

JSP 프로그래밍

(재)대덕인재개발원

01 웹 프로그래밍 기초

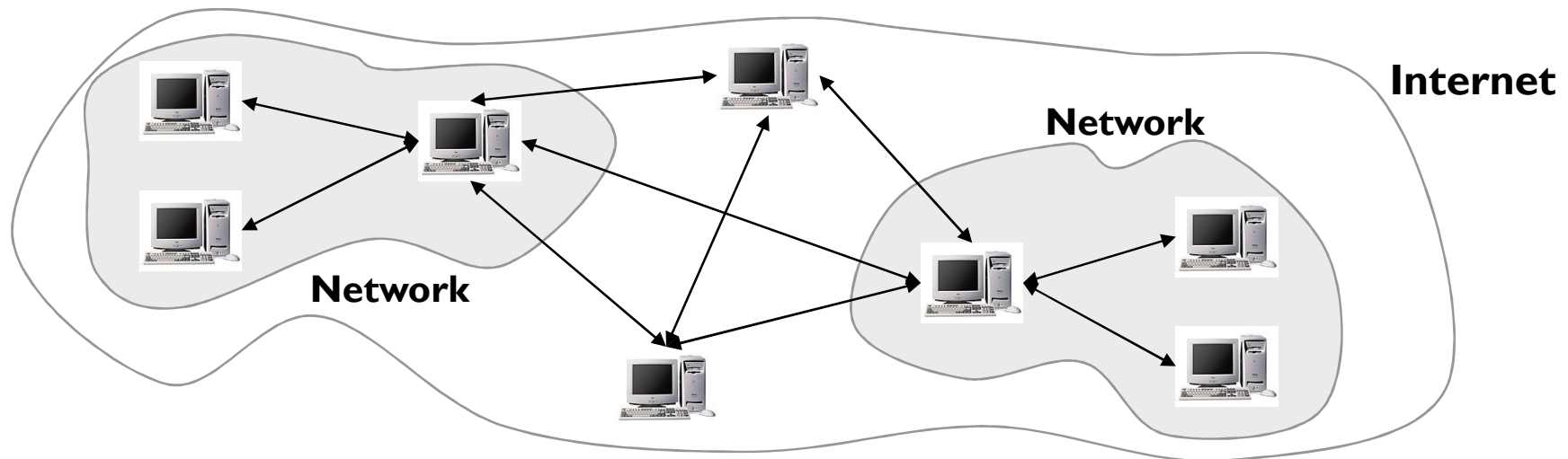


1.1 웹 (WEB)



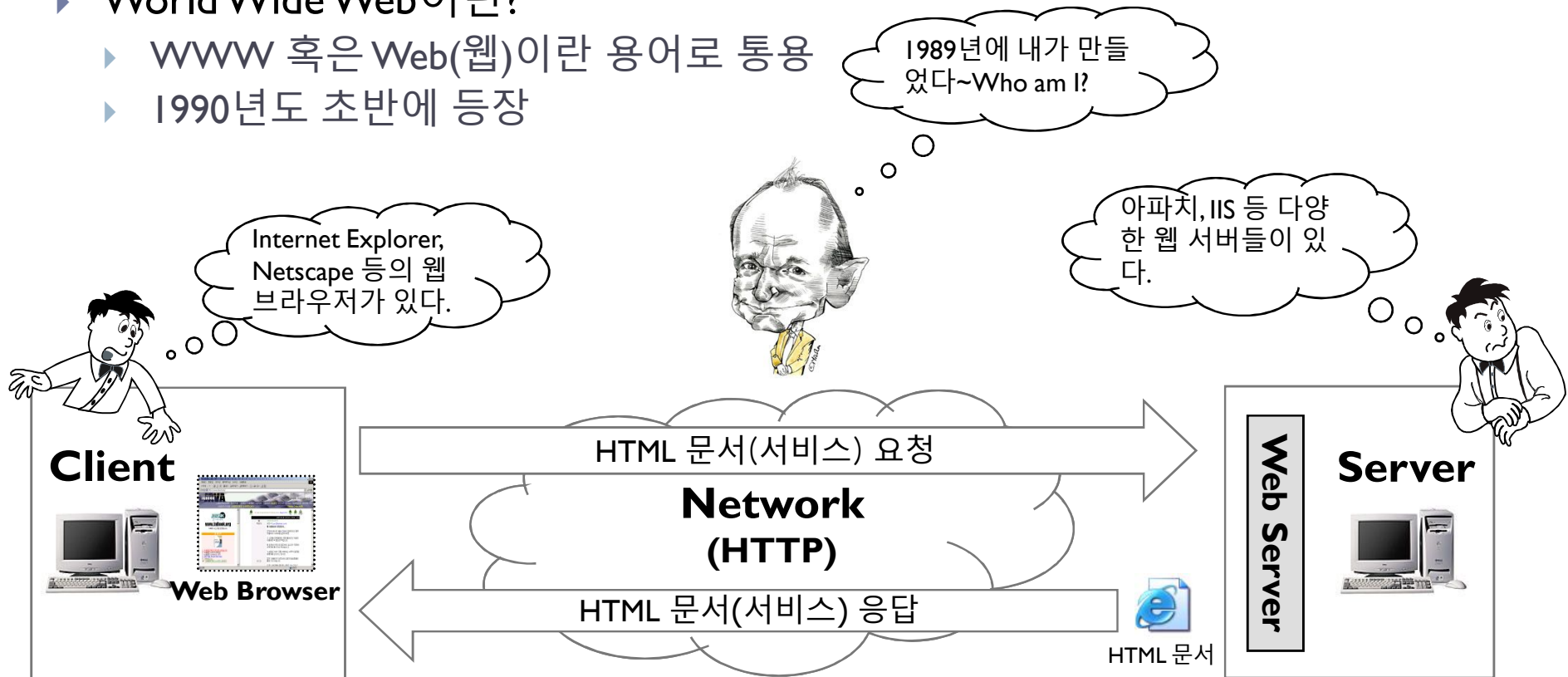
1.1.1 네트워크(Network)와 인터넷(Internet)

- ▶ 네트워크(Network)
 - ▶ 서로간의 통신을 위해 두 개 이상의 장치들이 연결되어 있는 통신 구조
- ▶ 인터넷(Internet)
 - ▶ 수많은 컴퓨터 네트워크들을 연결시키는 네트워크



1.1.2 월드 와이드 웹(World Wide Web)

- ▶ World Wide Web이란?
 - ▶ WWW 혹은 Web(웹)이란 용어로 통용
 - ▶ 1990년도 초반에 등장

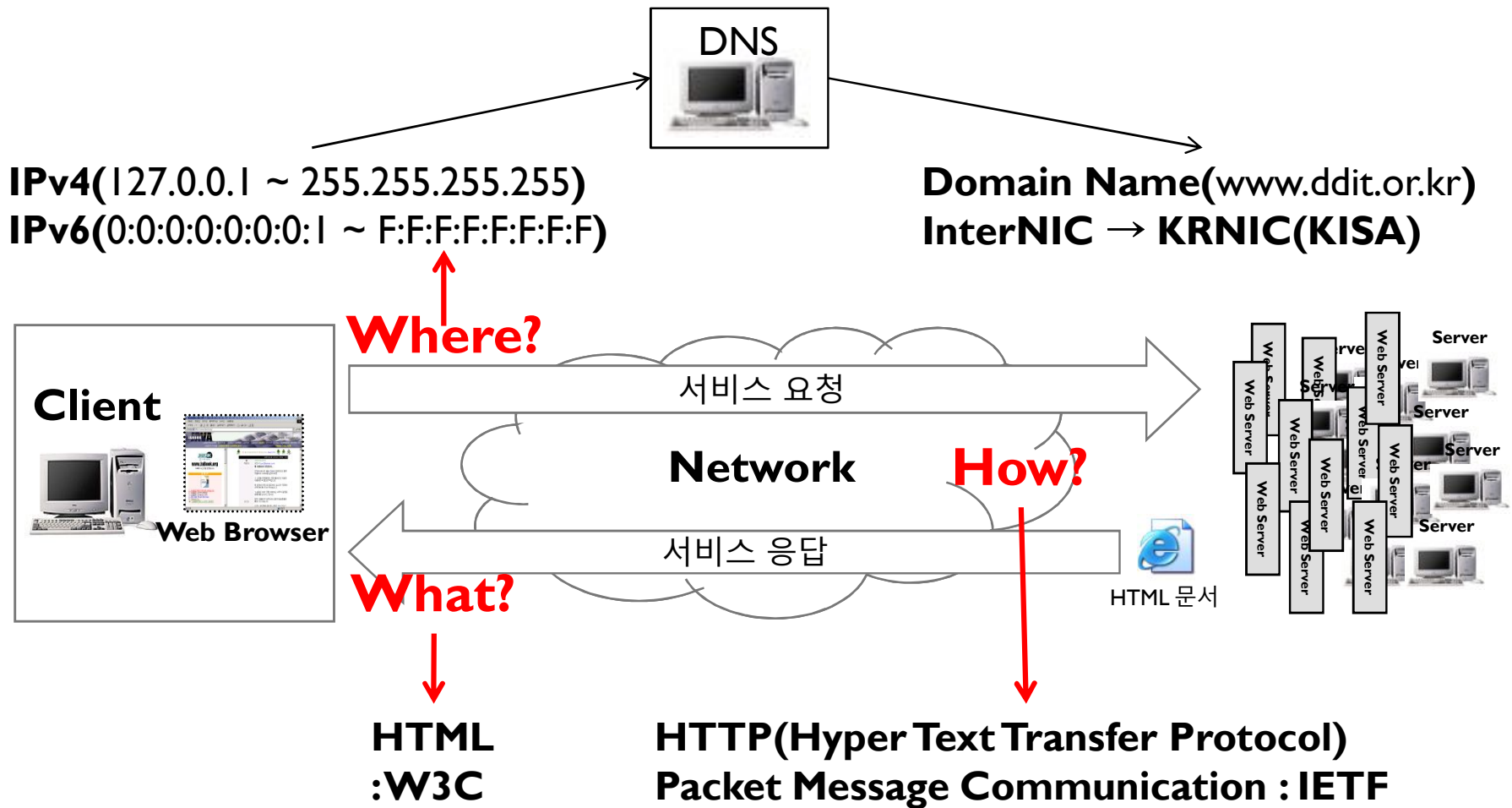


웹 브라우저는 서버로부터 HTML 문서를 다운받아 웹 브라우저 화면에 보여준다.

클라이언트가 HTML 문서를 요청하면 웹 서버는 HTML 문서를 보내준다.

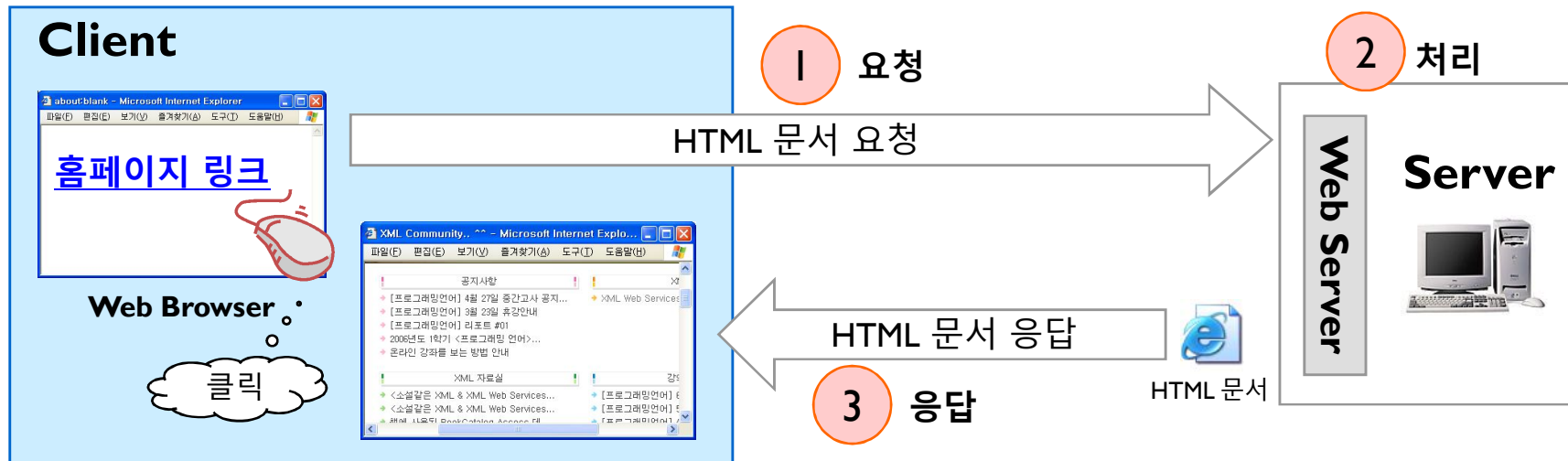
1.1.2 월드 와이드 웹(World Wide Web)

ICANN(Internet Cooperation for Assigned Names and Numbers)



1.1.3 웹의 주요 기능

- ▶ 하이퍼링크(Hyper Link)
 - ▶ 특정 단어나 이미지를 다른 문서와 연결



- ▶ 멀티미디어(Multimedia) 서비스
 - ▶ 음성과 정지 화상 및 동화상 등의 서비스를 제공

1.1.4 웹의 구성 요소

- ▶ 웹 서버(Web Server)

- ▶ 클라이언트의 요청을 받아서 처리한 후 결과를 클라이언트에 전송해 주는 주체

- ▶ 웹 클라이언트(Web Client)

- ▶ 필요한 데이터를 웹 서버에 요청하는 주체

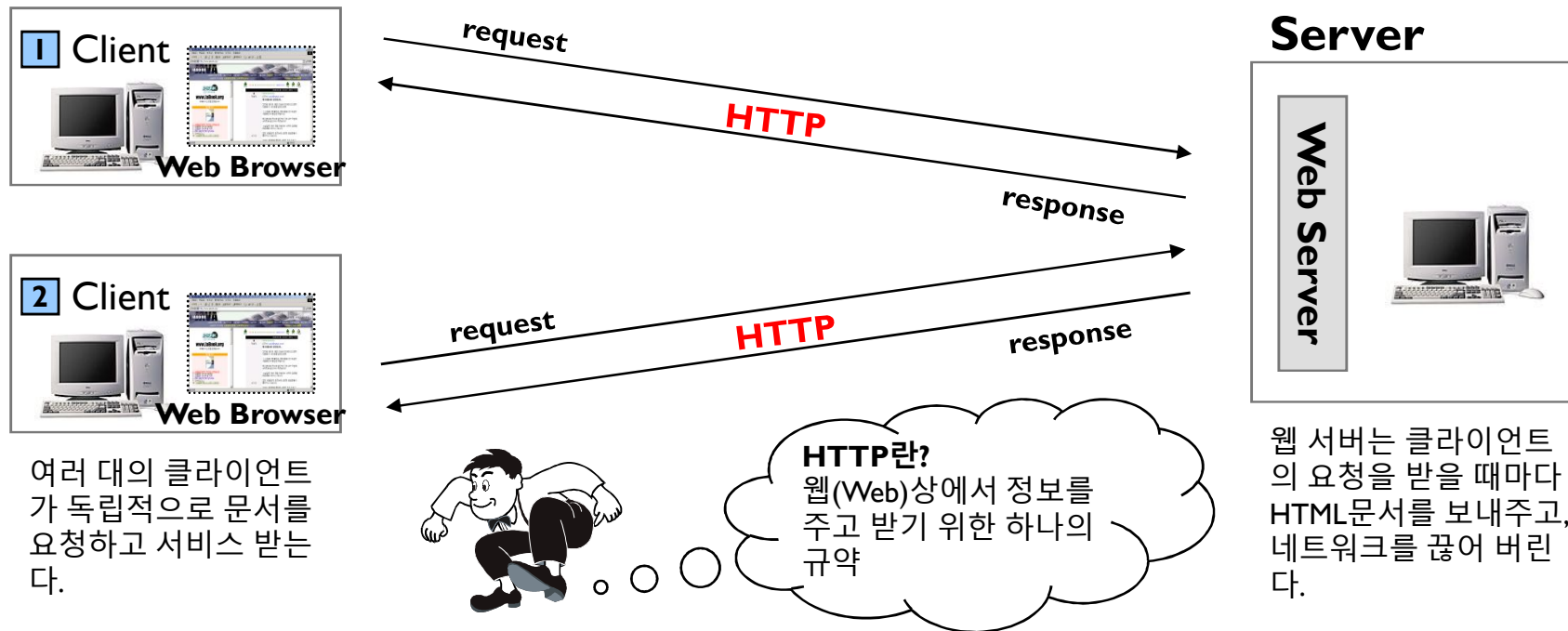
- ▶ 웹 브라우저(Web Browser)

- ▶ 웹 서버로부터 받은 HTML 문서를 해석하고 보여주는 애플리케이션
 - ▶ Internet Explorer, Netscape Browser, Mozilla Firefox



1.1.5 HTTP(HyperText Transfer Protocol)

- ▶ 프로토콜(Protocol)
 - ▶ 다른 시스템과 정보를 교환하기 위한 규약
 - ▶ 외계인과 대화하려면 외계어를 사용해야 한다.
- ▶ HTTP
 - ▶ 웹에서 정보 교환을 위해 사용되는 프로토콜



1.1.6 HTML(HyperText Markup Language)

▶ HTML

- ▶ 웹에서 사용되는 문서를 작성하는데 사용되는 언어
- ▶ 하이퍼링크와 멀티미디어를 표현할 수 있는 언어
- ▶ SGML(Standard Generalized Markup Language)에 기초하여 만들어진 언어로써 태그를 사용해서 데이터를 기술

```
<html>
  <head>
    <title>Woni JSP</title>
  </head>
  <body>
    <h2> Woni JSP</h2>
    <hr>
    Woni JSP는 여러분들께 JSP를 보다 더 쉽고 재미있게 알려주기 위한 책임니다.
    
  </body>
</html>
```



해석



1.1.7 전통적인 웹 구조의 한계

▶ 전통적 웹의 한계

▶ 정적인(Static) 웹 페이지만을 제공

- ▶ 이미 만들어져 있는 웹 페이지를 클라이언트의 요청에 따라 서비스하는 형태
- ▶ 고정된 웹 페이지의 제공만이 가능하다.

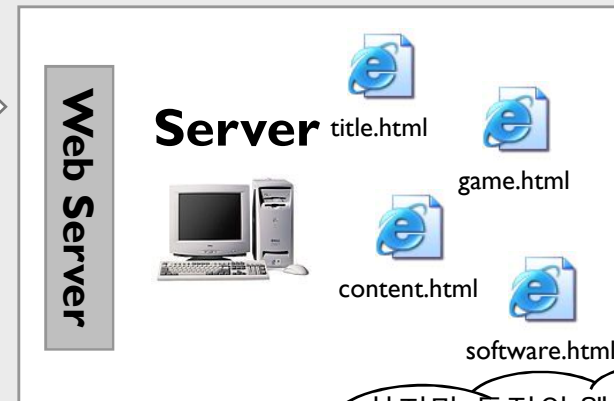
정적인 웹 페이지는 개발자가 미리 작성하여 서버에 올려놓고 서비스하는 형태로 제공된다.



"title.html" 문서 요청

"title.html" 문서 응답

title.html



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

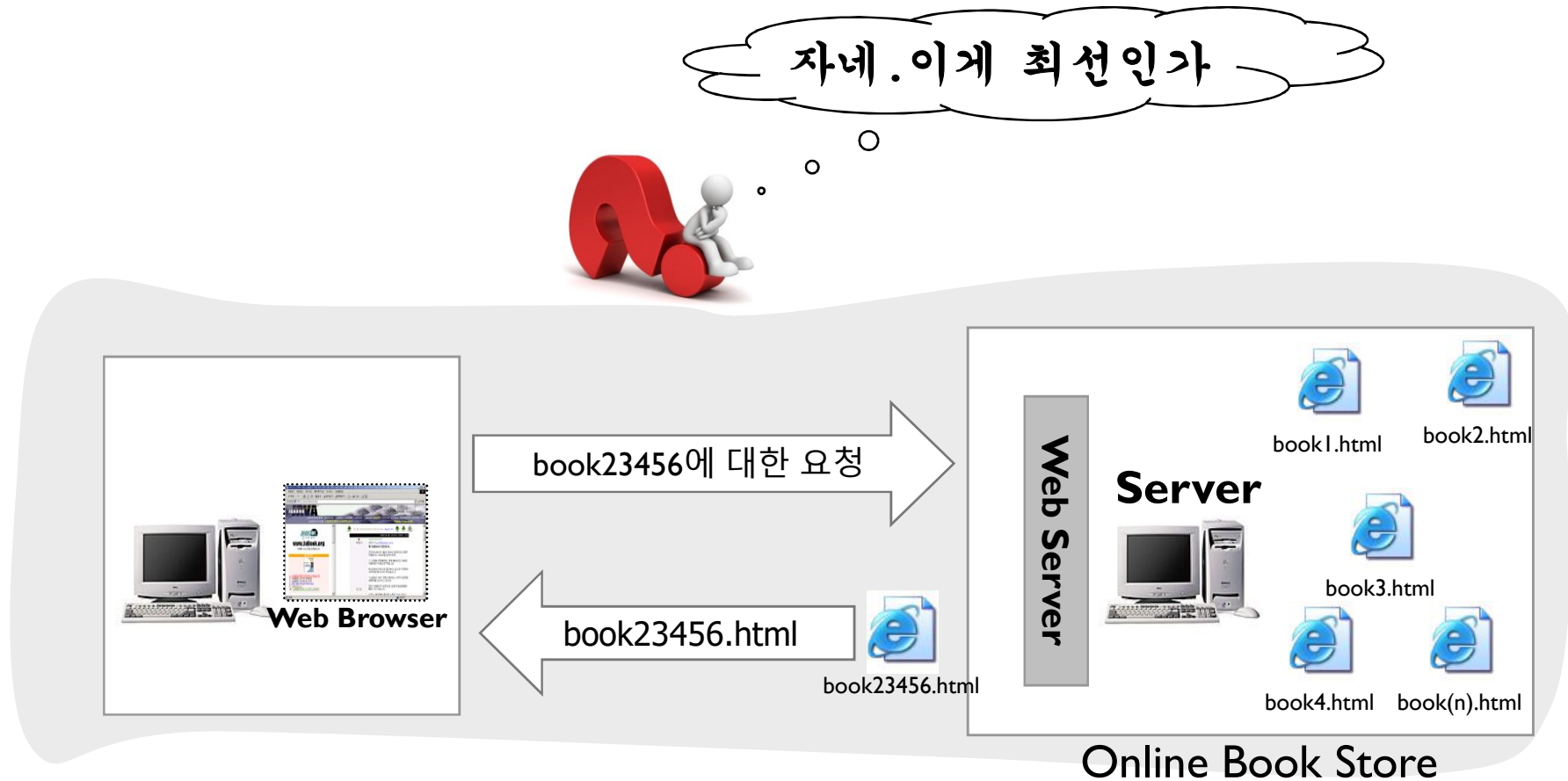
▶ 해결책

▶ 동적 웹 페이지(Dynamic Web Page)

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

▶ CGI(Common Gateway Interface) 모델

1.1.7 정적 서비스의 한계



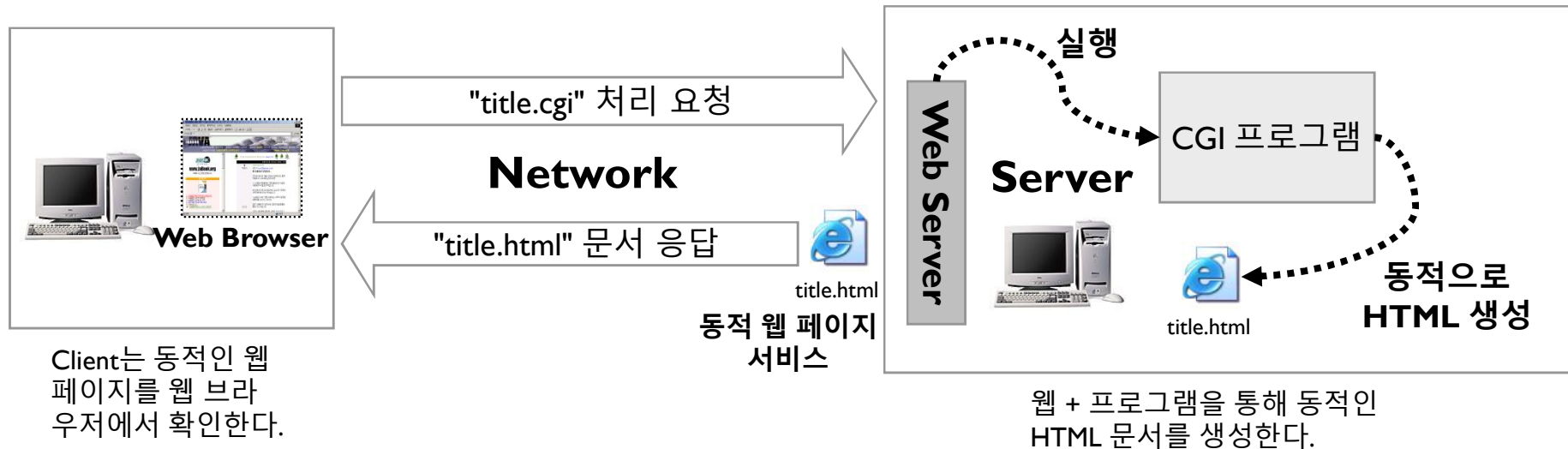
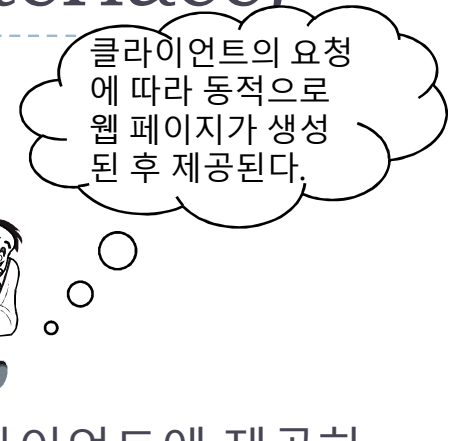
1.1.8 CGI(Common Gateway Interface)

▶ CGI

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

▶ CGI 특징

- ▶ 프로그래밍을 이용해서 동적으로 생성된 웹 페이지를 클라이언트에 제공하는 모델



1.1.9 확장 CGI

p.30

- ▶ 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)

요즘에는 대부분
확장 CGI를 사용해
서 서버 애플리케이션을
구현한다.



1.2 서블릿(Servlet)

1.2.1 서블릿이란?

p.32

- ▶ 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



1.2.2 서블릿 컨테이너 (Servlet Container) **p.31**

▶ 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.

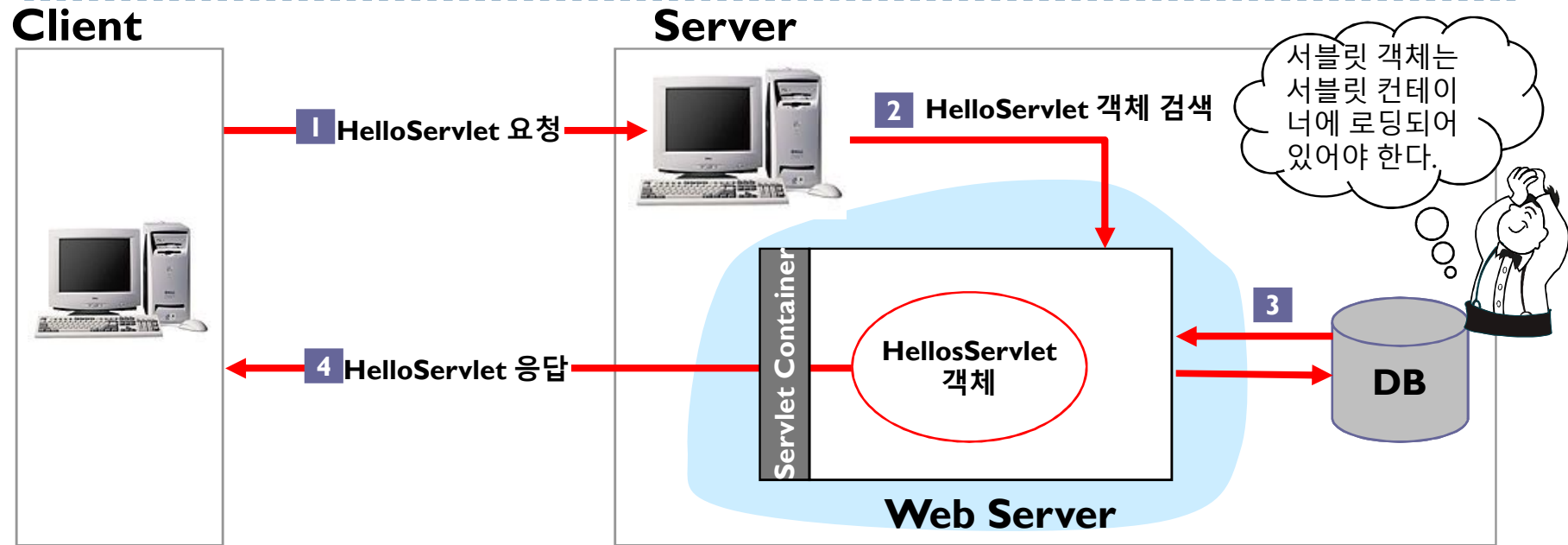


서블릿 컨테이너는
서블릿을 담아두고
클라이언트에게 서
비스해주는 웹 서
버이다.



1.2.3 서블릿의 요청과 처리

p.35



▶ 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 필요하다면 데이터베이스 연결 및 처리
4. HelloServlet 서블릿의 응답 전송

1.2.4 서블릿의 특징과 단점

p.35

▶ 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

HTML과 서블릿 코드와의 조화를 맞추기가 생각처럼 쉽지 않을걸~

▶ 서블릿의 단점

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



1.3 JSP(Java Server Page)

1.3.1 JSP란?

p.32,36

▶ JSP(Java Server Page)

- ▶ 서버릿 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 서버릿이다.
- ▶ 서버릿보다 편리하게 프로그래밍할 수 있다.
- ▶ JSP로 할 수 있는 것은 서버릿으로도 가능하다.

▶ 왜 서버릿 대신에 JSP를 사용하는가?

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



1.3.2 JSP의 요청과 처리

p.43

Client

Server

