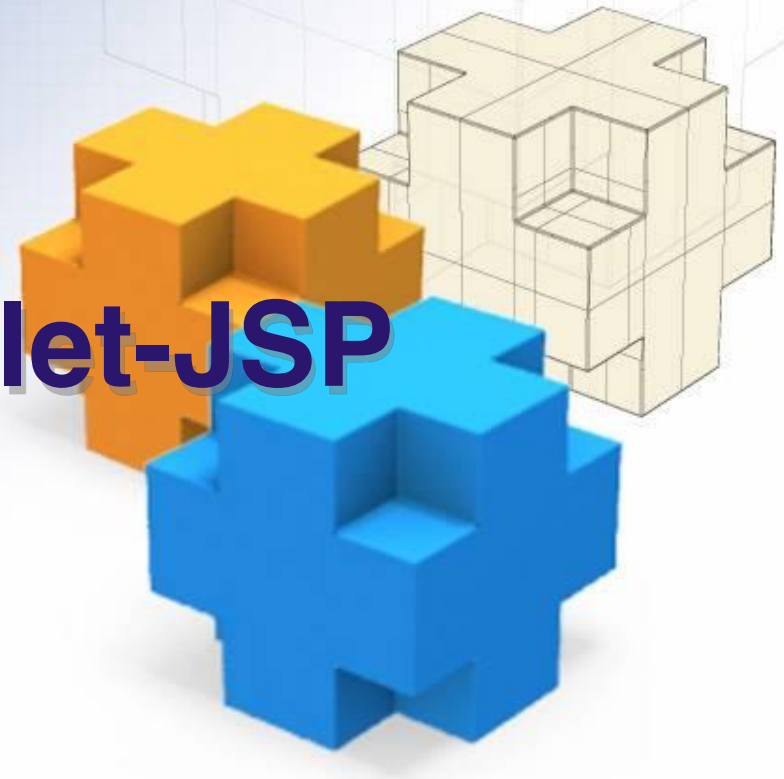
JSP Programming

# CGI-확장CGI-Servlet-JSP



# Conventional Web Architecture

## 1.1.7 전통적인 웹 구조의 한계

정적: 변하지않는 정보를 출력
동적: 변하는 정보를 출력
>요청이 있을때마다 변하는 시간출력이 예시임

정적인 웹 페이지는 개발자가 미리 작성하여 서버에 올려놓고 서비스하는 형태로 제공된다.

▣ 전통적 웹의 한계

◆ 정적인(Static) 웹 페이지만을 제공

– 이미 만들어져 있는 웹 페이지를 클라이언트의 요청에 따라 서비스하는 형태

– 고정된 웹 페이지의 제공만이 가능하다.


하지만, 동적인 웹 페이지는 프로그래밍을 통해 클라이언트의 요청에 따라 다양한 형태로 생성되어 제공된다.

▣ 해결책

◆ 동적 웹 페이지(Dynamic Web Page)

– 프로그래밍을 이용함으로써 요청이 들어 왔을 때 동적으로 웹 페이지를 생성하여 서비스를 제공

◆ CGI(Common Gateway Interface) 모델

# Common Gateway Interface (CGI)

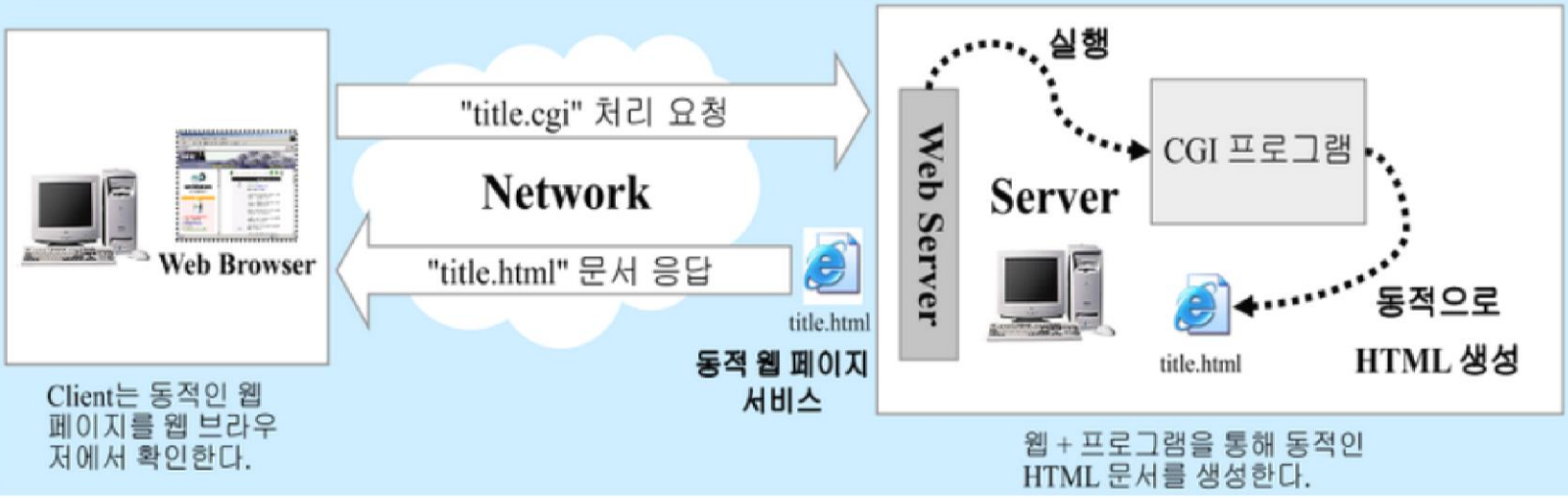
쓰레드란
〉 힙에있는 고정된 메모리만 공유하는것을 의미
찾아볼것

## CGI

- 정적인 HTML 문서 서비스의 한계 극복
- 서버-사이드(Server-Side) 스크립트 언어의 시초
- CGI = 웹 + 프로그래밍


## CGI 특징 (CGI 프로그래밍 예제)

- 프로그래밍을 이용해서 동적으로 생성된 웹 페이지를 클라이언트에 제공하는 모델



# 확장 CGI

동적웹프로그래밍을 위해 씨를 쓰냐 자바를 쓰냐의 차이
cgi 와 서브렛의 차이

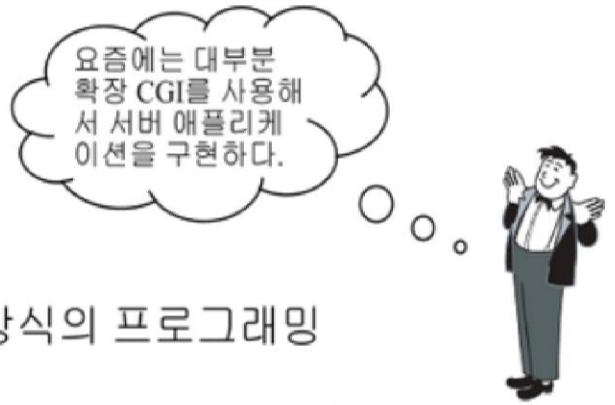
## CGI의 단점

- 클라이언트의 요청만큼 프로세서를 생성해서 처리하기 때문에 처리량에 한계가 있다.

## 확장 CGI

- 처음 클라이언트의 요청에 따라 라이브러리를 메모리에 로딩한 후, 메모리에 적재된 라이브러리를 사용해서 클라이언트의 요청을 처리하는 방식
- 다수의 요청이라도 한 개에 해당하는 메모리만을 사용하기 때문에 일반적인 CGI에 비해 효율적이다.

## 확장 CGI 언어




- 자바 기반의 확장 CGI 프로그래밍
  - 서블릿(Servlet)
    - 자바로 된 확장 CGI
    - 웹(HTTP) + 자바 프로그래밍
- 자바 기반의, 확장 CGI 기반의, 스크립트 방식의 프로그래밍
  - JSP(Java Server Pages)
    - 서블릿보다 쉽게 동적 웹 페이지를 생성할 수 있는 프로그래밍 언어
    - 웹(HTTP) + 프로그래밍(Java) + 스크립트(Script) 기능
    - 스크립트 방식의 서버 측 프로그래밍(Server Side Programming)

# Servlet

## 서블릿(Servlet)

- Server + Let(허용)의 합성어
- '서버(Server)에서 애플리케이션(Application)을 허용(Let)한다'라는 의미

## 서블릿의 구성

- 웹 + 자바 프로그래밍
- Java 프로그래밍 기반의 확장 CGI


## 서블릿을 실행하기 위한 필요 조건

- JDK(Java Development Kit) 설치 필요
- 서블릿 컨테이너(Servlet Container) 설치 필요
  - Sun Java System Web Server, TOMCAT, Resin

# Servlet Container

## 서블릿 컨테이너란?

- 서블릿 객체를 만들어 보관하는 곳
- 서블릿을 관리하고 서비스하는 프로그램

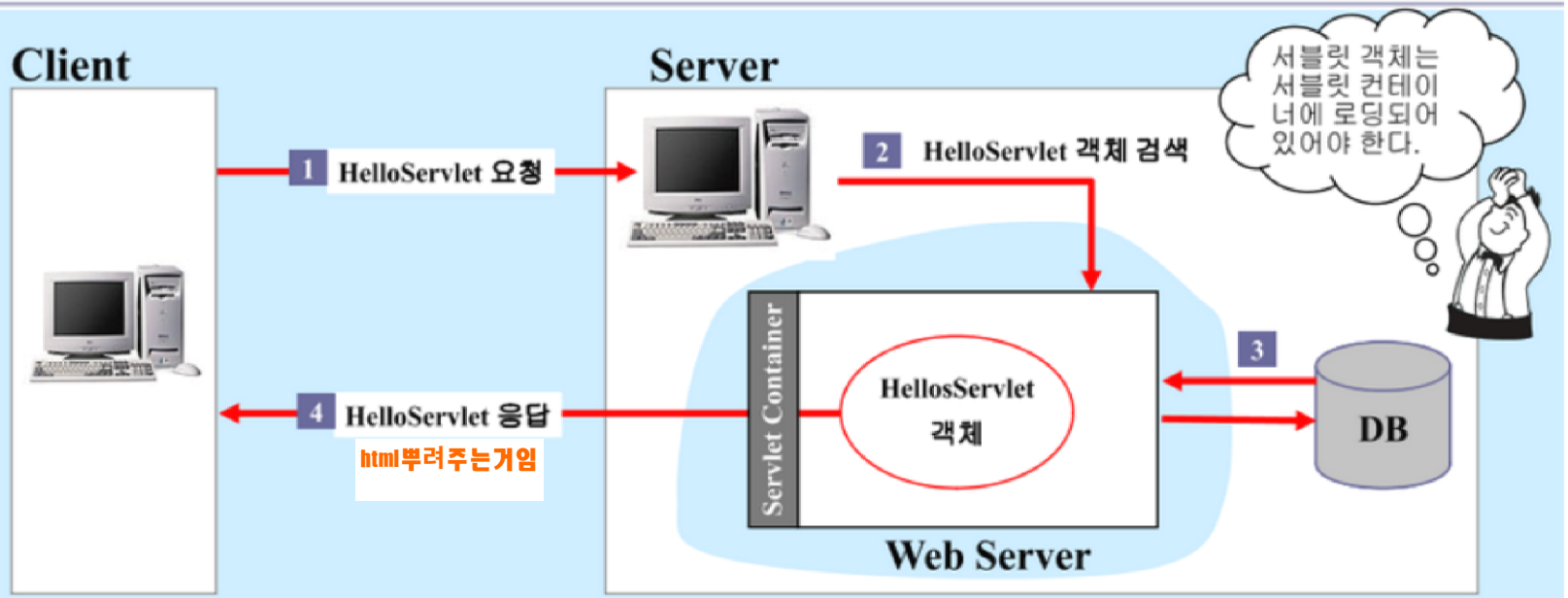

## 대표적인 서블릿 컨테이너

- 톰캣(TOMCAT), 레신(Resin) 등

## 서블릿 컨테이너의 역할

- 웹 서버 역할
- 자체적으로 웹 서버 기능이 있다.
- 서블릿을 담고 있다가 클라이언트의 요청에 따라 서비스하는 역할을 한다.

# Servlet의 요청과 처리




▣ 서블릿 실행 절차

1. 클라이언트로부터 HelloServlet 요청
2. 서블릿 컨테이너에서 해당 서블릿(HelloServlet) 객체 검색 (있다면 진행, 없다면 생성)
3. 해당 서블릿(HelloServlet)을 처리하기 위해 데이터베이스 작업이 필요하면 데이터베이스 연결 및 처리
4. HelloServlet 서블릿의 응답 전송

# Servlet의 특징 & 단점

## ▣ 서블릿의 특징

- ◆ 자바 언어를 사용
- ◆ 서블릿은 자바 프로그램 차원에서 동적으로 웹 페이지를 서비스한다.
- ◆ 스레드(Thread) 방식
  - 기존 CGI : 클라이언트 접속 → Process 생성
  - 서블릿 : 클라이언트 접속 → Thread 생성


## ▣ 서블릿의 단점

- ◆ 화면에 표현될 HTML 코드를 프로그램적으로 작성해야 한다.
- ◆ 서비스하기 전에 반드시 컴파일을 해야 한다.

# What is JSP?

## JSP(Java Server Page)

- 서블릿 기반 위에 보다 편리하게 웹 프로그래밍을 할 수 있도록 만든 동적 웹 페이지 작성 언어
- 웹 + 자바 프로그래밍 + 소스코드 자동 생성(스크립트)

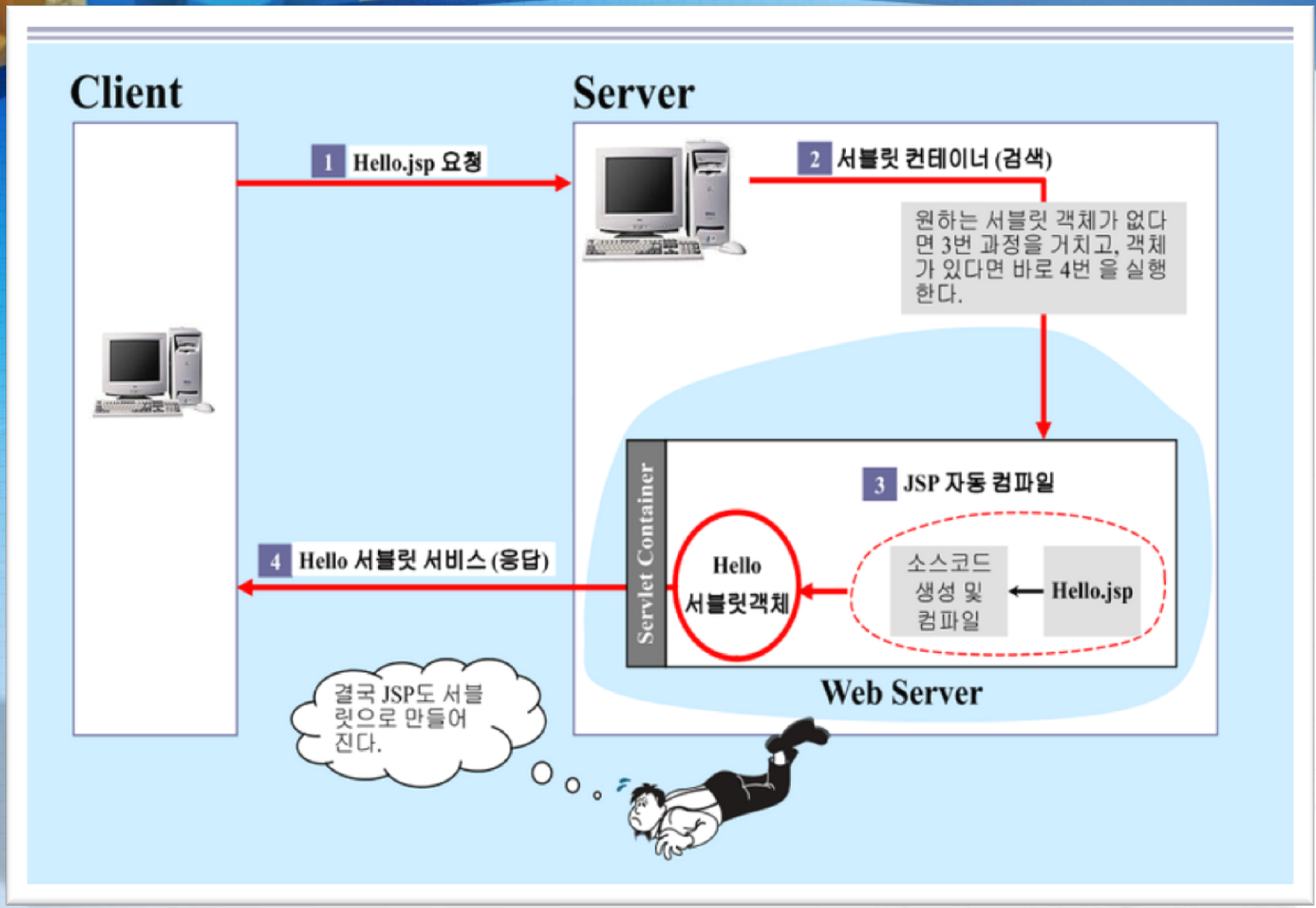
## JSP의 등장

- 자바 계열의 확장 CGI는 서블릿이다.
- 서블릿보다 편리하게 프로그래밍할 수 있다.
- JSP로 할 수 있는 것은 서블릿으로도 가능하다.

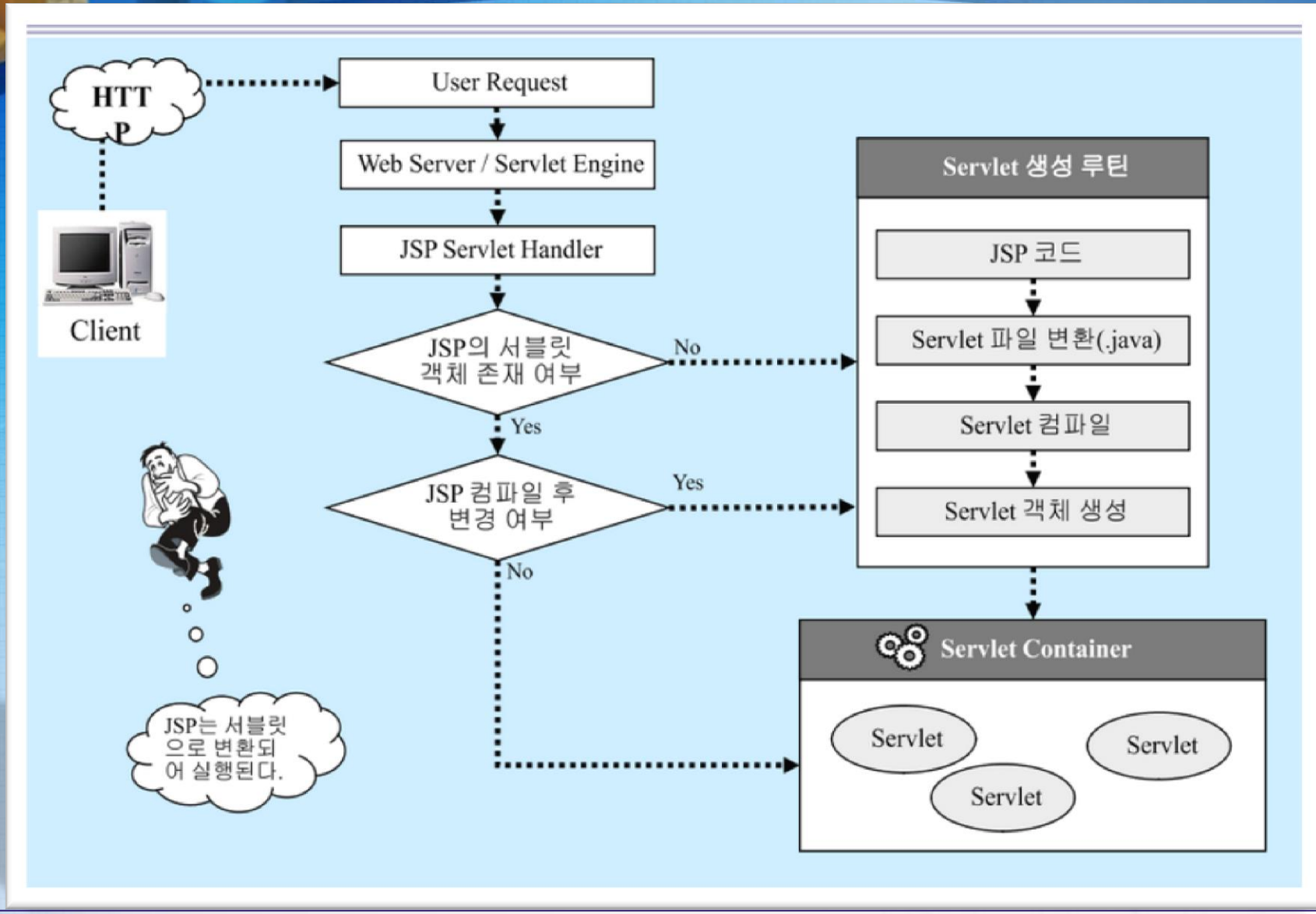

## 왜 서블릿 대신에 JSP를 사용하는가?

- 컴파일 하기 싫어서
- 필요한 부분에만 프로그램을 작성하기 위해서
- 결과적으로 프로그래머의 편의성을 위해서 JSP가 탄생

# JSP의 요청과 처리


Client
Server
1 Hello.jsp 요청
2 서블릿 컨테이너 (검색)
원하는 서블릿 객체가 없다면 3번 과정을 거치고, 객체가 있다면 바로 4번 을 실행한다.
3 JSP 자동 컴파일
Servlet Container
Hello
서블릿객체
소스코드
생성 및
컴파일
Hello.jsp
4 Hello 서블릿 서비스 (응답)
Web Server
결국 JSP도 서블릿으로 만들어진다.

# JSP의 실행과정 (1/2)


HTTP
Client
User Request
Web Server / Servlet Engine
JSP Servlet Handler
JSP의 서블릿 객체 존재 여부
No
Yes
JSP 컴파일 후 변경 여부
Yes
No
JSP는 서블릿으로 변환되어 실행된다.
Servlet 생성 루틴
JSP 코드
Servlet 파일 변환(.java)
Servlet 컴파일
Servlet 객체 생성
Servlet Container
Servlet
Servlet
Servlet

# JSP의 실행과정 (2/2)

## 1.3.3 JSP의 컴파일과 실행 과정

JSP는 JSP 코드를 만든 후 웹 서버에서 서비스가 이루어지는 디렉터리에 넣어 두기만 하면 나머지는 JSP 내부에서 전부 알아서 진행된다. JSP 페이지가 처음 요청을 받았을 때는 당연히 .class 파일로 컴파일이 된 후 서블릿 컨테이너에 적재되어 서비스가 이루어진다.

클라이언트에서 JSP를 요청했을 때 해당 JSP의 서블릿 객체가 서블릿 컨테이너에 있는지 검사하고, 만약 있다면 바로 실행을 하게 되지만 없다면 서블릿 생성 루틴을 거치게 된다. JSP는 JSP 파일이 수정되었는지를 검사하는 검사 루틴을 가지고 있기 때문에 JSP 파일이 수정되었는지 검사한 후 수정되었다면 다시 서블릿 생성 루틴을 거쳐 서블릿 컨테이너에 적재된다.

위 그림의 JSP 작업 흐름도를 자세히 살펴보면 제일 먼저 웹 서버나 서블릿 엔진이 클라이언트의 요청을 받게 된다. 서버에서 요청을 받게 되면 JSP Handler 즉, JSP 컨테이너 프로세스(JSP Container Process)라고 불리는 루틴을 거치면서 서블릿 객체의 존재 여부와 파일 변경 여부를 확인하게 된다. 만약 변경되었다면 새로 서블릿 소스 파일을 생성한 후 컴파일 과정을 거쳐 실행하게 된다. 만약 서블릿 컨테이너에 이미 로딩이 되어있고 변경되지 않았다면 서블릿 컨테이너에 적재되어 있는 서블릿 객체를 이용해서 서비스가 이루어진다.

□ **JSP의 컴파일과 실행 과정**

◇ 클라이언트의 요청 도착
◇ JSP 컨테이너 프로세스를 거치면서 서블릿 객체의 존재 유무와 변경 여부를 검사
◇ 서블릿 객체가 존재하지 않는다면 JSP 파일을 서블릿 코드로 변환하고 컴파일하여 서블릿 컨테이너에 적재
◇ JSP 파일이 변경되었다면 JSP 파일을 서블릿 코드로 변환하고 컴파일 하여 서블릿 컨테이너에 적재
◇ 서블릿 객체가 서블릿 컨테이너에 로딩되어 있고 JSP 코드가 변경되지 않았다면 서블릿 컨테이너에 적재되어 있는 서블릿 객체를 이용해서 서비스 수행