



SEOIL UNIVERSITY

JSP 01

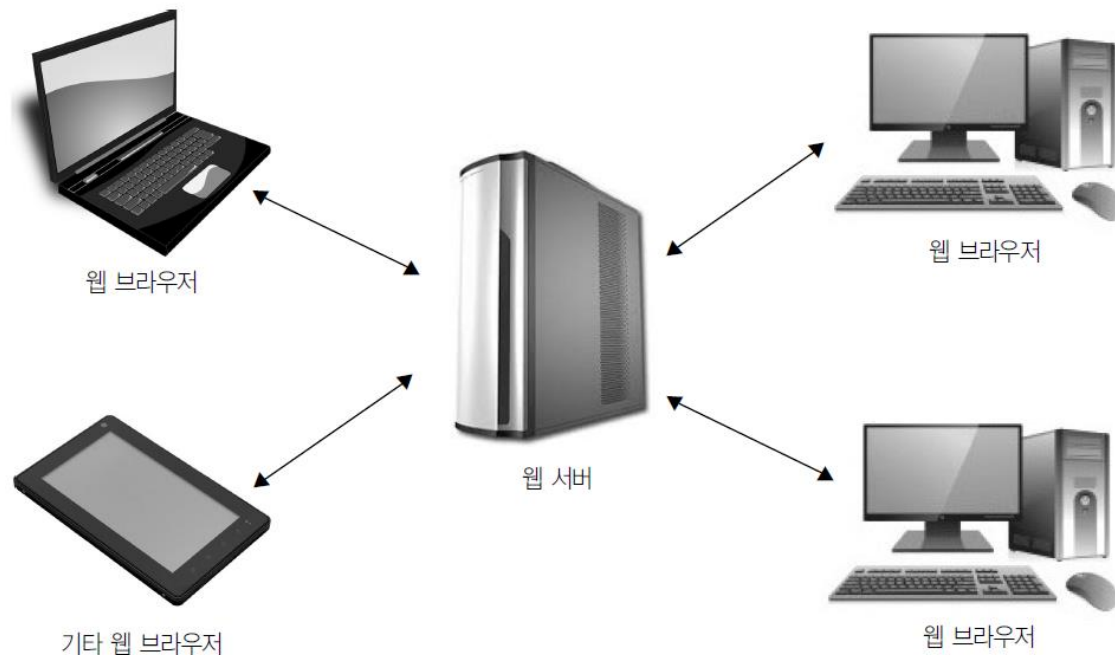


SEOIL UNIVERSITY

New Vision,
Young Challenge!

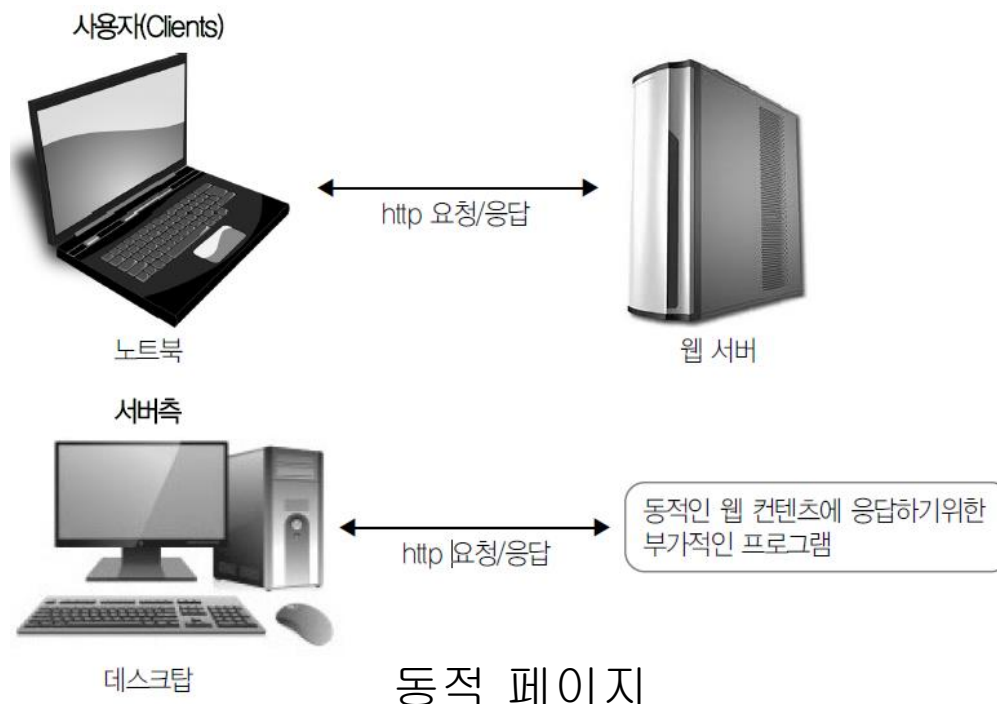
내게 힘이 되는대학! 서일대학교

- 요청 : 클라이언트에서 서버로 정보를 요구하기 위해 보내는 메시지입니다. 이 요청 방식에는 GET방식과 POST 방식
- 응답 : HTTP에서 요구된 메시지에 대한 응답, HTML, 이미지 등이 응답의 내용이 됨



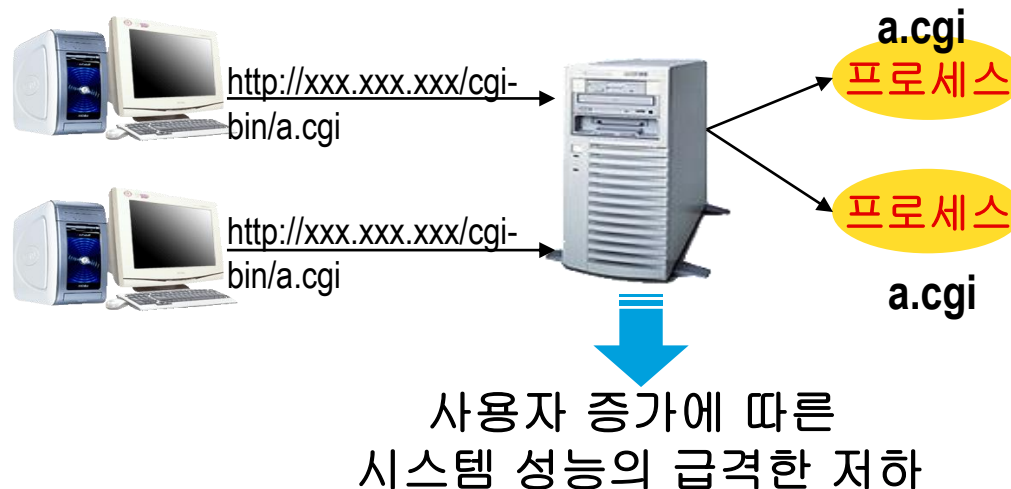
■ 정적 페이지, 동적 페이지

- 정적 페이지 : 변하지 않는 성격을 가진 페이지
- 동적 페이지 : 자동으로 생성된 페이지, 기존의 웹 서버에 새로운 기능을 가진 프로그램을 탑재하여 동적으로 HTML을 생성



■ CGI(Common Gateway Interface)

- 동적 페이지를 위한 기술
- 웹 서버와 동적 콘텐츠 생성을 맡은 프로그램 사이에서 정보를 주고 받는 인터페이스
- 초기 웹 프로그래밍에 사용된 기술
- 프로세스 단위로 실행되기 때문에 사용자 증가 시 급격한 성능 저하



■ 확장 CGI

- 전통적인 CGI 방식의 단점 보완
- 프로세스 생성방식의 변화로 시스템의 부하를 줄임
- ASP, PHP, Servlet, JSP 등



ASP

MS사에서 만들어진
확장 CGI,
특정 웹 서버와 OS에
동작



PHP

ASP와는 달리
특정 영역에서만 동작
하지않음,
적은 명령어들로서 프
로그래밍이 가능, 기
능들이 미약



Servlet

Sun사, Java 언어를
기반으로 하여 동적인
컨텐츠를 생성,
Java 코드안에 HTML
태그가 혼재되어 있어
서 작업에 대한 분리
적인 측면에서 효율성
이 떨어짐

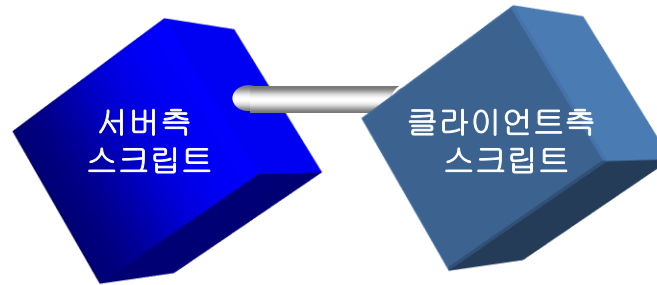


JSP

Sun사, Java언어를
기반, HTML태그 사
이에 스크립트 언어
형식으로 프로그램
을 작성

■ 동적 콘텐츠 처리를 위한 기술들

용 어		설명
CGI		<ul style="list-style-type: none"> - 동적 콘텐츠 처리를 위한 가장 전통적인 방법 - CGI용 프로그램은 CGI 규약만 지키면 어떤 언어라도 가능 - 프로세스 기반 프로그램으로 시스템의 효율성 측면에서 시스템에 많은 부하
확장 CGI	ASP	<ul style="list-style-type: none"> - 비주얼 베이직 언어의 문법들중 일부를 사용 - ActiveX 컴포넌트로 접근가능 - IIS 와 사용(윈도우 운영체제)
	PHP	<ul style="list-style-type: none"> - C언어와 비슷한 문법을 사용 - 리눅스처럼 소스가 공개되어 있음 - 윈도우, 유닉스 운영체제지원, 아파치, IIS 서버 등 지원
	Servlet/JSP	<ul style="list-style-type: none"> - Java 언어를 기반으로 하여 Java 언어의 장점을 수용 - 플랫폼에 대해 독립적임과 동시에 컴파일된 파일을 재컴파일 없이 즉시적인 이식이 가능 - 사용자정의 태그로 기능 확장이 가능



- ✓ 서버측에서 동적인 콘텐츠 생성을 담당하는 기술
- ✓ 서버측에 위치해서 실행되는 스크립트 언어
- ✓ **HTML**과 스크립트 언어를 함께 사용할 수 있는 기술로 웹 서버에서 해석
- ✓ 데이터베이스 연동 처리 등 다양한 구현이 가능
- ✓ 별도의 컴파일 과정없이 **HTML** 태그 수정 가능
- ✓ **JSP, Servlet, PHP, CGI, ASP**

- ✓ 웹 브라우저에서 실행되는 언어
- ✓ 자바 스크립트, **VB** 스크립트

- 웹 프로그래밍 언어들 중의 하나
- 동적인 페이지를 생성하기 위한 서버측 스크립트 언어
- 자바언어를 기반
 - 객체지향적
 - 플랫폼 독립적
 - 네트워크 지향적
 - 뛰어난 보안성
 - 멀티스레드 기능
 - 친근한 코드

■ 서버 스크립트인 JSP로 구현한 예

```
<%@ page import="java.util.*,java.text.*" %>
<html>
<head><title>example</title></head>
<body>
<%
    Date date = new Date();
    SimpleDateFormat simpleDate = new SimpleDateFormat("yyyy-MM-dd");
    String strdate = simpleDate.format(date);
%>
오늘 날짜는 <%=strdate%>입니다.
</body>
</html>
```

빈즈 자바컴포넌트를 사용가능

최초 서블릿컴파일 후 메모리에서 처리
(사용자 접속이 많아도 원활하게 처리)

다른 서블릿간의 간편한 데이터 공유
(page, request, session, application scope)

자바의 모든 기능을 사용가능
(확장성)

IBM, 오라클, 썬, BEA 등에서 강력하게 지원

사용자정의 태그 생성 및
다양한 기능의 태그라이브러리 이용가능(JSTL 등)

다양한 운영체제와 개발/실행환경 지원

JSP의 특징

완전한 자바파일의 형태

JSP와는 달리
자바 코드 안에 **HTML**
태그들이 포함

JSP 페이지는 하나의
서블릿으로 변환이 되고
다시 컴파일 되어 실행

JSP는 브라우저에 표현
이 될 부분에 보다 유용
하고 서블릿은 브라우저
에 표시될 필요가 없지
만 내부적으로 처리되어
야 할 부분에 보다 유용
하게 사용

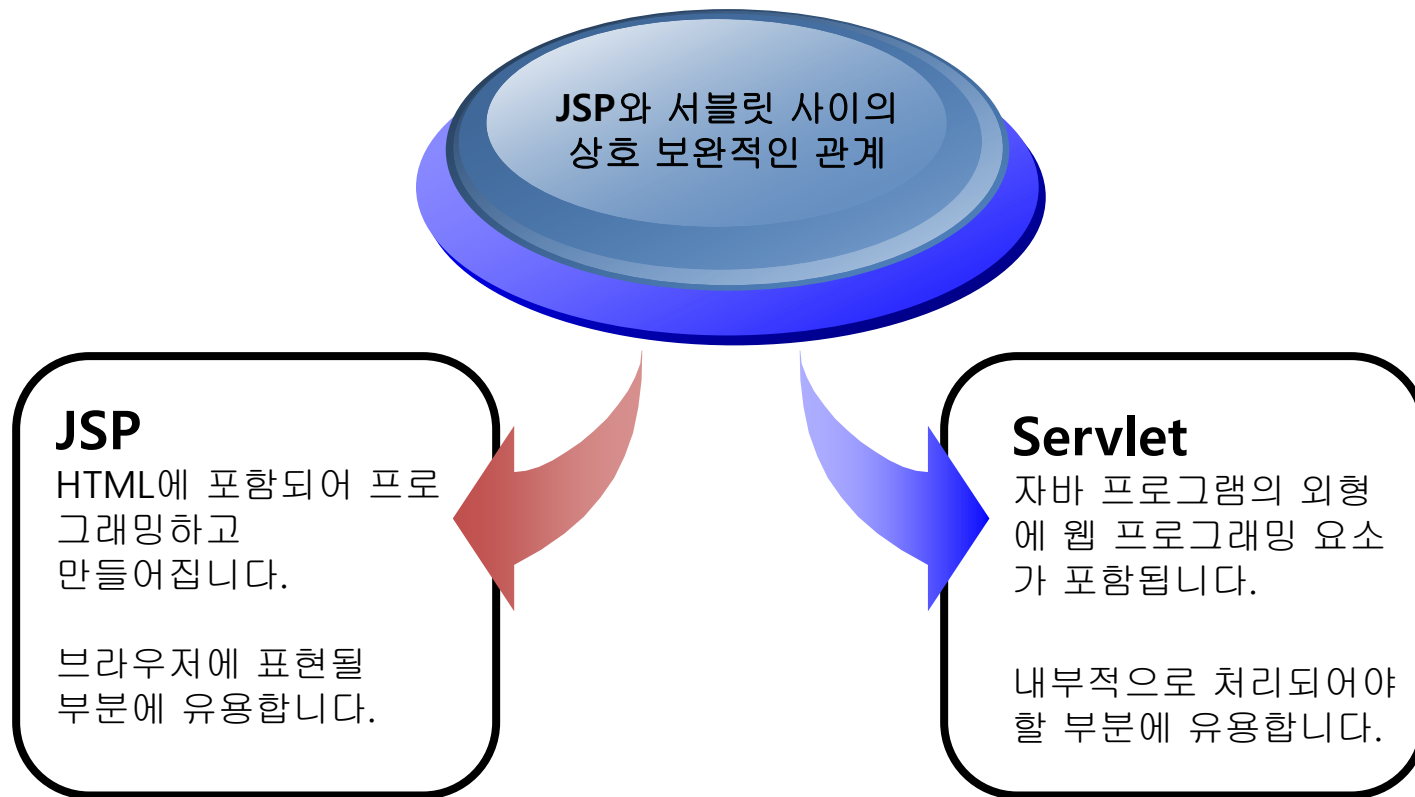
❖ Servlet으로 구현한 예

```
public class ServletForm extends HttpServlet{
    public void service(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException{
        PrintWriter out;
        String username = request.getParameter("UserName");
        String userpass = request.getParameter("UserPass");

        response.setContentType("text/html;charset=euc-kr");
        out = response.getWriter();

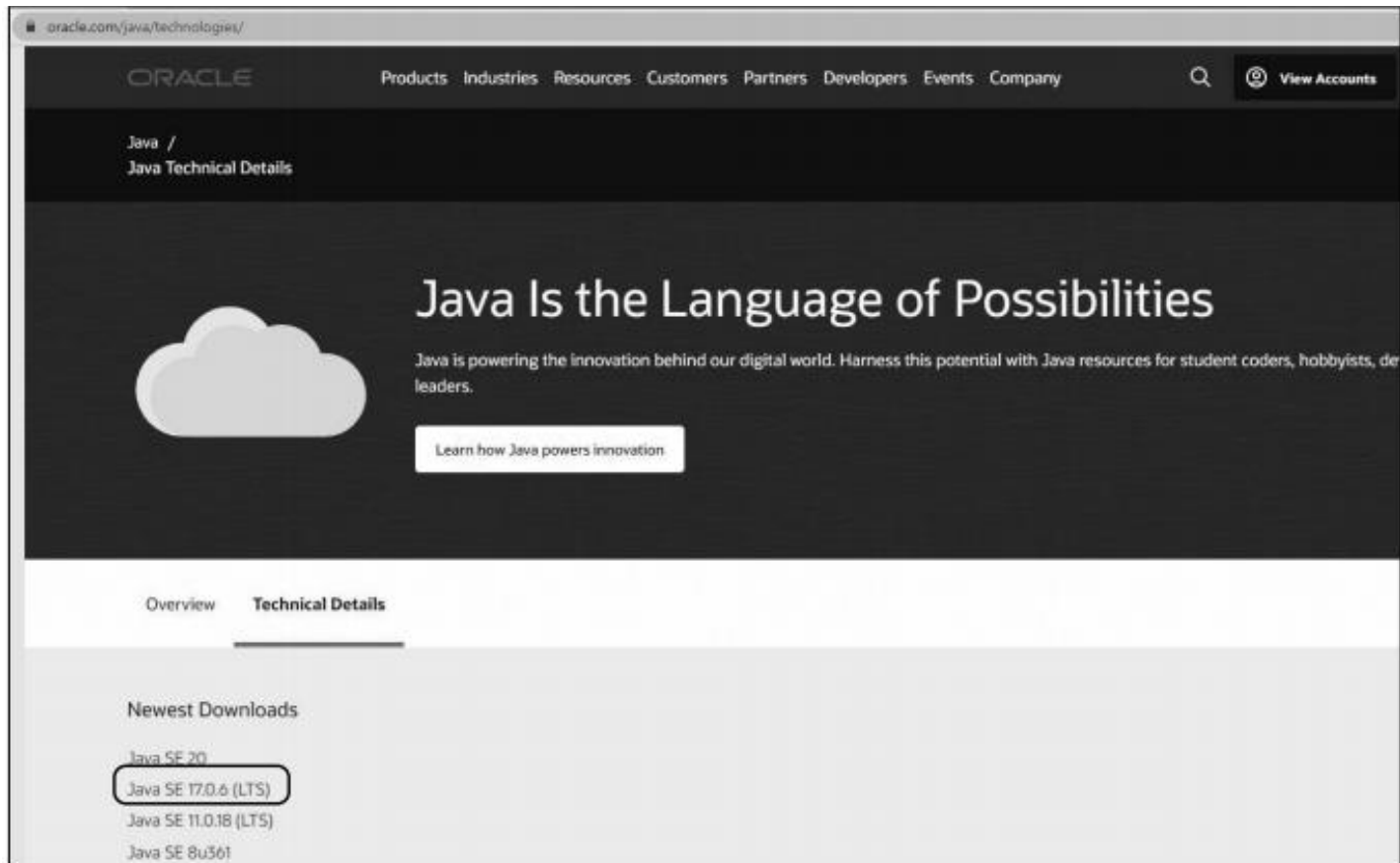
        out.println("<HTML>");
        out.println("<BODY>");
        ...
        out.println("</HTML>");
        out.println("</BODY>");

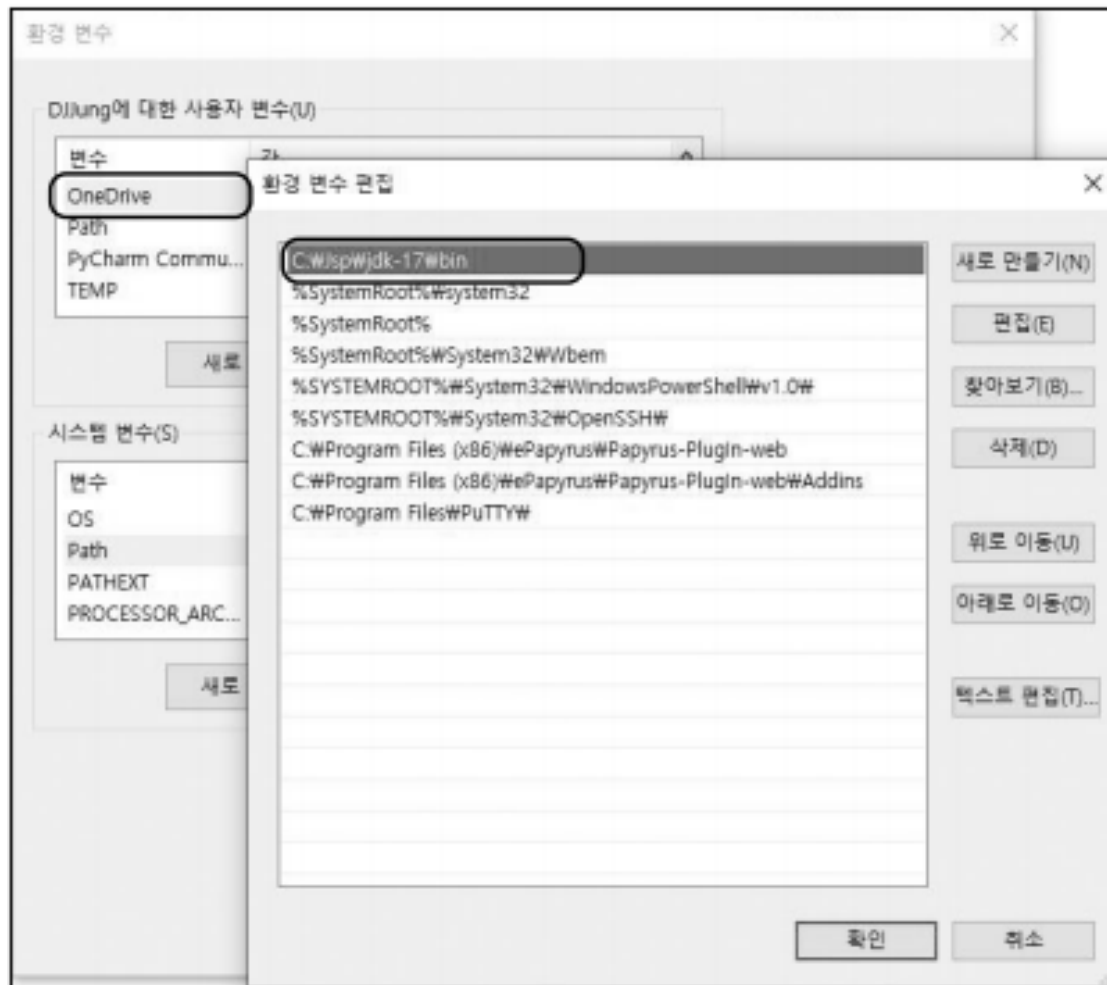
        out.close();
    }
}
```





- JDK 다운로드
- <https://www.oracle.com/java/technologies>





- Java
- Javac

C:\W>java -version ↵ Enter

```
C:\Windows\system32\cmd.exe
C:\W>java -version
java version "11.0.1" 2018-10-16 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.1+13-LTS)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.1+13-LTS, mixed mode)

C:\W>
```

■ JSP 컨테이너(Container)란?

- JSP를 사용하기 위해서 웹서버와 JSP 컨테이너가 필요
- JSP 컨테이너는 JSP를 동작할 수 있도록 하는 웹서버
- 동적인 페이지 생성을 맡아서 처리할 프로그램으로 JSP 페이지에 대한 요청을 처리

<http://tomcat.apache.org/>
에서 무료로 다운로드 및 사용 가능

Sun사에서 개발,
JSP, Servlet의 공식적인 컨테이너

독립적으로
웹서버(IIS, Apache)와 연동가능

Tomcat의 특징

← → ↺ 🏠 🔒 안전함 | https://tomcat.apache.org/download-90.cgi

Download

Which version?
Tomcat 9
Tomcat 8
Tomcat 7
Tomcat Connectors
Tomcat Native
Taglibs
Archives

Documentation

Tomcat 9.0
Tomcat 8.5
Tomcat 8.0
Tomcat 7.0
Tomcat Connectors
Tomcat Native
Wiki
Migration Guide
Presentations

Problems?

Security Reports
Find help
FAQ
Mailing Lists
Bug Database
IRC

Get Involved

Overview
Source code
Buildbot
Tools

Media

Twitter
YouTube
Blog

the OpenPGP keys of Tomcat's Release Managers. We also provide SHA-1 and SHA-512 checksums for every release file. After you download the file, you for your download, and make sure it is the same as ours.

Mirrors

You are currently using **http://apache.tt.co.kr/**. If you encounter a problem with this mirror, please select another mirror. If all mirrors are failing, there end of the mirrors list) that should be available.

Other mirrors:

9.0.8

Please see the [README](#) file for packaging information. It explains what every distribution contains.

Binary Distributions

- Core:
 - [zip \(pgp, sha1, sha512\)](#)
 - [tar.gz \(pgp, sha1, sha512\)](#)
 - [32-bit Windows zip \(pgp, sha1, sha512\)](#)
 - [64-bit Windows zip \(pgp, sha1, sha512\)](#)
 - [32-bit/64-bit Windows Service Installer \(pgp, sha1, sha512\)](#)
- Full documentation:
 - [tar.gz \(pgp, sha1, sha512\)](#)
- Deployer:
 - [zip \(pgp, sha1, sha512\)](#)
 - [tar.gz \(pgp, sha1, sha512\)](#)
- Extras:
 - [JMX Remote jar \(pgp, sha1, sha512\)](#)
 - [Web services jar \(pgp, sha1, sha512\)](#)
- Embedded:
 - [tar.gz \(pgp, sha1, sha512\)](#)
 - [zip \(pgp, sha1, sha512\)](#)

Source Code Distributions

- [tar.gz \(pgp, sha1, sha512\)](#)
- [zip \(pgp, sha1, sha512\)](#)

Apache Tomcat Setup: Configuration Options

Configuration
Tomcat basic configuration.

Server Shutdown Port: 8005

HTTP/1.1 Connector Port: 80

AJP/1.3 Connector Port: 8009

Windows Service Name: Tomcat9

Create shortcuts for all users: ☐

Tomcat Administrator Login (optional)

User Name: admin

Password:

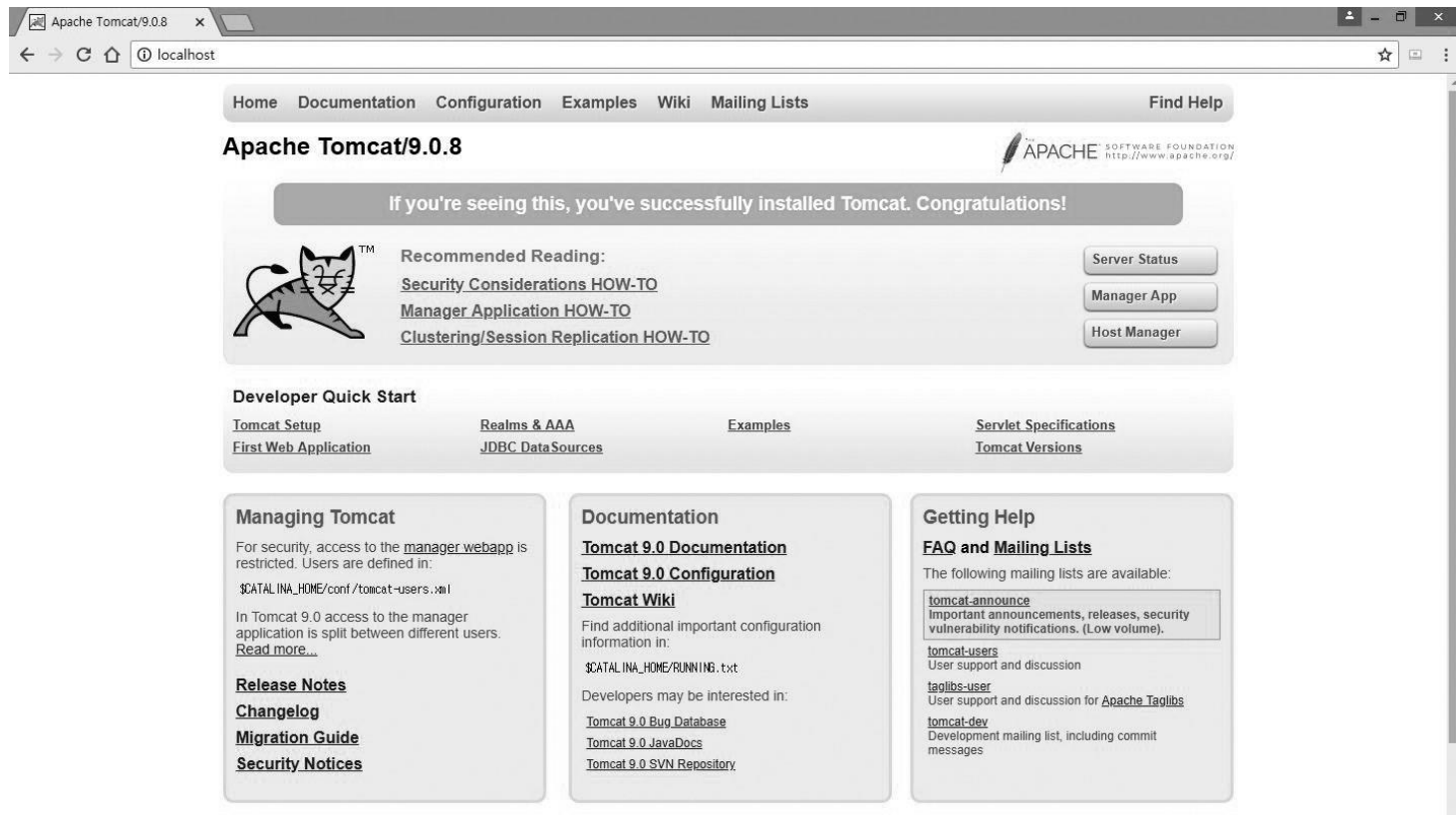
Roles: admin-gui,manager-gui

Nullsoft Install System v3.03

< Back Next > Cancel

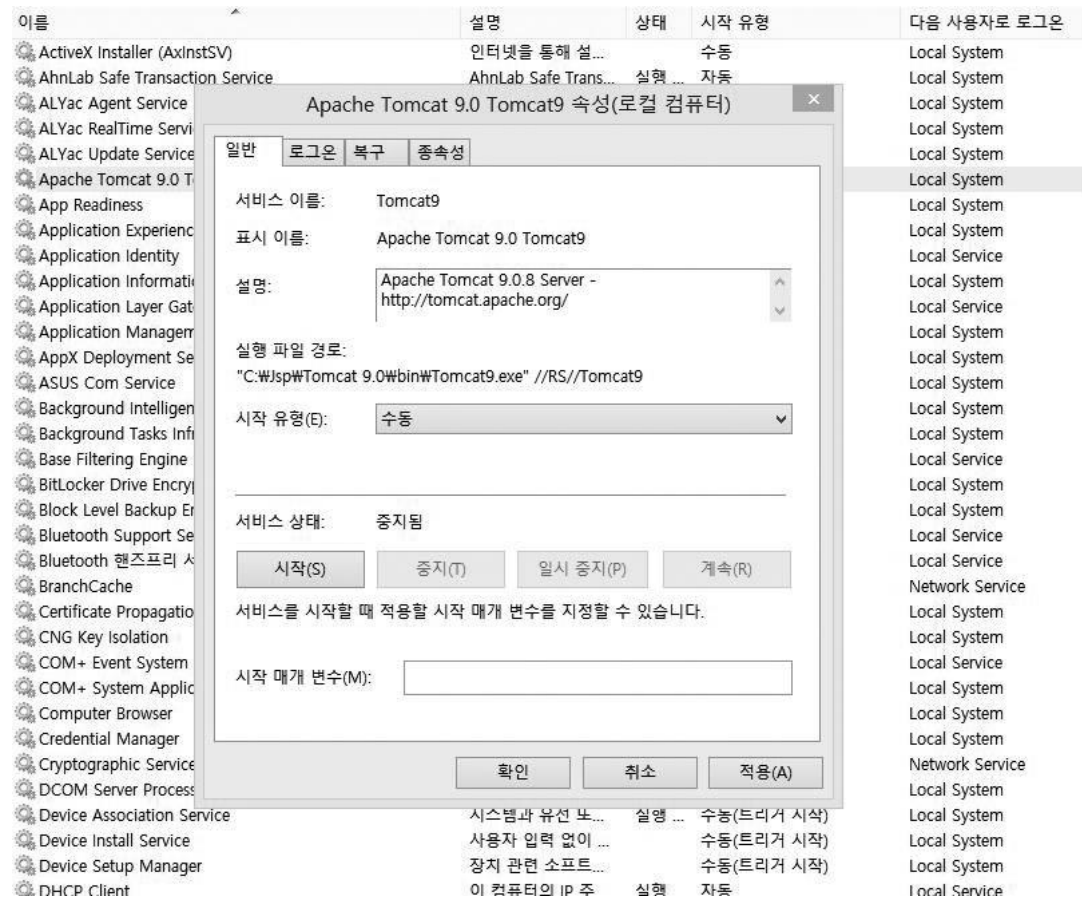
■ 정상작동 여부 확인

- 브라우저 주소창에 'localhost' 입력하여 톰캣 정상작동 여부 확인

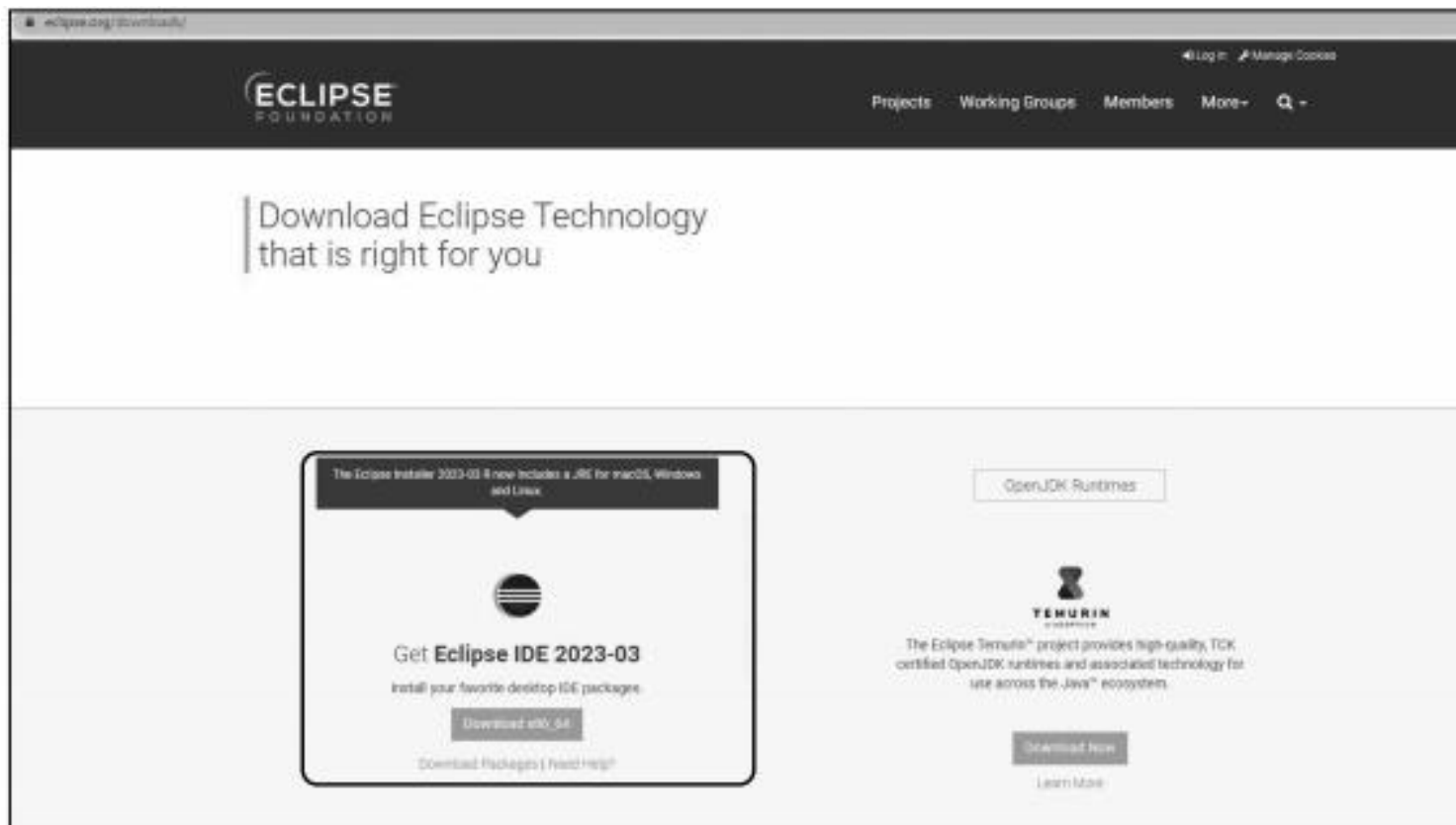


■ 서비스 설정

- 이클립스 관리 목적을 위해 윈도우 [서비스]에서 시작 유형을 [수동]



- eclipse 다운로드 및 압축해제
- <http://eclipse.org/downloads>



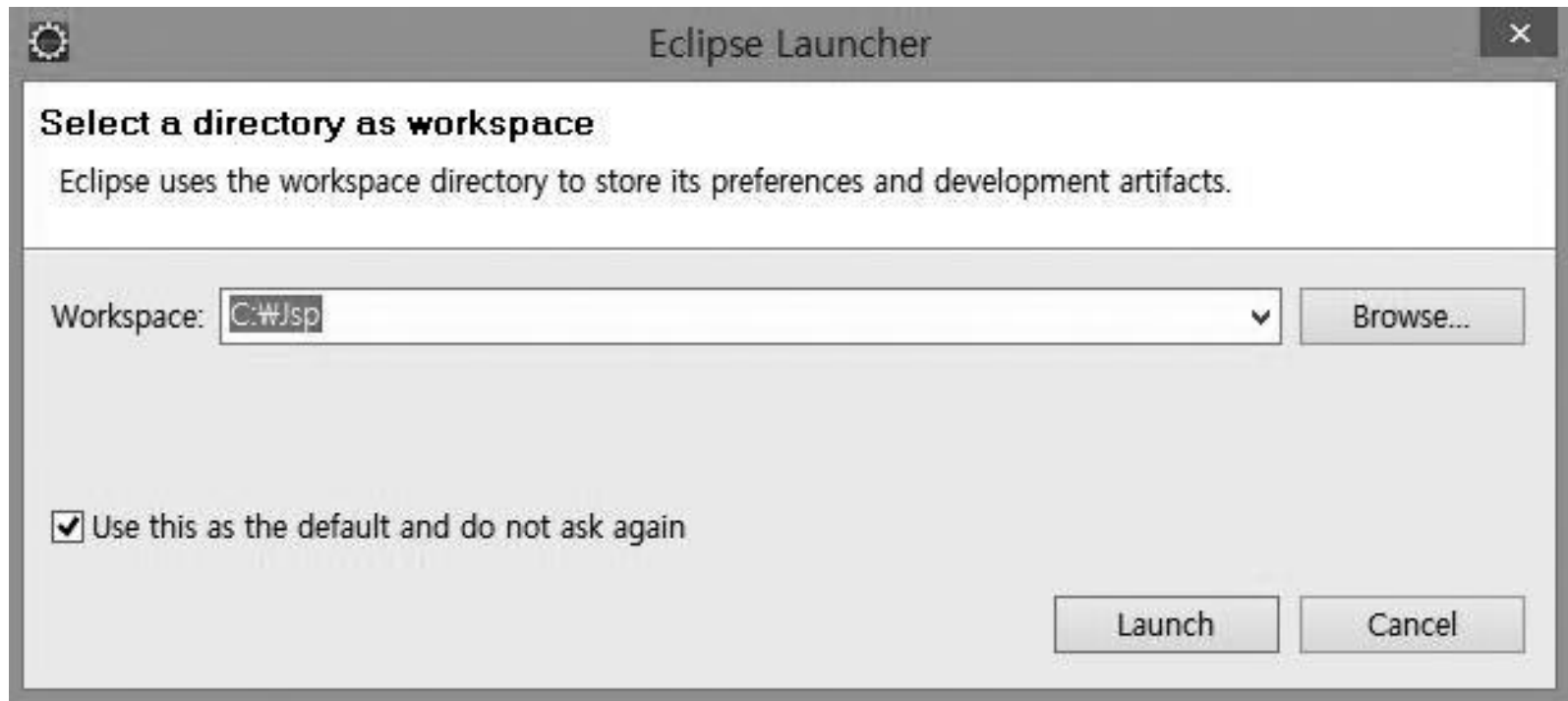
■ eclipse 인스톨러 실행

- Eclipse IDE for Java EE Developers 선택



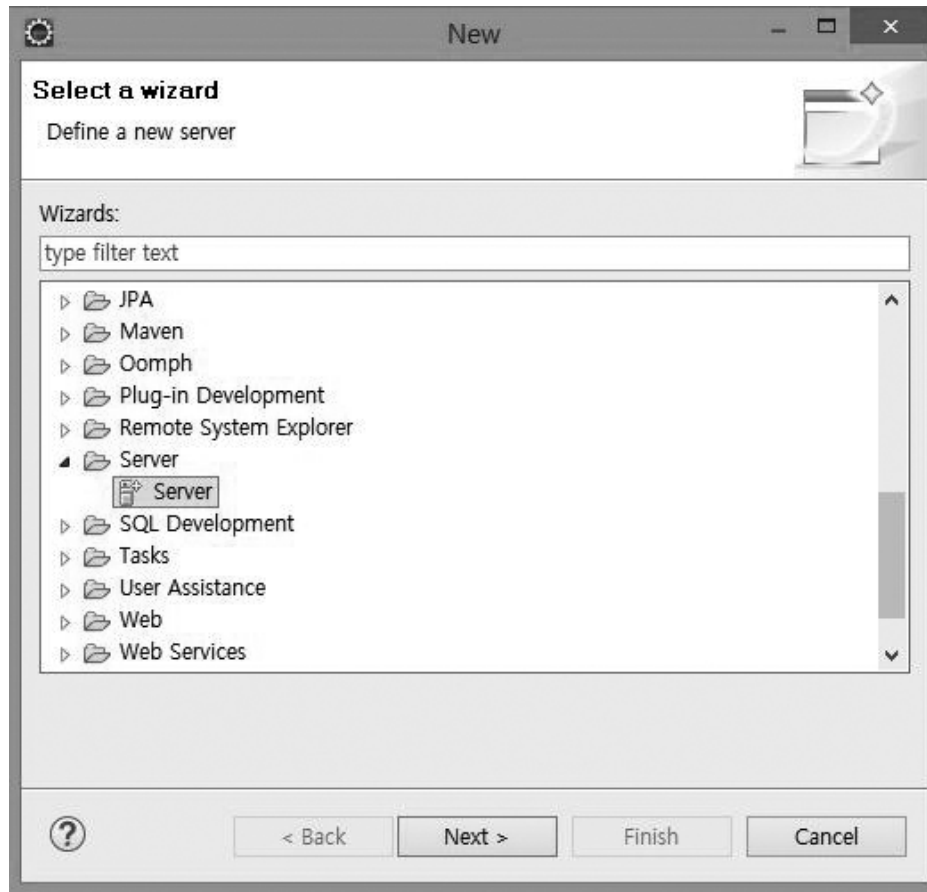
■ Workspace 경로 지정

- 'C:₩Jsp'로 Workspace 경로 지정



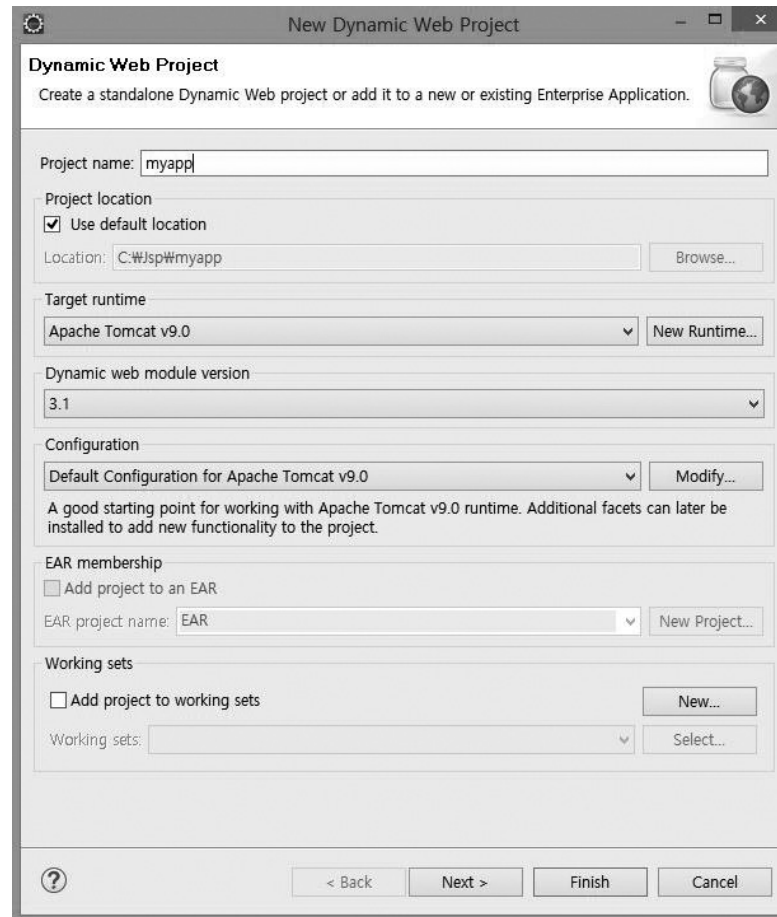
■ Tomcat 9.0 지정

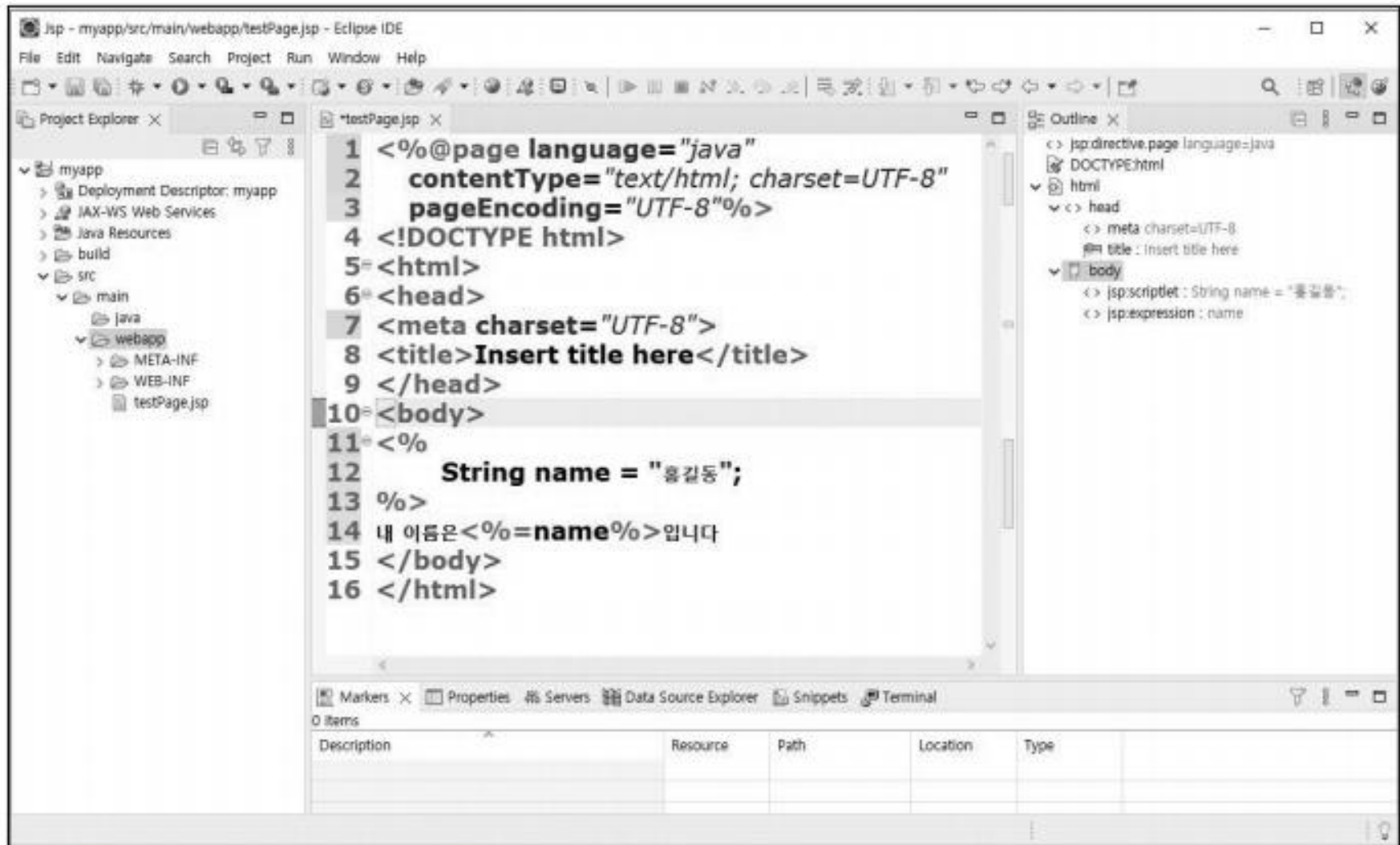
- 상단 메뉴의 [File]-[New]-[Other]-[Server]를 통해 Tomcat9.0 설치 경로 지정



■ 프로젝트 생성

- [File]-[New]-[Dynamic Web Project]를 통해 프로젝트 myapp 생성







■ 스크립트 요소란?

- JSP 프로그래밍에서 사용되는 문법의 표현 형태

■ 스크립트 요소

- 선언문(Declaration)
- 스크립트릿 (Scriptlet)
- 표현식 (Expression)
- 주석 (Comment)

```
<%@ page contentType="text/html;charset=EUC-KR"%>
<title>JSP스크립트 Example</title>
<body>
  <h1>Script Example1 </h1>
  <%!
      String declaration = "Declaration";

  %>
  <%!
      public String decMethod(){
          return declaration;
      }
  %>
  <%
      String scriptlet = "Scriptlet";
      String comment = "Comment";

      out.println("내장객체를 이용한 출력 : " + declaration + "<p>");
  %>
```

선언문의 출력1 : <%=declaration%> <p>
 선언문의 출력2 : <%=decMethod()%> <p>
 스크립트릿의 출력 : <%=scriptlet%> <p>
 <!--JSP주석부분-->
 <!-- JSP 주석1 : <%=comment%> --> <p>
 <%-- JSP 주석2 : <%=comment%> --%>
 <% /* 주석
 (여러줄 주석)
 */
 %>
 <%// 주석(한줄 주석)%>
 </body>
 </html>

Script Example1

내장객체를 이용한 출력 : Declaration

선언문의 출력1 : Declaration

선언문의 출력2 : Declaration

스크립트릿의 출력 : Scriptlet

❖ 선언문이란?

- JSP에서 사용될 변수나 메소드를 선언할 수 있는 영역들의 요소
- JSP 페이지 내에서 변수 및 메소드를 선언 후 변수나 메소드를 이용하여 필요한 동적으로 **HTML**코드를 생성하는데 사용
- 멤버 변수 : 선언문에 선언된 변수
- 선언문에서 선언된 변수로 호칭하는 것은 자바에서 변수의 종류가 여러 가지 있다는 의미
- 선언문 문법

```
<%!  
    멤버변수 및 메소드를 선언하는 영역  
%>
```

- 선언문에서 선언된 변수는 JSP 페이지가 서블릿 코드로 변환이 되면 서블릿 클래스의 멤버변수로 변환됨.

```
<%!  
    String name = "JSPStudy";  
    int year = 2013;  
%>
```

<서블릿 클래스에서 선언된 멤버변수>

변수명	변수값
name	JSPStudy
year	2013

```
<%@ page contentType="text/html;charset=EUC-KR"%>
<h1>Declaration Example1 </h1>
<%
    String name = team + " Fighting!!!";
%>
<%!
    String team = "Korea";
%>
출력되는 결과는 ? <%=name%>
```

Declaration Example1

출력되는 결과는 ? Korea Fighting!!!

- 선언문에서 선언된 메소드는 JSP페이지 내에서는 일반적인 형태의 메소드로 선언됨.

```
<%!  
    String name = "Korea";  
    public String getName(){  
        return name;  
    }  
%>
```

- getName() 메소드는 멤버 변수 name값을 리턴 시켜주는 메소드로 선언.
- name변수가 선언문에서 선언이 되었기 때문에 멤버변수의 역할이 되면서 접근이 가능한 것.

```
<%@ page contentType="text/html;charset=EUC-KR"%>
<h1>Declaration Example2</h1>
<%!
    int one;
    int two = 1;

    public int plusMethod(){
        return one + two;
    }

    String msg;
    int three;
%>
```

Declaration Example2

one 와 two의 합은 ? 1
String msg의 값은 ? null
int three의 값은 ? 0

```
one 와 two의 합은 ? <%=plusMethod()%> <p>
String msg의 값은 ? <%=msg%> <p>
int three의 값은 ? <%=three%>
```

■ 스크립트릿이란?

- JSP 페이지가 서블릿으로 변환되고 요청될 때 `_jspService`(Tomcat 기준으로 설명) 메소드 안에 선언이 되는 요소.
- 스크립트릿은 선언문과 달리 선언된 변수는 지역 변수로 선언이 되고 메소드 선언은 할 수 없음.
- 만약 선언을 하게 되면 메소드 안에 메소드를 선언한 것이기 때문에 만들 수가 없음.
- 스크립트릿 문법

```
<%
```

```
    이곳에 필요한 자바코드를 삽입합니다.(지역 변수 선언, for, while, if 등...)
```

```
%>
```

■ 표현식이란?

- 동적인 JSP 페이지를 브라우저로 표현을 하기 위한 요소.
- 변수를 출력하거나 메소드의 결과값을 브라우저에 출력 할 수 있음.
- 스크립트릿 코드 내에서 out이라는 내장객체를 통해 브라우저에 출력 가능.
- 스크립트릿과 달리 변수나 메소드를 출력하고 할 때 세미콜론(;)은 표기하지 않음.
(서블릿 코드로 변환될 때 자동적으로 세미콜론은 붙여짐)
- 표현식 문법

⟨%=변수 혹은 메소드%⟩

```
<%@ page contentType="text/html;charset=EUC-KR"%>
<h1>Scriptlet Example1 </h1>
<%!
    int one;
    String msgOne;
%>
<%
    int two = 31;
    String msgTwo = "Scriptlet Example";
%>
<%= two + " : " + msgTwo%> <br/>
<%=application.getRealPath("/")%>
```

Scriptlet Example1

31 : Scriptlet Example

C:\Jsp\metadata\plugins\org.eclipse.wst.server.core\tmp0\wtpwebapps\myapp\


```
<%@ page contentType="text/html;charset=EUC-KR"%>
<h1>Scriptlet Example2</h1>
<%
    float f = 2.3f;
    int i = Math.round(f);
    java.util.Date date = new java.util.Date();
%>
실수 f의 반올림값은? <%=i%> <p>
현재의 날짜와 시간은? <%=date.toString()%>
```

Scriptlet Example2

실수 f의 반올림값은? 2

현재의 날짜와 시간은? Sat Jun 15 11:31:21 KST 2013

■ 표현식이란?

- 동적인 JSP 페이지를 브라우저로 표현을 하기 위한 요소.
- 변수를 출력하거나 메소드의 결과값을 브라우저에 출력 할 수 있음.
- 스크립트릿 코드 내에서 out이라는 내장객체를 통해 브라우저에 출력 가능.
- 스크립트릿과 달리 변수나 메소드를 출력하고 할 때 세미콜론(;)은 표기하지 않음.
(서블릿 코드로 변환될 때 자동적으로 세미콜론은 붙여짐)
- 표현식 문법

⟨%=변수 혹은 메소드%⟩

■ 표현식이란?

- 동적인 JSP 페이지를 브라우저로 표현을 하기 위한 요소.
- 변수를 출력하거나 메소드의 결과값을 브라우저에 출력 할 수 있음.
- 스크립트릿 코드 내에서 out이라는 내장객체를 통해 브라우저에 출력 가능.
- 스크립트릿과 달리 변수나 메소드를 출력하고 할 때 세미콜론(;)은 표기하지 않음.
(서블릿 코드로 변환될 때 자동적으로 세미콜론은 붙여짐)
- 표현식 문법

⟨%=변수 혹은 메소드%⟩

```
<%@ page contentType="text/html;charset=EUC-KR"%>
<h1>Expression Example1 </h1>
<%!
    String name[] = {"Java","JSP","Android","Struts"};
%>
<table border="1" width="200">
<% for (int i=0;i<name.length;i++){%>
<tr> <td> <%=i%> </td>
<td> <%=name[i]%> </td>
</tr>
<%=}%>
</table>
```

Expression Example1

0	Java
1	JSP
2	Android
3	Struts

```
<%@ page contentType="text/html;charset=EUC-KR"%>
```

```
<h1>Expression Example2</h1>
```

```
<%
```

```
    java.util.Date date = new java.util.Date();
```

```
    int hour = date.getHours();
```

```
        int one = 10;
```

```
        int two = 12;
```

```
%>
```

```
<%!
```

```
    public int operation(int i,int j){
```

```
        return i > j ? i : j;
```

```
    }
```

```
%>
```

```
지금은 오전일까요 오후일까요? <%= (hour < 12)? "오전":"오후"%> <p>
```

```
one 와 two 둘 중에 큰 숫자는? <%= operation(one,two)%>
```

Expression Example2

지금은 오전일까요 오후일까요? 오후

one 와 two 둘 중에 큰 숫자는? 12

■ 주석이란?

- 프로그램에 직접적인 영향을 미치지 않지만 개발자들이 소스 분석 내용 및 파일 설명 처리를 위해서 없어서는 안 될 꼭 필요한 요소

■ HTML 형식의 주석

```
<!-- Fighting <%=name%> -->
```

■ JSP 형식의 주석

```
<%-- Fighting <%=name%> --%>
```

■ 스크립트 요소의 주석

```
<% /*주석.....여러 줄 주석 및 부분 주석)*/ %>
```

```
<% //주석....(한줄 주석)%>
```

```
<%=name/*주석*/%>
```

```
<%=/*주석*/%>
```

```
<%@ page contentType="text/html;charset=EUC-KR"%>
<h1>Comment Example1</h1>
<%
    String name = "Korea";
%>
<!-- 주석 부분입니다. '소스보기'에서 보이지요. -->
<%--
    이 부분은 JSP 페이지에서만 보이고 '소스보기'를 해도 보이지 않습니다.
    브라우저에 보내지 않는 문장입니다.
--%>
<!--<%=name%> 주석에서도 동적인 변수 사용-->
<%=name/* 표현식에 주석 부분 입니다*/%> Fighting!!!
```

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR"/>
</head>
<h1>If-else Example</h1>
<FORM METHOD=POST ACTION="if.jsp">
이름 : <INPUT TYPE="text" NAME="name"> <p>
좋아하는 색깔 : <SELECT NAME="color">
    <option value="blue" selected>파란색</option>
    <option value="red">붉은색</option>
    <option value="orange">오렌지색</option>
    <option value="etc">기타</option>
</SELECT> <p>
<INPUT TYPE="submit" VALUE="보내기">
</FORM>
</body>
</html>
```

```
<%@ page contentType="text/html; charset=EUC-KR"%>
<h1>If-else Example</h1>
<%!
    String msg;
%>
<%
    String name = request.getParameter("name");
    String color = request.getParameter("color");

    if (color.equals("blue")) {
        msg = "파란색";

    } else if (color.equals("red")) {
        msg = "붉은색";

    } else if (color.equals("orange")){
        msg = "오렌지색";

    } else{
        color = "white";
        msg = "기타색";
    }
%>
<body bgcolor=<%=color%>>
<b><%=name%></b>님이 좋아하는 색깔은 <b><%=msg%></b>입니다.
</body>
```



```
<%@ page contentType="text/html; charset=EUC-KR"%>
<h1>For Example</h1>
1에서 10까지 합은 ?<p>
<%
    int i,sum = 0;
        for(i=1;i<=10;i++){
            if(i<10){
%>
                <%= (i + " + ")%>
<%
                }else{
                    out.println(i + " = ");
                }//if-else
            sum += i;
        }//for
%>
<%=sum%>
```



SEOIL UNIVERSITY

감사합니다.