개발환경 구성, JSP 기본구조

524730-1
2019년 봄학기
3/11/2019
박경신

인터넷과 웹 서비스

- 인터넷과 WWW(World Wide Web)
  - 인터넷은 TCP/IP 기반의 네트워크가 전세계적으로 확대되어 하나로 연결된 '네트워크의 네트워크'
  - WWW(World Wide Web)는 인터넷 기반의 서비스 중 하나

이름	프로토콜	포트	기능
WWW	http	80	웹 서비스
Email	SMTP/POP3/IMAP	25/110/114	이메일 서비스
FTP	ftp	21	파일 전송 서비스
telnet	telnet	23	원격 호스트 접속 서비스
DNS	DNS	83	도메인 이름 변환 서비스
News	NNTP	119	인터넷 뉴스 서비스

인터넷과 웹 서비스

- HTTP (Hyper Text Transfer Protocol)
  - Protocol이란 네트워크에 연결된 컴퓨터가 서로 통신(대화)하기 위한 규약
  - HTTP는 www 서비스를 위한 프로토콜
  - 웹 서버와 클라이언트는 HTTP를 이용해 통신
- URL (Uniform Resource Locator)
  - URL은 웹 어플리케이션에 요청을 구분하기 위한 용도로 사용됨
  - 예: http://www.google.com/search?hl=en&q=jsp&aq=f&oq=
    - 프로토콜: http
    - 호스트: www.google.com
    - 포트: 80 (http 프로토콜의 기본 포트)
    - 경로: /search
    - 쿼리문자열: hl=en&q=jsp&aq=f&oq=

웹 프로그래밍 기술

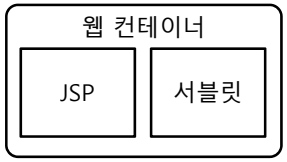
- HTML과 웹 프로그래밍 기술
  - HTML(HyperText Markup Language): www 서비스를 표현하기 위해 사용하는 언어
  - HTML은 텍스트 파일로 정적인 정보만 처리 가능
  - 동적인 컨텐츠 처리하기 위해 CGI(Common Gateway Interface), Fast CGI, PHP(Personal Home Page), ASP(Active Server Page), JSP(Java Server Page) 등의 기술이 사용
- 클라이언트 스크립트
  - Javascript가 대표적. 웹 브라우저가 스크립트 해석의 주체
  - 웹 브라우저 핸들링은 가능하지만 서버 연동은 불가능
- 서버 스크립트
  - ASP.NET, PHP, Servlet, JSP 등. HTML과 스크립트 언어를 함께 사용할 수 있는 기술로 웹 서버에서 해석.
  - 데이터베이스 연동 처리 등 다양한 구현이 가능
  - 별도의 컴파일 과정없이 HTML 태그 수정 가능

서블릿과 JSP

- 자바를 만든 Sun에서 정한 웹 개발 표준
  - 서블릿(Servlet): 웹 어플리케이션을 구현하기 위해 작성하는 **실행 코드**
  - JSP(Java Server Page): 서블릿 기반의 서버 **스크립트** 기술
- JSP의 특징
  - 자바 기반 스크립트 언어: 자바의 기능을 그대로 사용 가능
  - HTTP에 대한 클라이언트의 요청 처리/응답
  - 웹 어플리케이션에서 결과 화면을 생성할 때 주로 사용
- JSP를 사용하는 이유
  - 자바 언어에 기반하여 플랫폼(리눅스, 윈도우 등)에 독립적
  - 자바 언어에 대한 깊은 이해 없이도 초기 학습 가능 - 스크립트 언어는 상대적으로 자바 언어보다 단순
  - **스프링(Spring)**이나 스트러츠(Struts)와 같은 **프레임워크**와 완벽하게 연동

웹 컨테이너

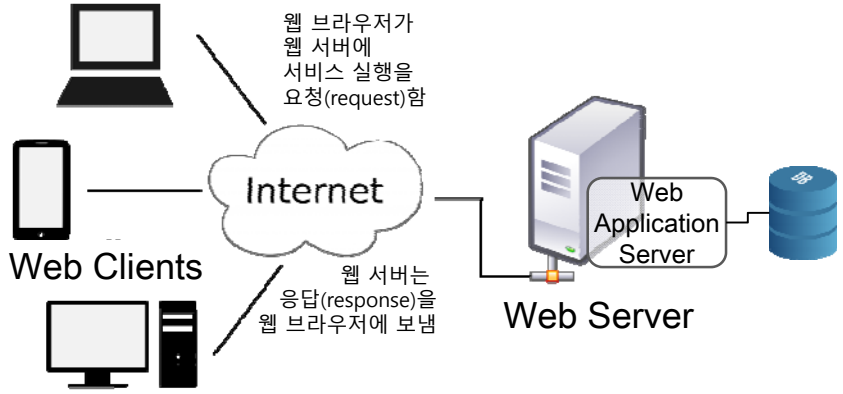
- 웹 컨테이너
  - JSP, Servlet 실행 환경을 제공하며, WAS (Web Application Server)라고 함
  - 웹 브라우저의 요청에 대한 응답을 웹 서버에 전달하는 기능을 제공하며 웹 응용 프로그램의 보안, 관리, 라이프 사이클관리 등의 서비스를 제공



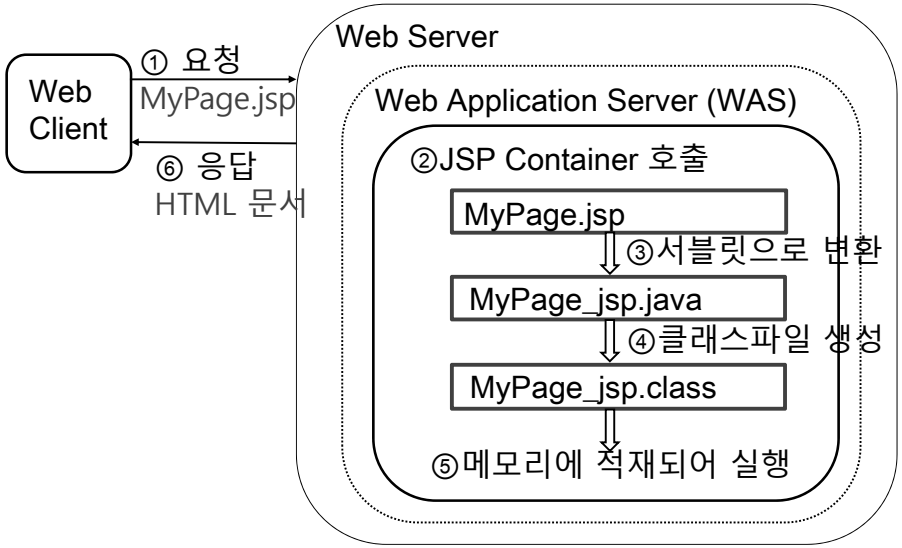
- 주요 웹 컨테이너
  - 톰캣(Tomcat) : http://tomcat.apache.org/
  - 제티(Jetty) : http://www.mortbay.org/jetty/

JSP Architecture

- JSP(Java Server Pages) 웹 어플리케이션 구조

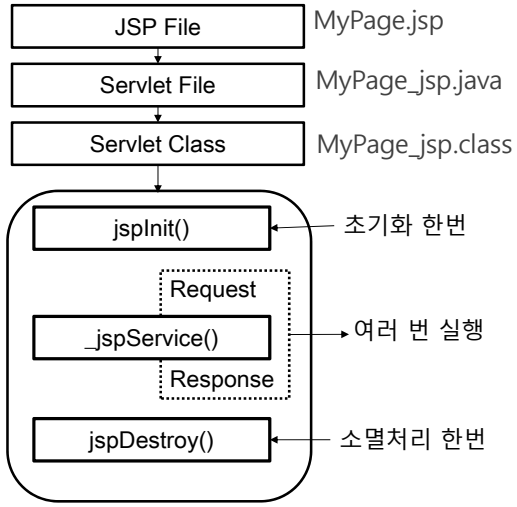


JSP 동작 원리



JSP Lifecycle

- JSP 생명주기
  - JSP Page Translation
  - JSP Page Compilation
  - Class Loading
  - Initialization
  - Execution
  - Cleanup

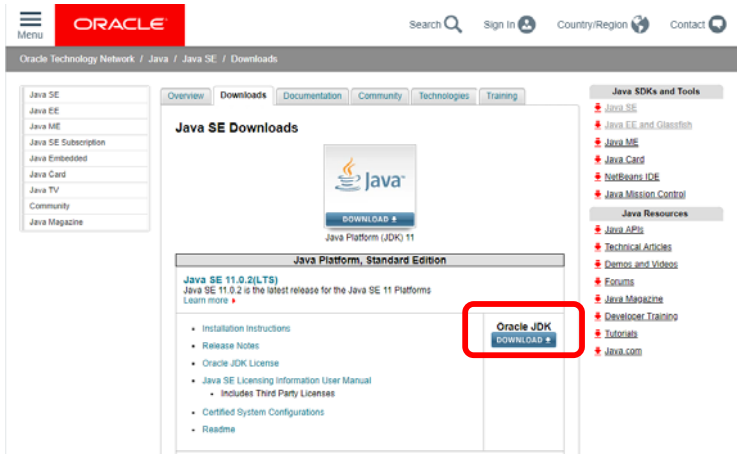


JSP 개발 환경 셋팅

- JDK 설치 : 자바 개발 도구
  - 환경변수 **JAVA_HOME** 설정
- 웹컨테이너 설치 : 톰캣, 제티 등
  - 환경변수 **CATALINA_HOME** 설정 (톰캣 설치 디렉토리)
- IDE 설치 : 이클립스 등
  - **elipse-workspace-jsp** 이클립스 작업 디렉토리 설정

JDK 설치

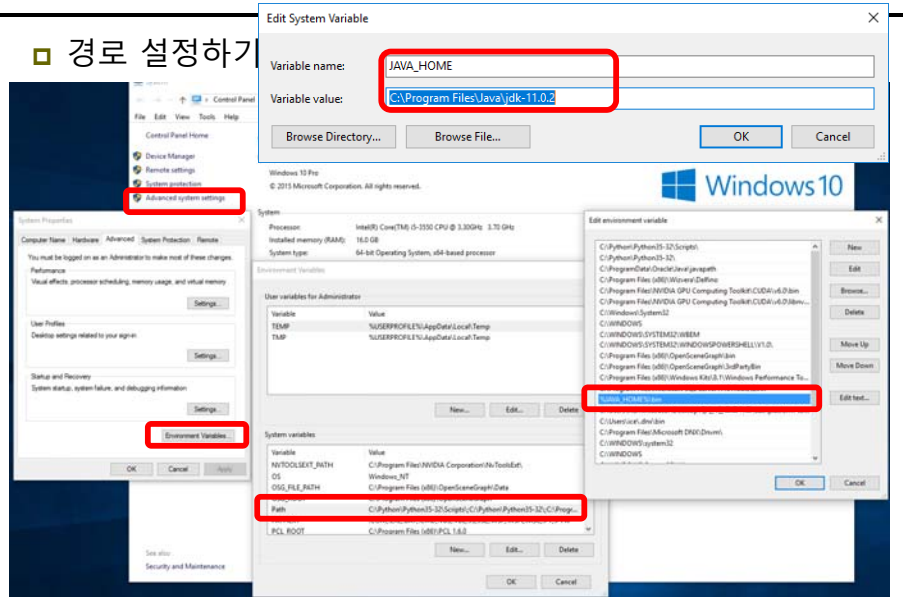
- Java SE Development Kit (JDK11.0.2) Download
  - http://www.oracle.com/technetwork/java/javase/downloads/index.html



JDK 설치

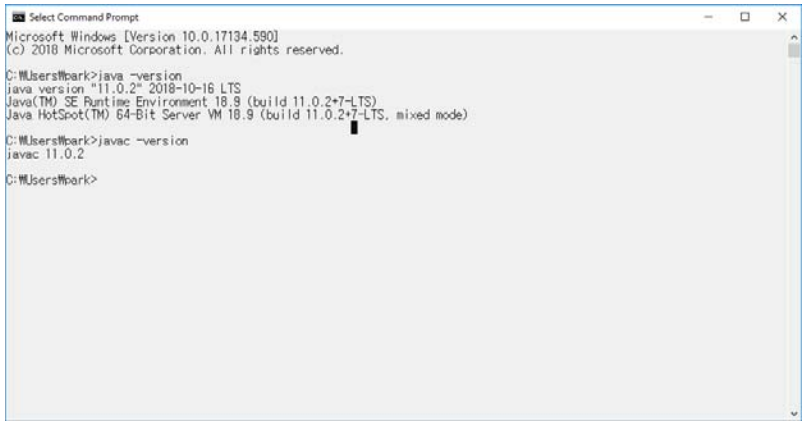
JDK 설치

- 경로 설정하기



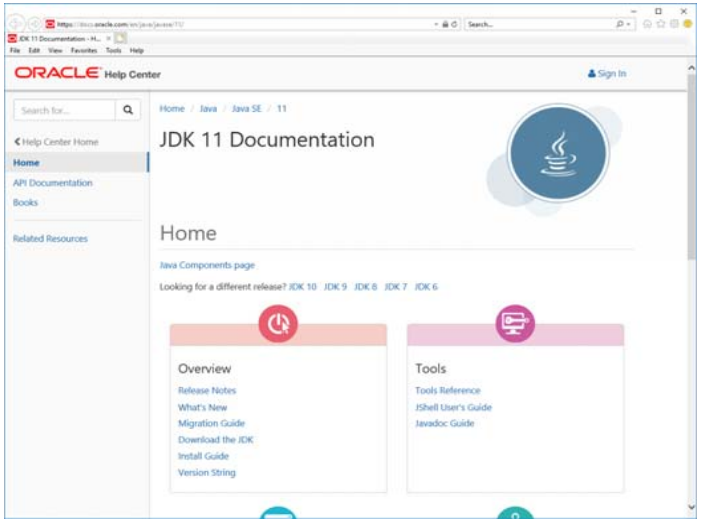
JDK 설치

- 설치된 자바 버전 확인



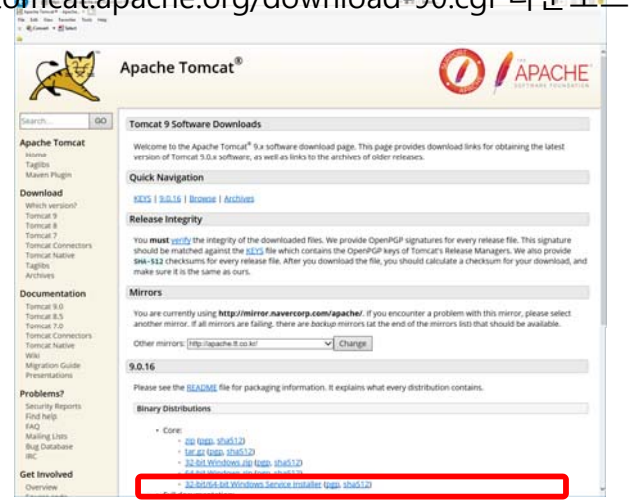
자바 API 문서

- https://docs.oracle.com/en/java/javase/11/



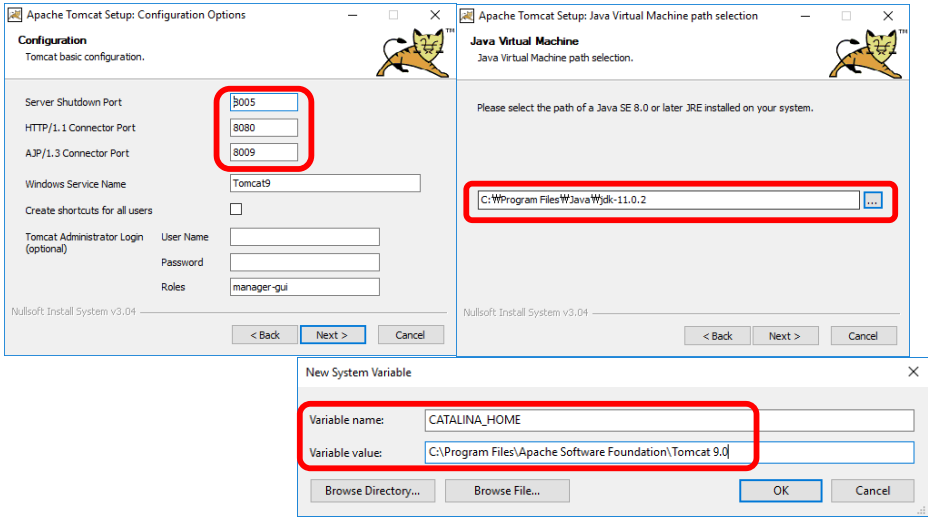
Tomcat 설치하기

- Tomcat9 (apache-tomcat-9.0.16.exe)
  https://tomcat.apache.org/download-90.cgi 다운로드



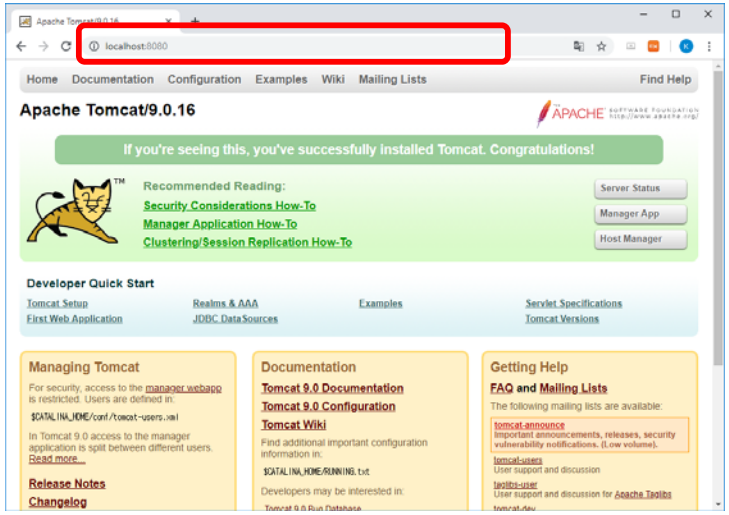
Tomcat 설치하기

- 톰캣 설치하기 및 설치 후 환경변수 설정



Tomcat 실행

- 톰캣 실행하기

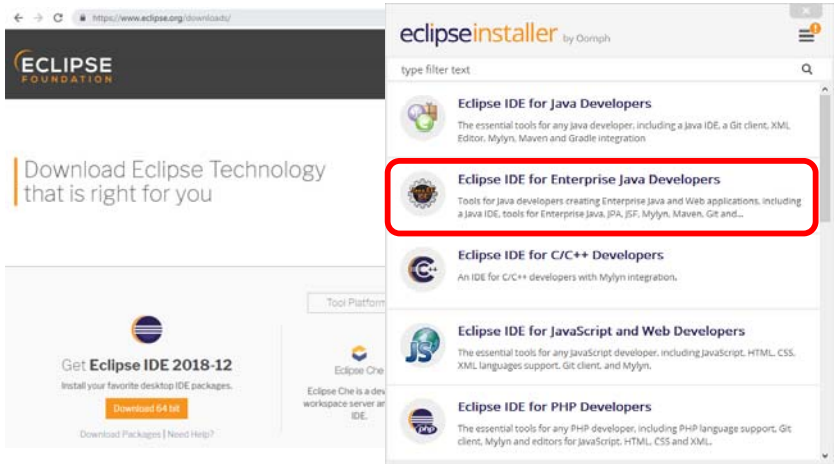


자바 IDE 소개와 설치

- 자바 IDE (Integrated Development Environment)
  - 통합 개발 환경
  - 편집, 컴파일, 디버깅을 한번에 할 수 있는 통합된 개발 환경
- 이클립스(Eclipse)
  - 자바 응용 프로그램 개발을 위한 통합 개발 환경
  - IBM에 의해 개발된 오픈 소스 프로젝트
  - http://www.eclipse.org/downloads/ 에서 다운로드
- 넷빈즈(Netbeans)
  - https://netbeans.org/
- IntelliJ Idea
  - http://www.jetbrains.com/idea/download/

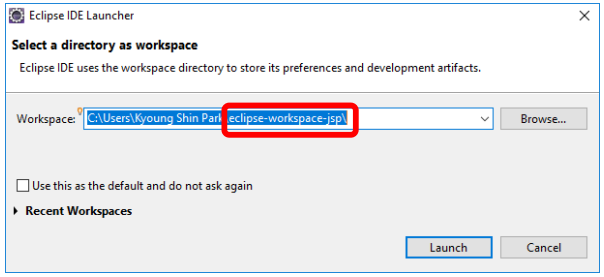
Eclipse 설치하기

- 이클립스 IDE 2018-12 (eclipse-inst-win64.exe) https://www.eclipse.org/downloads/ 다운로드



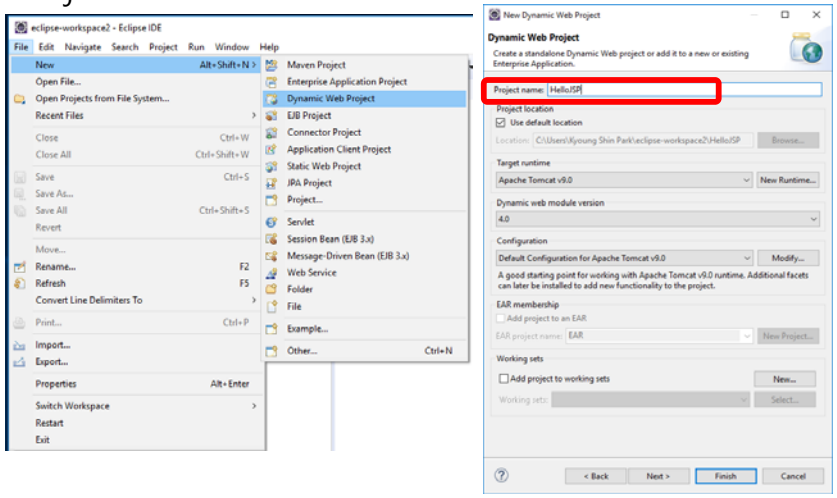
Eclipse 설치하기

- 이클립스 작업 디렉토리 설정하기



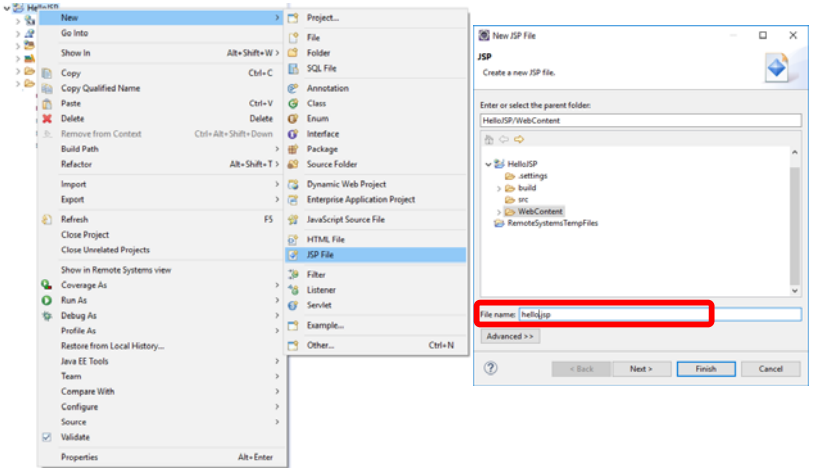
JSP 프로그램 작성

- 이클립스 프로젝트 생성 File->New->Dynamic Web Project



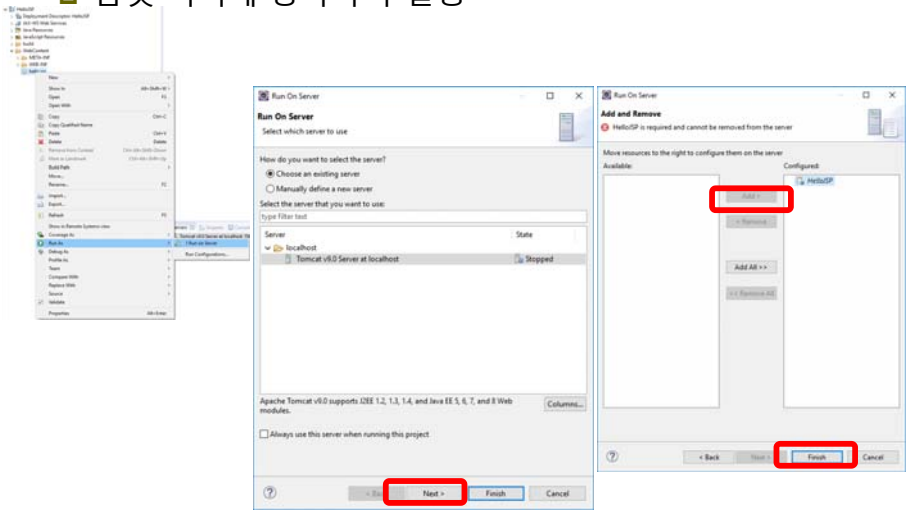
JSP 프로그램 작성

- 클래스에 소스 코드 추가



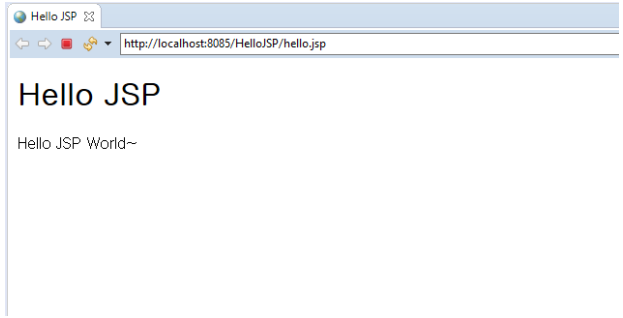
JSP 프로그램 실행

- 톰캣 서버에 등록하여 실행



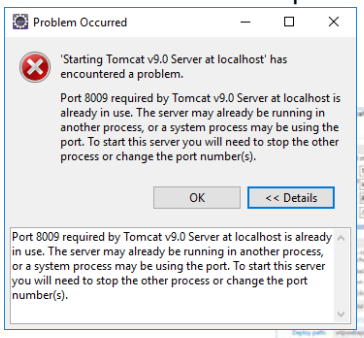
JSP 프로그램 실행

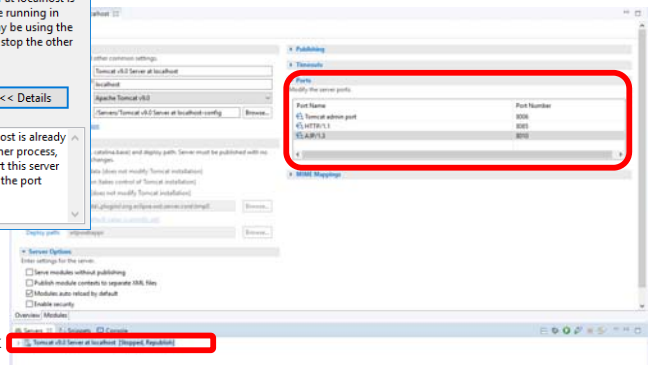
- 실행결과 확인하기



JSP 프로그램 실행

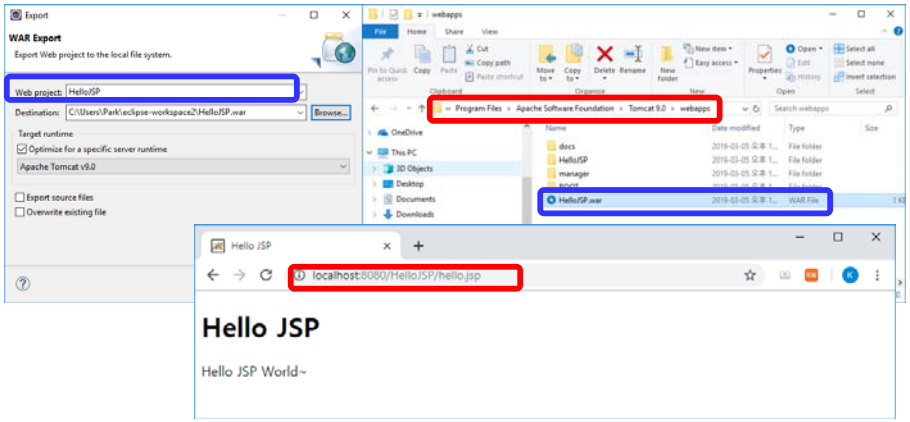
- 'Starting Tomcat v9.0 Server at localhost' has encountered a problem 에러가 발생 시





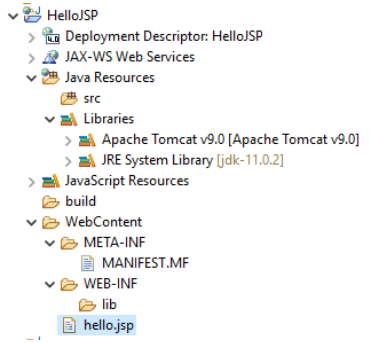
톰캣에 JSP 프로젝트 배포하기

1. JSP 프로젝트를 **war** 파일로 Export
2. **war** 파일을 톰캣의 웹루트(webapps)에 복사
3. http://localhost:포트/프로젝트명 접속

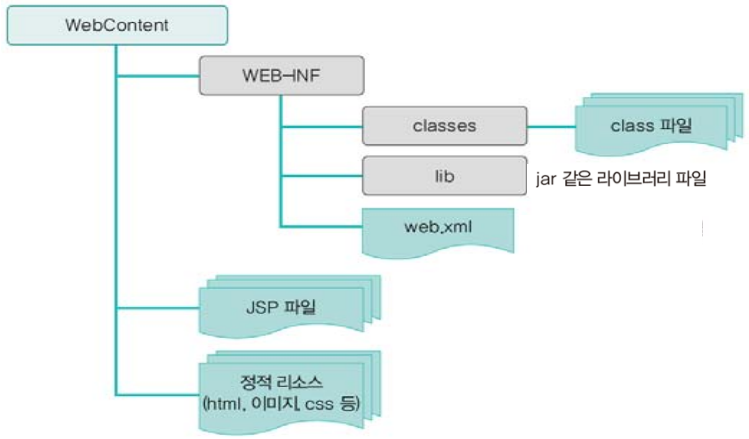


동적 웹 프로젝트 구조

- Dynamic Web Project 디렉토리 구조

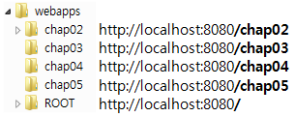


동적 웹 프로젝트 구조



동적 웹 프로젝트 구조

- 웹 어플리케이션 디렉터리 이름 → 컨텍스트 경로
- 컨텍스트 경로 → URL
  - 예, 컨텍스트 경로가 /chap02인 경우
    URL은 http://host:port/chap02로 매핑
- 톰캣의 경우 [톰캣]₩webapps 디렉터리에 어플리케이션 디렉터리 위치



LAB0

- HelloJSP 프로그램을 작성한다.
  - JDK 설치
  - IDE 설치
  - 톰캣 설치
  - 환경설정
  - 이클립스를 사용한 JSP 프로그램 작성
  - 프로젝트 디렉토리 안에 보고서 (1~2장)를 넣고 Lab0_학번_이름.zip 압축한 후 제출