

# Java Web Programming 입문

(Servlet/JSP)

19일차

# 오늘의 키워드

## ▶ Web

- 네트워크 (Network)
- 인터넷 (Internet)
- World Wide Web
- 프로토콜 (Protocol)
- HTTP (Hypertext Transfer Protocol)
- HTML (Hypertext Markup Language)
- 정적 웹 페이지 (Static Web Page)
- 동적 웹 페이지 (Dynamic Web Page)

## ▶ 서블릿 (Servlet)

## ▶ JSP (Java Server Page)

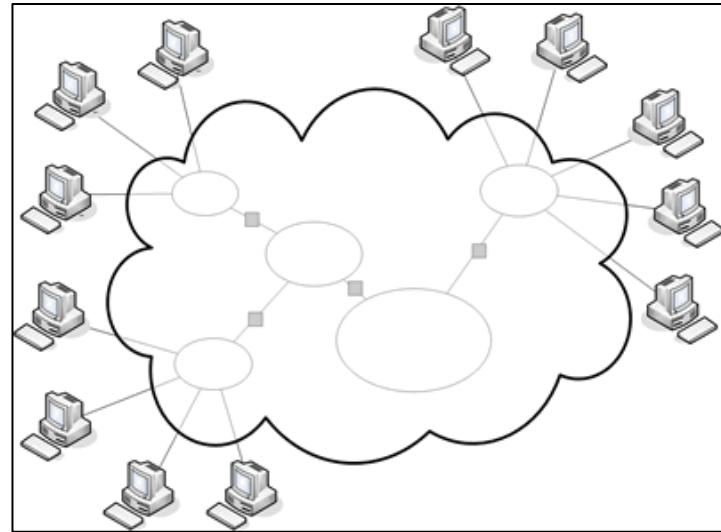
## ▶ 환경 Setting

- JDK 설치 (생략)
- Tomcat 설치
- Oracle 설치 (생략)

# Web

## ▶ 네트워크 (Network)

- Net(그물) + Work(작업)
- 복수의 컴퓨터가 통신/상호간 작업을 위해 연결되어 있는 통신구조
- 여러 컴퓨터들 간의 작업/자원 공유



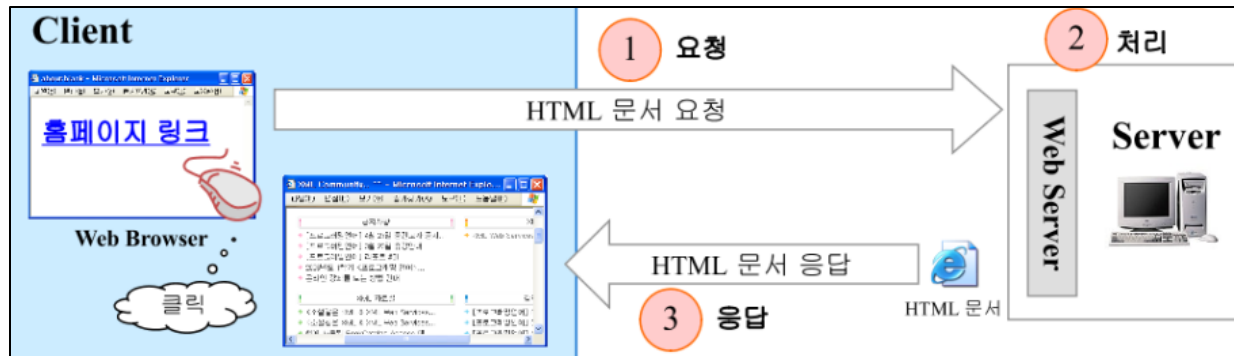
## ▶ 인터넷 (Internet)

- Inter(상호) + Network(네트워크)
- 복수의 Network가 서로 연결
- 전세계 Network가 하나로 연결
- 전 세계에 존재하는 Network들을 연결시키는 Network

# Web

## ▶ World Wide Web

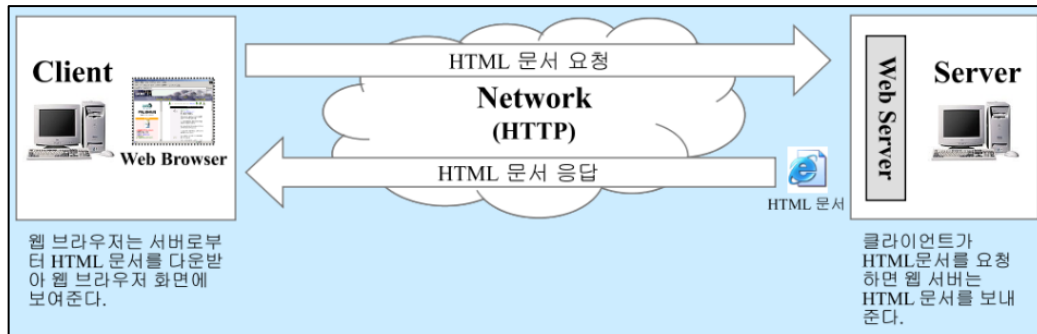
- WWW, Web
- Internet 자체가 아니라 Internet Service 중 하나 (가장 잘 나가는 Internet Service)
- FTP, Telnet 같은 Text 기반 Service 가 아닌 Multimedia service 제공
- Hyper Link 를 통해 정보 탐색이 쉬움 (강점)
  - Web 에 존재하는 문서들을 연결(Link)
  - Web-Browser 내 특정부분에 마우스 포인터가 손가락 모양으로 변하는 지점
    - 마우스로 클릭 시 연결된 문서가 열림
  - 가장 편리한 Internet Service로의 교두보



# Web

## ▶ World Wide Web

- HTML (Hypertext Markup Language) 언어를 이용하여 작성
- Web 동작
  - Client 는 Web-Browser 를 통해 Server에 정보를 요청
  - Server 는 Client 요청에 맞는 HTML 문서는 Client로 전송
  - Client 의 Web-Browser 는 Server에서 전송해준 HTML 문서를 해석, 사용자에게 보여줌



- 구성요소
  - Web Server
    - Web Client의 Request(요청)에 따라 HTML 문서를 제공 (Response)
  - Web Client
    - 필요한 정보 (HTML 문서) 를 Web Server에 Request(요청)
  - Web Browser
    - Web Server에서 제공된 HTML 문서를 해석 후 사용자에게 결과를 보여주는 Program

# Web

## ▶ 프로토콜 (Protocol)

□ **protocol** [ˈprɒtəˌkɔːl; -kɑːl, ˈprəʊtəkəl]

1. [명사]U 외교 의례, 의전

a breach of protocol ❖ 외교 의례 위반

the protocol of diplomatic visits ❖ 외교 방문상의 의전

2. [명사]C(전문 용어) (조약의) 초안[원안]; (합의안·조약의) 보충 협약

the first Geneva Protocol ❖ 제네바 조약 초안

It is set out in a legally binding protocol which forms part of the treaty. ❖ 그것은 그 조약의 일부를 이루는, 법적 구속력이 있는 보충 협약으로 만들어진다.

3. [명사]C(컴퓨터) 프로토콜, 통신 규약

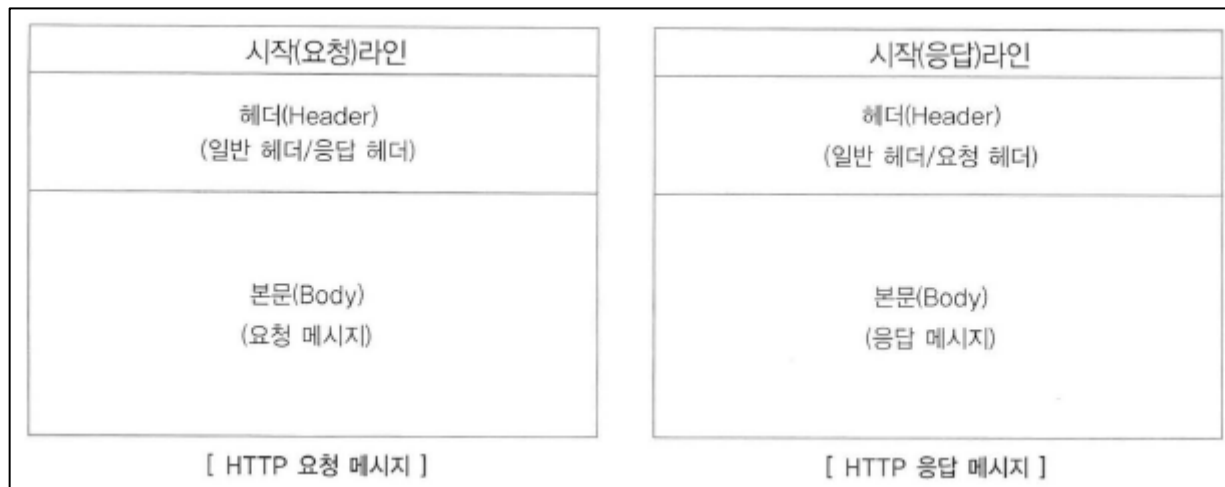
4. [명사]C(전문 용어) (과학 실험·의료 치료의) 계획서

◦ 컴퓨터들간의 정보 교환을 위한 약속

- 내가 '0110' 이후에 '0101' 이라고 보내면 이건 Boolean 타입의 'TRUE' 야!
- How? : 어떻게 상호간에 통신을 할 것이냐?
- What? : 어떤 내용으로 통신 할 것이냐?
- When? : 언제 통신 할 것이냐?

# Web

- ▶ HTTP (Hypertext Transfer Protocol)
  - Web Server 와 Web Browser 간의 통신 규약
    - HTML 문서, 파일 등의 정보 교환을 위한 Protocol
  - Web Browser 을 통해 Homepage 접속 시 사용하는 'http://' 의미



# Web

## ▶ HTML (Hypertext Markup Language)

- Web 에서 사용되는 Web Page 제작용 언어
- 이 언어를 Web Browser가 해석 후 우리가 보는 화면으로 표현
- Tag라는 기호를 이용, Hyper Link 및 Multimedia 같은 정보를 표현

	형 식	예 문
1	〈태그〉 문장 〈/태그〉	〈h1〉 테스트 〈/h1〉
2	〈태그 속성="값"〉 문장 〈/태그〉	〈a href="a.html"〉 테스트 〈/a〉
3	〈태그〉 or 〈태그/〉	〈br〉 or 〈br/〉

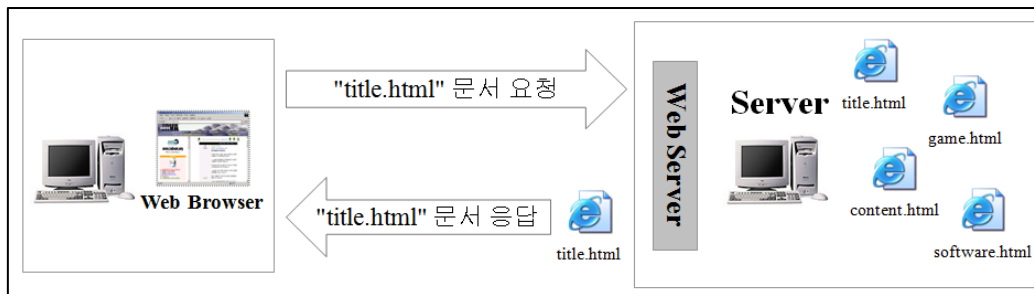
<b>형식</b>	<b>&lt;html&gt;</b>	: HTML 문서의 시작
	<b>&lt;head&gt;</b>	: HTML 문서의 헤더 부분 시작
	헤더 정보	
	<b>&lt;/head&gt;</b>	: HTML 문서의 헤더 부분 끝
	<b>&lt;body&gt;</b>	: HTML 문서의 바디 부분 시작
	본문 내용	
	<b>&lt;/body&gt;</b>	: HTML 문서의 바디 부분 끝
	<b>&lt;/html&gt;</b>	: HTML 문서의 종료



# Web

## ▶ 정적 웹 페이지 (Static Web Page)

- Client에게 제공하기 위해 미리 작성되어 있는 HTML
- 정보를 제공할 HTML 문서만 미리 만들어져 있는 상태
- Client의 Request에 따라 미리 작성된 HTML 을 그대로 제공



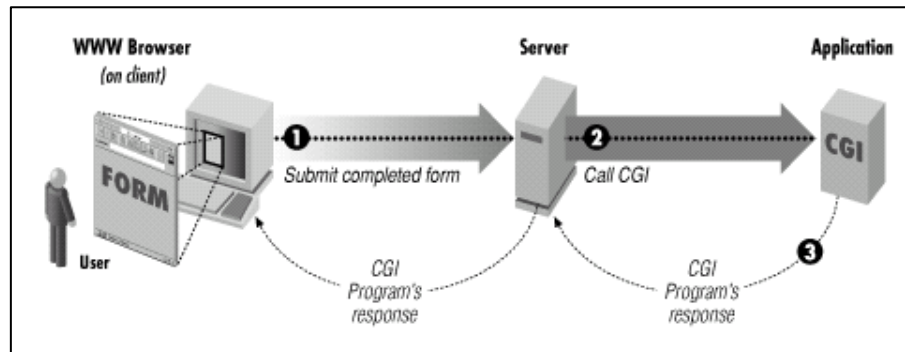
## ▶ 동적 웹 페이지 (Dynamic Web Page)

- Programming을 통해 Client의 Request에 따라 Dynamic 하게 작성되어 제공
- 실시간으로 작업 처리, Dynamic하게 생성된 문서를 Client에게 제공
- Server에서 Client가 원하는 작업을 처리후, 결과만을 Service
- Dynamic Web Page 생성을 위한 Programming Model이 CGI 방식

# Web

## ▶ CGI (Common Gateway Interface)

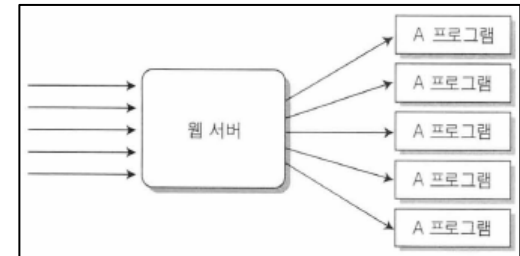
- Web Server 와 외부 Program 간의 정보를 주고받는 방식, 규약, 약속
- Web + Programming
- ASP, PHP, Perl, JSP 등 모두 CGI 범주에 들어간다고 볼 수 있음
- 단순 HTML 방식으로는 Dynamic Web Page 작성이 어려움
  - Dynamic 하게 Data를 처리하여 Client 에 HTML 문서를 전송
  - 이를 위해 Programming 적 필요성이 나타남
- CGI 동작
  - Web Server로 전송된 Client의 Request 로 CGI Program 호출
  - CGI Program이 Request에 따라 동적으로 HTML 생성
  - 생성된 HTML 문서(Response)를 Web Server를 통해 Client 로 전송



# Web

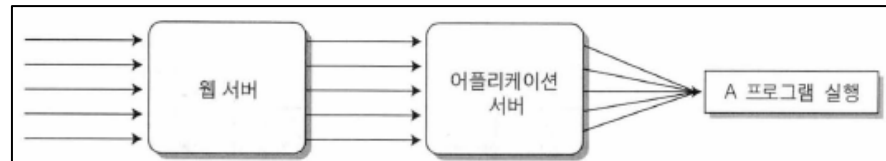
## ▶ CGI 의 한계

- Process 기반?
- Client의 Request 수 만큼 Process를 생성하여 처리
- 10개의 Client가 동일 Request를 보내어도 10개의 Process 생성 후 서비스



## ▶ 확장 CGI

- 하나의 Process가 여러 개의 Thread를 생성하여 Client의 Request 처리
- 동작
  - 처음 Client의 Request 일 경우 해당 Library 를 Memory에 Loading 후 처리
  - 다른 Client의 동일한 Request 일 경우 Memory에 있는 Library를 이용 후 처리



## ▶ Java 를 쓰는 확장 CGI ?

- Servlet
  - Java 기반의 확장 CGI Programming Language
- JSP (Java Server Page)
  - Java 를 이용하여 작성하는 확장 CGI 기반 Script Language

# Servlet

## ▶ Servlet

- Server + Let

- Server에서 Program 사용을 ‘허(Let)’ 하노라.
- Client 의 Request 에 따라 Servlet을 실행 후 결과를 Client에 전송

- Web 에서 사용 가능한 Java Program 을 구현하기 위한 Programming 방법
- Java 기반의 확장 CGI Application 을 구현하기 위한 언어

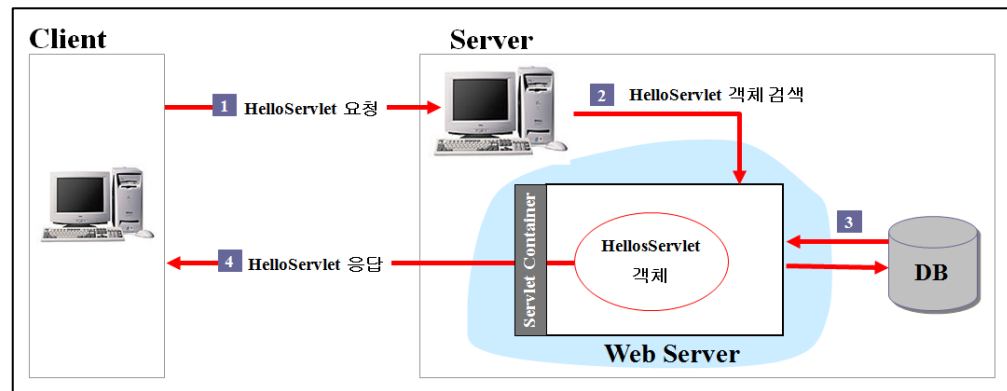
## ▶ 필수품

- JDK (Java Development Kit)
- Servlet Container
  - Tomcat, Web Sphere 등

# Servlet

## ▶ Servlet Container

- Servlet 을 실행하기 위해 담아두는 용기
  - Servlet Object 를 만들어 보관
  - Servlet 을 관리, 서비스
  - 우리 실습용으로는 톰캣(Tomcat) 사용
- 
- 역할
    - Web Server 기능을 가지므로 Web Server 역할이 가능
    - Compile 된 Servlet Object 를 보관
    - Client 의 Request 에 따라 Servlet 실행하여 Service 제공, 관리



# Servlet

## ▶ 특징

- Java 를 사용하여 작성
  - 기본적인 Java 에 대한 사용법을 알고 있어야 함
- Dynamic Web Application 구현이 가능
- Process 기반 CGI 가 아니라 Thread 형태로 동작

## ▶ 단점

- 최종적으로 출력될 정적 HTML 문서가 Servlet 내에 작성됨
- 디자인(Design) 영역과 비즈니스 로직 (Business Logic) 이 한 코드내에 혼재
- Java 파일이므로 Service 되기 전 .class 파일로 Compile이 선행되어야 함
- 이러한 단점을 극복하여 좀 더 쉽게 Web Application을 구현하기 위해 사용하는 방법이 JSP

# Servlet

## ▶ Servlet 예제

```
import javax.servlet.*;
import javax.servlet. http*;
import java.io*;
import java.util I.Calendar;

public class ServletTest extends HttpServlet{
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException{
        response.setContentType("text/html");
        response.setContentType("text/html");

        Calendar c = Calendar.getInstance();
        int hour = c.get(Calendar.HOUR_OF_DAY);
        int minute = c.get(Calendar.MINUTE);
        int second = c.get(Calendar.SECOND);

        PrintWriter out=response.getWriter();

        out.write("<HTML><HEAD><TITLE>ServletTest</TITLE></HEAD>");
        out.write("<BODY><H1>");
        out.write("현재시각은 ");
        out.write(Integer.toString(hour));
        out.write("시 ");
        out.write(Integer.toString(minute));
        out.write("분 ");
        out.write(Integer.toString(second));
        out.write("초입니다 ");
        out.write("</H1></BODY></HTML>");
        out.close();
    }
}
```

# JSP (Java Server Page)

## ▶ JSP (Java Server Page)

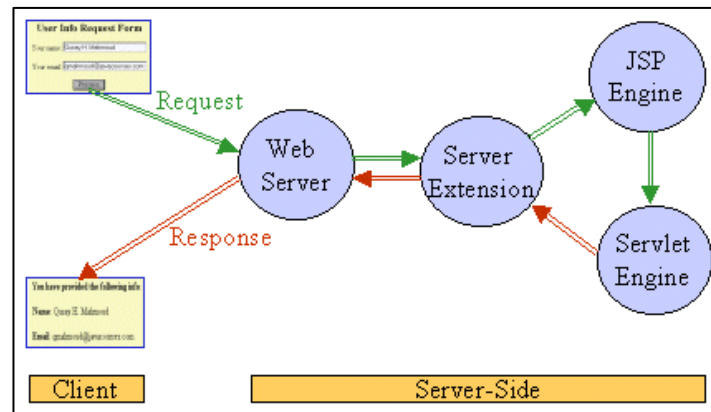
- Servlet 기반의 Dynamic Web Application 구현용 언어
- Web Application 제작 작업에 사용되는 Script Language
  - JSP 로 작성된 Code는 실행시 Servlet으로 자동 변환 후 Compile 되고 나서 실행
  - Server Side Script Language
  - Web + Java + Script
- Java 기반 Dynamic Web Application 을 구현하기 위한 Script Language
- Servlet의 복잡성을 해소하기 위해
  - Programmer 에게 편의성을 제공하기 위해 등장
- Design 영역과 Program 영역 분리가 가능
- Servlet은 .class Compile을 직접 해야 하지만,
  - JSP 는 Code 작성후 실행 자도 변환, Compile
- Servlet 이 사용되지 않는 것은 아님



# JSP (Java Server Page)

## ▶ 동작

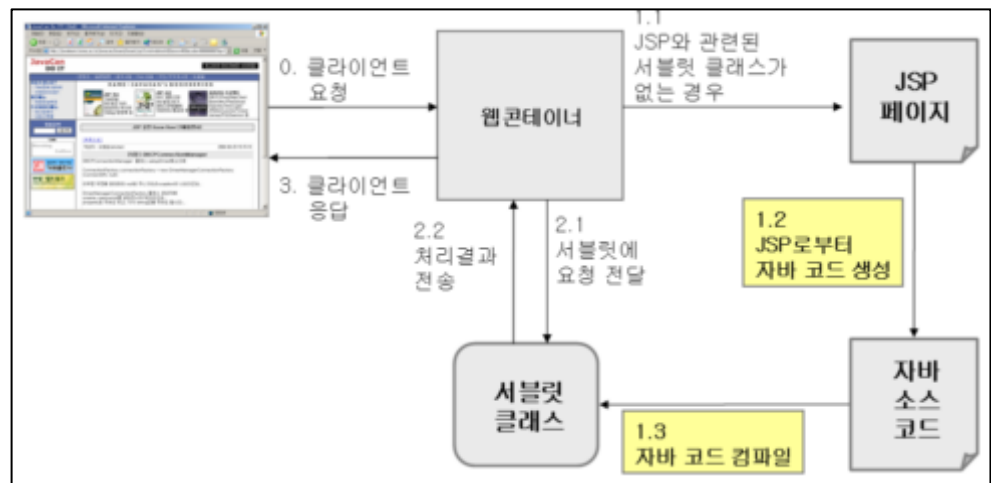
- Client의 Request 전송
- Servlet Container 내에 전송된 해당 Servlet Object 가 있는지 검사
- Servlet Object 가 없다면
  - JSP Code를 Servlet Code 로 변환 후 Compile 하여 Servlet Object 생성
- Servlet Object 가 있다면(변경도 없었다면) 그대로 사용
- Client 의 Request 에 대한 작업 처리후 Service



# JSP (Java Server Page)

## ▶ JSP Compile

- Client 의 Request 도착
- JSP Container Process 를 거치며 Servlet Object 의 존재(변경) 유무 검사
- Servlet Object 가 존재하지 않거나, 변경된 적이 있다면
  - JSP 파일을 Servlet Code로 변환 후 Compile
  - Compile 된 Servlet Object 를 Servlet Container에 보관
- Servlet Object 가 이미 Servlet Container에 Loading 되어 있고
  - JSP Code 의 변경도 없었다면
  - Servlet Container 에 보관되어 있는 Servlet Object 를 이용

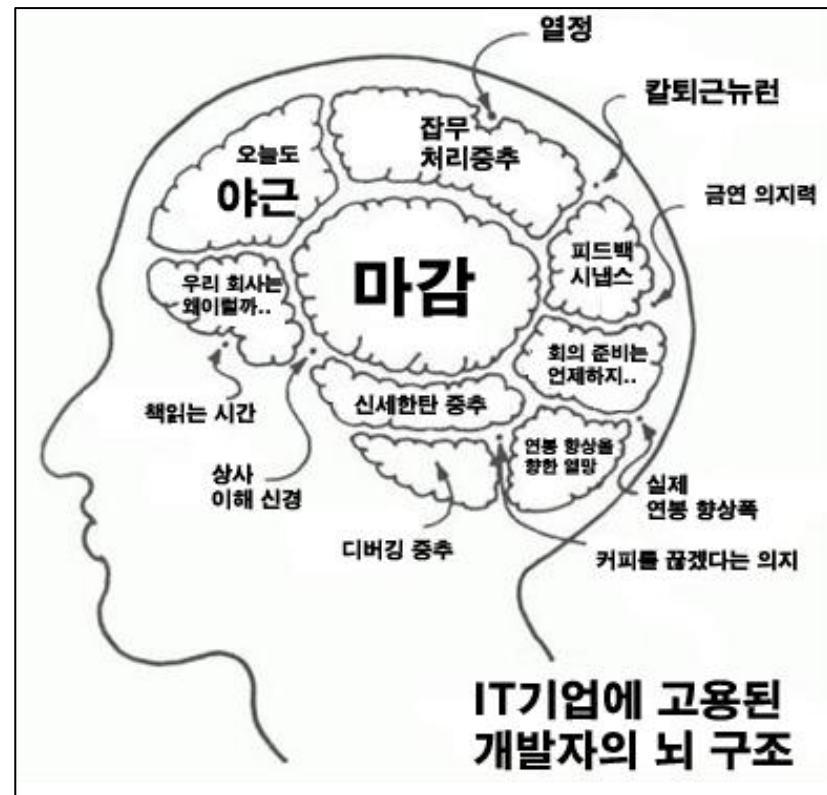


# 환경설정

## ▶ JDK (J2SE Development Kit)

○ ...

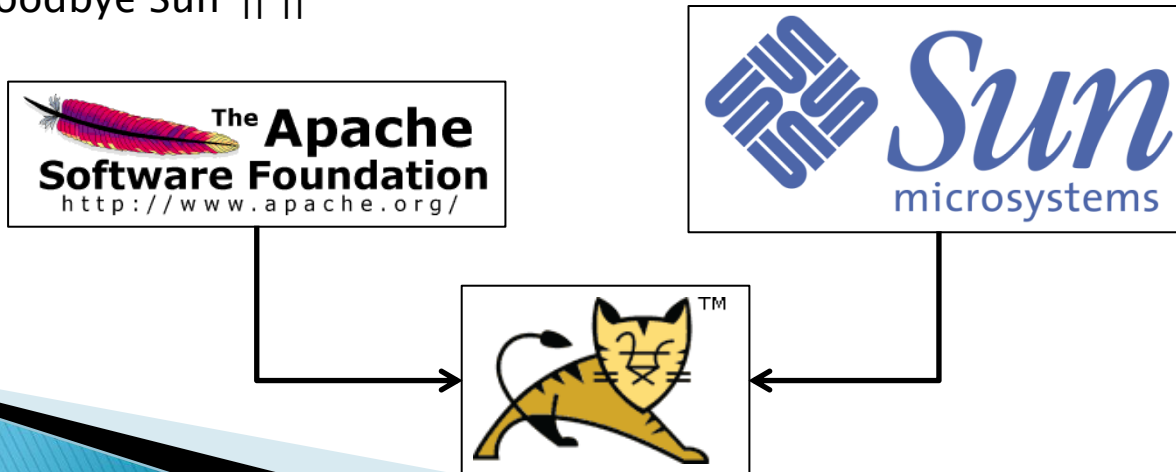
○ 생략 ㅎㅎ



# 환경설정

## ▶ Tomcat

- Servlet, JSP를 실행시키기 위한 Servlet Container
  - Servlet, JSP 같은 Java Code를 해석하여 보관
  - Client 의 Request에 맞는 Servlet Object를 사용하여 Service
- Web Server 기능을 포함하고 있음
- Servlet Container 는 무수히 많음
  - 가볍고, 사용이 편리해서 실습용으로 참 좋아요~
- Sun Microsystems, Apache Software Foundation이 Jakarta Project를 통해 공동으로 개발한 Servlet Container
  - Goodbye Sun TT TT



# 환경설정

- ▶ Tomcat 설치 (수정중)
  - <http://tomcat.apache.org/>

## Download

- [Which version?](#)
- [Tomcat 7.0](#)
- [Tomcat 6.0](#)
- [Tomcat 5.5](#)
- [Tomcat Connectors](#)
- [Tomcat Native](#)
- [Archives](#)

## Binary Distributions

- Core:
  - [zip \(pgp, md5\)](#)
  - [tar.gz \(pgp, md5\)](#)
  - [32-bit Windows zip \(pgp, md5\)](#)
  - [64-bit Windows zip \(pgp, md5\)](#)
  - [64-bit Itanium Windows zip \(pgp, md5\)](#)
  - [32-bit/64-bit Windows Service Installer \(pgp, md5\)](#)
- Deployer:
  - [zip \(pgp, md5\)](#)
  - [tar.gz \(pgp, md5\)](#)

## Source Code Distributions

- [tar.gz \(pgp, md5\)](#)
- [zip \(pgp, md5\)](#)

# 환경설정

## ▶ Oracle 설치

- ...
- 생략 ㅎㅎ



# 오늘 숙제

- ▶ Java + Tomcat + Eclipse
  - Hello JSP!