- JSTL

- ✓ JSTL, Jsp Standard Tag Library
- ✓ 자주 사용하는 자바 코드(if, for 등)를 태그 형식으로 제공하는 태그 라이브러리
- ✓ Tomcat에서 기본적으로 제공되지 않기 때문에 라이브러리를 추가해야 사용할 수 있음

JSTL 라이브러리 다운로드
<https://tomcat.apache.org/download-taglibs.cgi>

Standard-1.2.5

Source Code Distributions

- [Source README](#)
- [zip](#) (pgp, sha512)

Jar Files

- [Binary README](#)
- Impl:
 - [taglibs-standard-impl-1.2.5.jar](#) (pgp, sha512)
- Spec:
 - [taglibs-standard-spec-1.2.5.jar](#) (pgp, sha512)
- EL:
 - [taglibs-standard-jstlel-1.2.5.jar](#) (pgp, sha512)
- Compat:
 - [taglibs-standard-compat-1.2.5.jar](#) (pgp, sha512)

JSTL 라이브러리

- JSTL 라이브러리 종류

라이브러리	주요 기능	접두어(prefix)	URI
코어	변수 처리, 제어문 처리	c	http://java.sun.com/jsp/jstl/core
국제화	숫자 및 날짜 형식, 지역	fmt	http://java.sun.com/jsp/jstl/fmt
XML	XML 코어, XML 반환	x	http://java.sun.com/jsp/jstl/xml
데이터베이스	SQL	sql	http://java.sun.com/jsp/jstl/sql
함수	컬렉션 처리, 문자열 처리	fn	http://java.sun.com/jsp/jstl/functions

- JSTL 태그 사용을 위한 디렉티브

라이브러리	디렉티브
코어	<code><%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %></code>
국제화	<code><%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %></code>
XML	<code><%@ taglib prefix="x" uri="http://java.sun.com/jsp/jstl/xml" %></code>
데이터베이스	<code><%@ taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql" %></code>
함수	<code><%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %></code>

JSTL Core 라이브러리

- Core 라이브러리
 - ✓ 변수 선언, 조건문, 반복문 등의 기능을 지원하는 태그
 - ✓ 자바의 주요 기능을 태그로 대신할 수 있기 때문에 매우 유용한 라이브러리임
 - ✓ Core 라이브러리를 사용하면 Jsp 페이지 내에서 자바 코드를 대부분 없앨 수 있음
- Core 라이브러리 태그 기능

종류	태그	의미
변수	<c:set>	변수 선언
	<c:remove>	변수 제거
제어문	<c:if>	조건문
	<c:choose>	switch문(하위 태그로 <c:when>, <c:otherwise> 있음)
	<c:forEach>	반복문
	<c:forTokens>	구분자를 이용해 토큰 분리
URL	<c:import>	다른 자원을 Jsp 페이지로 가져옴
	<c:redirect>	리다이렉트(하위 태그로 <c:param> 사용 가능)
	<c:url>	지정한 URL로 이동(하위 태그로 <c:param> 사용 가능)
기타	<c:catch>	예외 처리
	<c:out>	JspWriter에 내용 처리 후 출력