

강의 초기 뭐라했는지



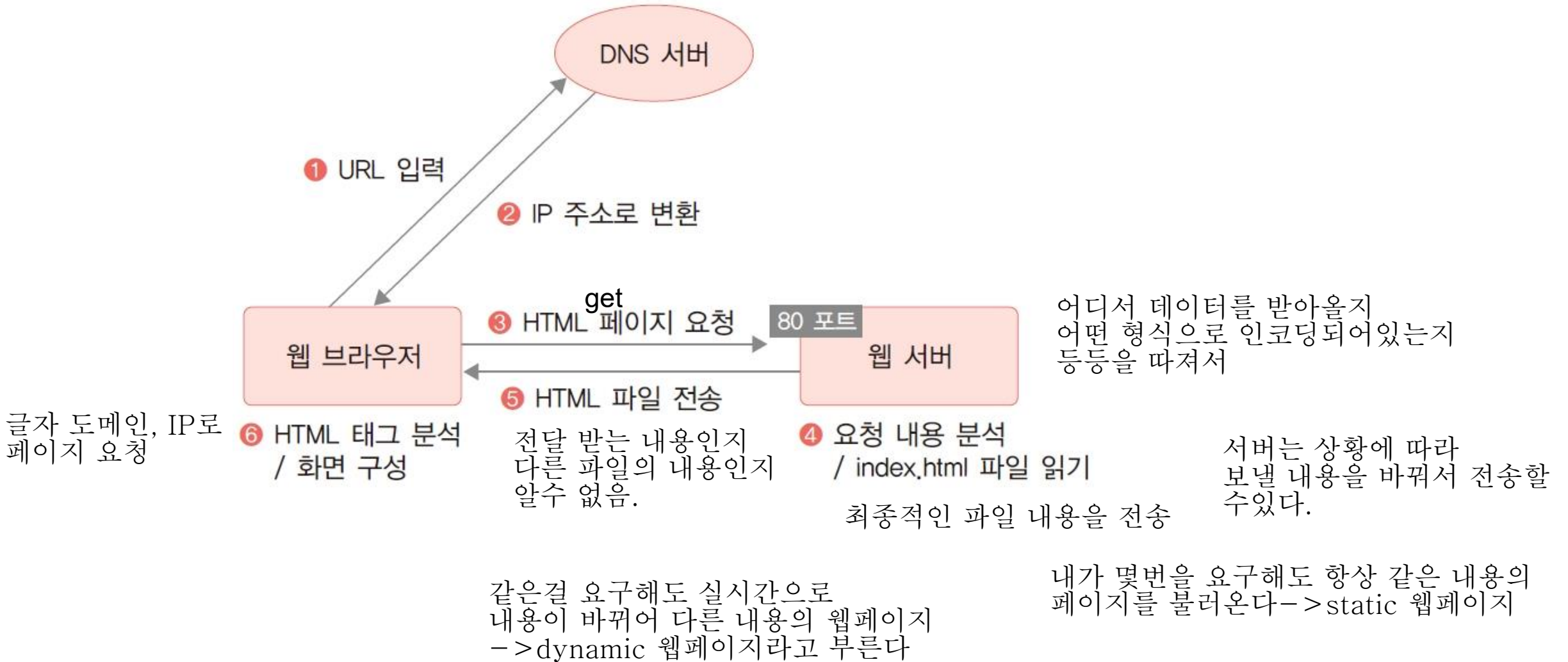
2025년 상반기 K-디지털 트레이닝

웹 어플리케이션 개요

[KB] IT's Your Life

1 HTTP 클라이언트와 서버

✓ 웹의 동작 과정

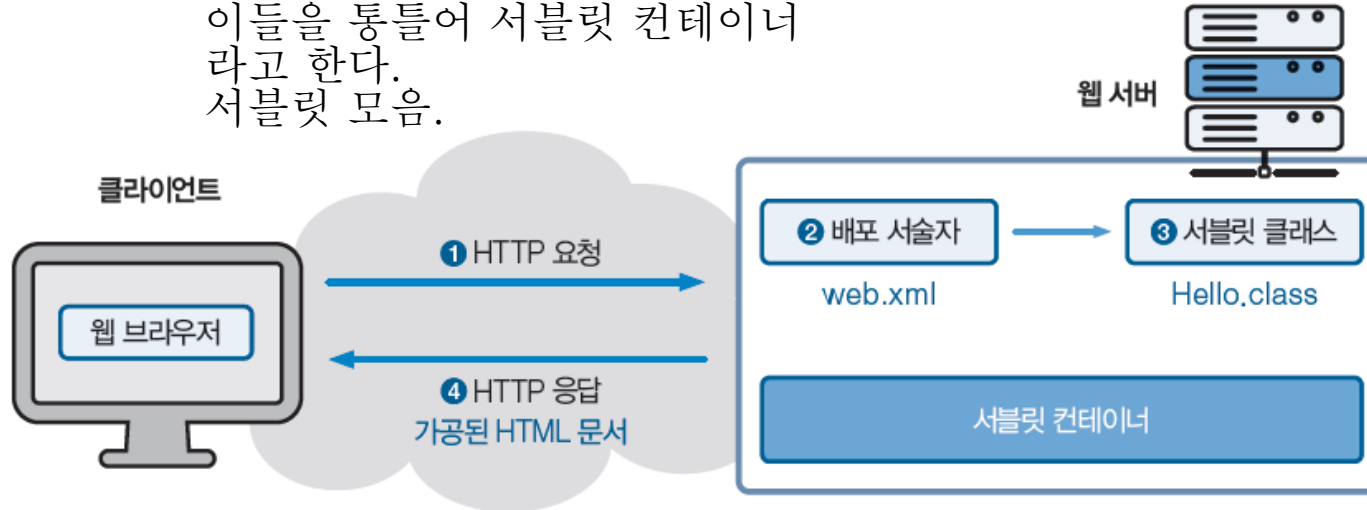


1 HTTP 클라이언트와 서버

✓ 웹의 동작 과정

자바 기반 웹어플 흐름

톰캣 말고도 제우스 등등
상용 제품들이 있다.
이들을 통틀어 서블릿 컨테이너
라고 한다.
서블릿 모음.



이때 서술자가 명시하는
자바 프로그램안에 들어있는 객체를
자바코드를 서블릿이라고 함.
작은 코드조각이라는 뜻도 있음

내용이을 변경되는 웹페이지를 구성하기 위해선
다른 웹 프로그램이 필요하다.
jsp servlet, django dotnet등이 이런것이다.

배포서술자
어떻게
동작하는지에 대한
서술자가 필요함

왜?
Tomcat이라는
WAS
web app server
의 부품으로서
우리의 JAVA
프로그램이
들어간다.
그때 이제 우리가
작성한 자바 프로그램이
무엇을 어디에 요청을
처리하는지
기능이 무엇인지
톰캣에게 알려줘야함.
톰캣이 서술자를 분석해서
자바의 객체나
기능이나 경우의 수를 파악

1 HTTP 클라이언트와 서버

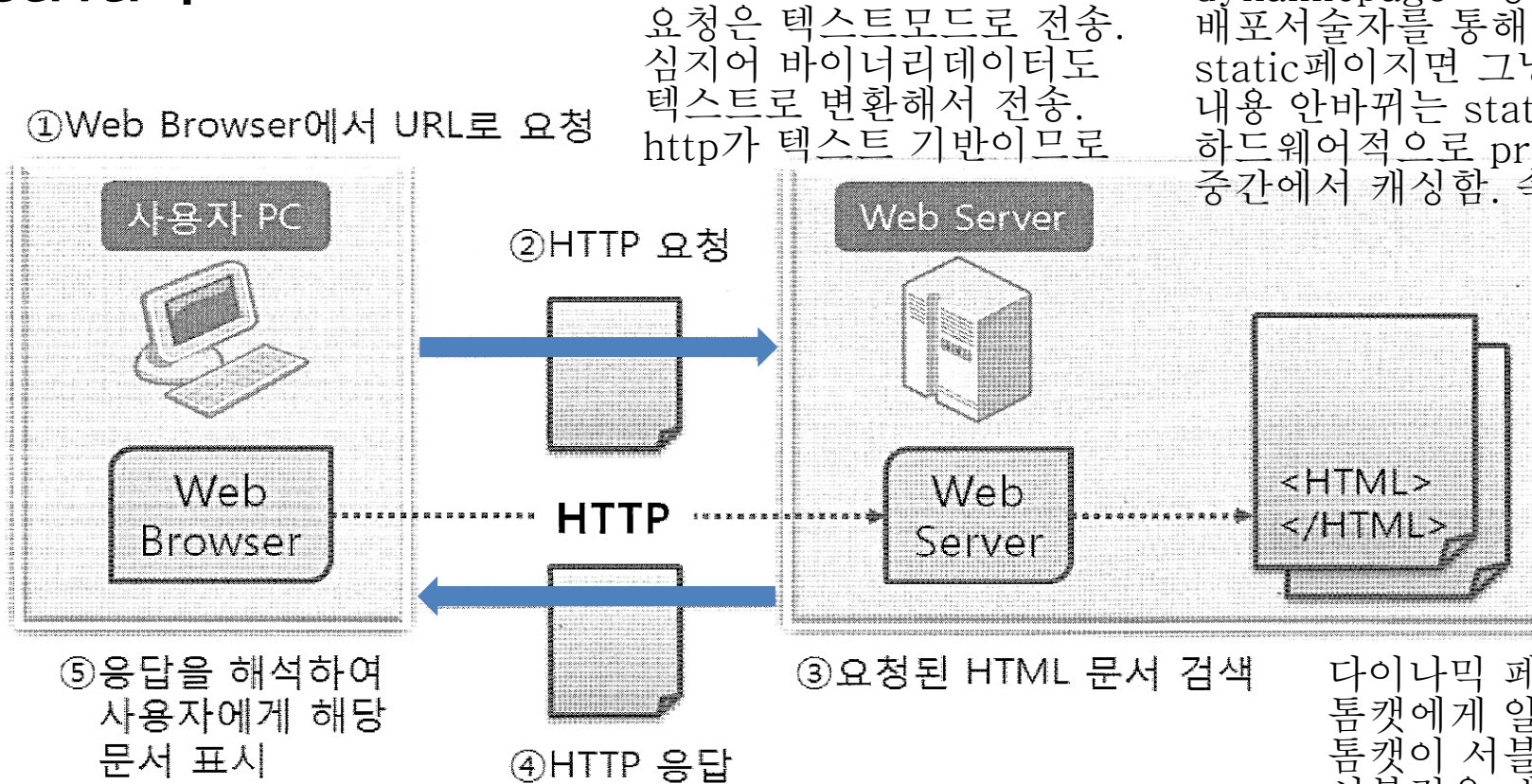
✓ URL: Uniform Resource Locator

프로토콜://호스트 번호:포트 번호/경로/파일명



1 HTTP 클라이언트와 서버

✓ Client와 Server 구조



요청은 텍스트모드로 전송.
심지어 바이너리데이터도
텍스트로 변환해서 전송.
http가 텍스트 기반이므로

웹서버는 일단 static page요청인지 dynamicpage요청인지 배포서술자를 통해 구별함.
static페이지면 그냥 파일읽고 내보냄.
내용 안바꾸는 static. 그로므로 캐싱도함
하드웨어적으로 proxy서버가 있어서
중간에서 캐싱함. 속 성능 느긋

다이나믹 페이지면
톰캣에게 알려주고
톰캣이 서블릿에게 요청하고
서블릿은 내용바꾸고
위하는 결과 반환하여
그것을 전송하는 과정.

1 HTTP 클라이언트와 서버

✓ HTTP Request

- 요청 라인(Request Line) 1줄 정보
 - HTTP 메서드 방식 및 요청 URL과 프로토콜 정보
- 요청 헤더(Request Header) n줄 정보
 - 웹 브라우저 정보, 언어, 인코딩 방식, 요청 서버 정보 등 추가 정보
- 요청 본체(Request Body) 0줄 이상
 - 요청에 필요한 내용

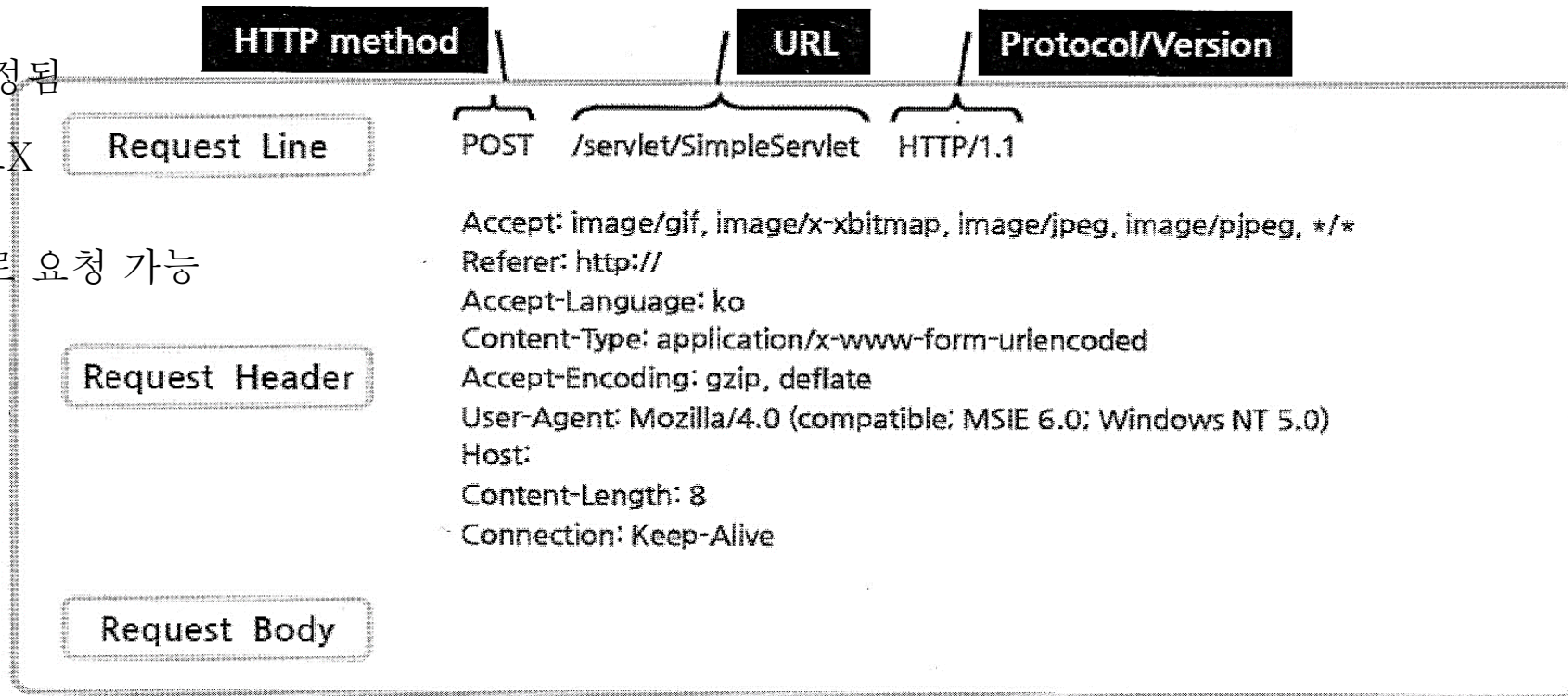
빈줄 \n
으로 구분

메소드 타입에 따라
바디의 필요 여부가 결정됨

get delete는 바디 필요X

get post는 form태그로 요청 가능

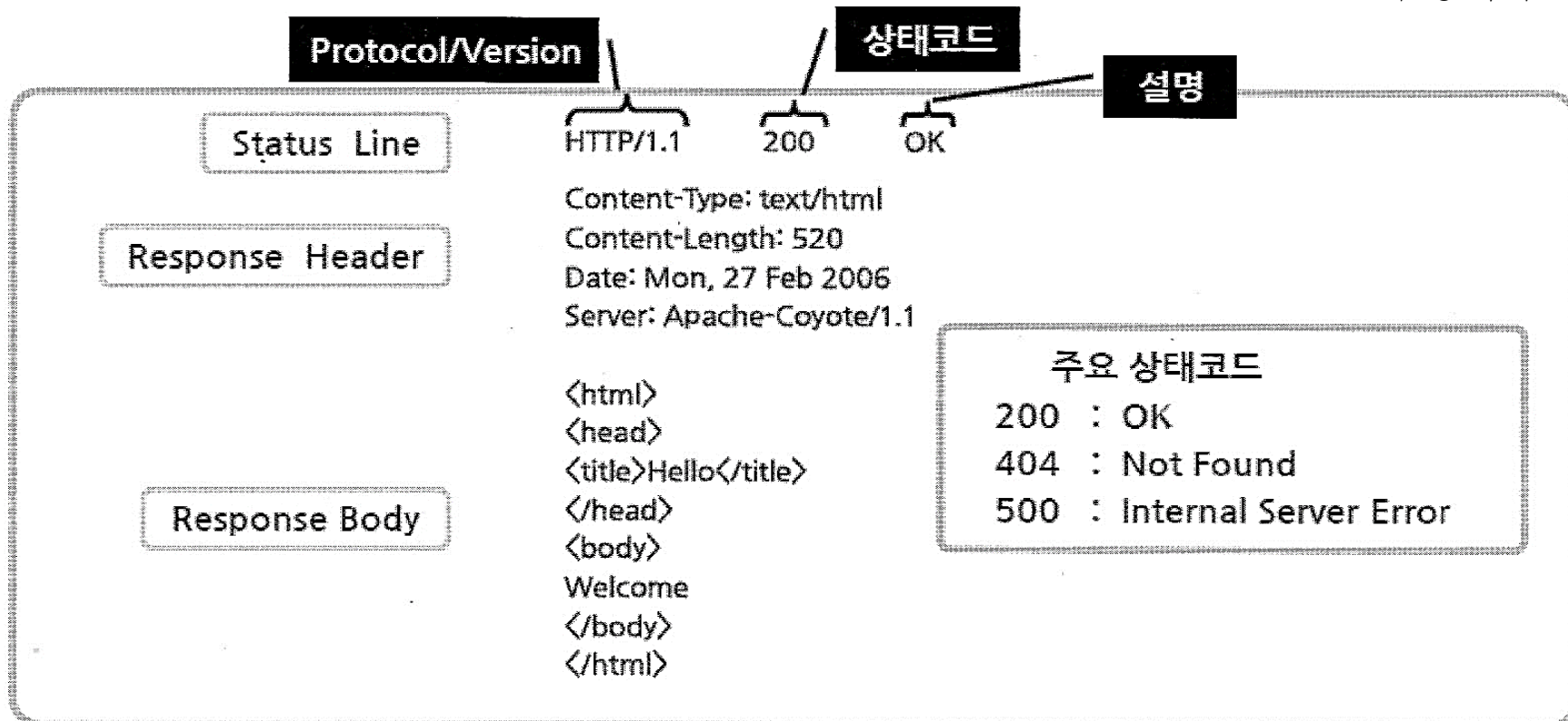
나머지 메소드는
js로, ajax로 가능



✓ HTTP Response

- **상태 라인** 1줄
 - 응답 상태 코드 및 프로토콜 정보
- **응답 헤더** 줄n개
 - 응답 처리 날짜, 인코딩 방식, 요청 서버 정보 등과 같은 추가 정보
- **응답 본체**
 - 응답에 필요한 내용. 일반적으로 HTML 문서

텍스트가 아닌 데이터를 보낼 때는
헤더의 타입을
수정해야 한다



300 redirect

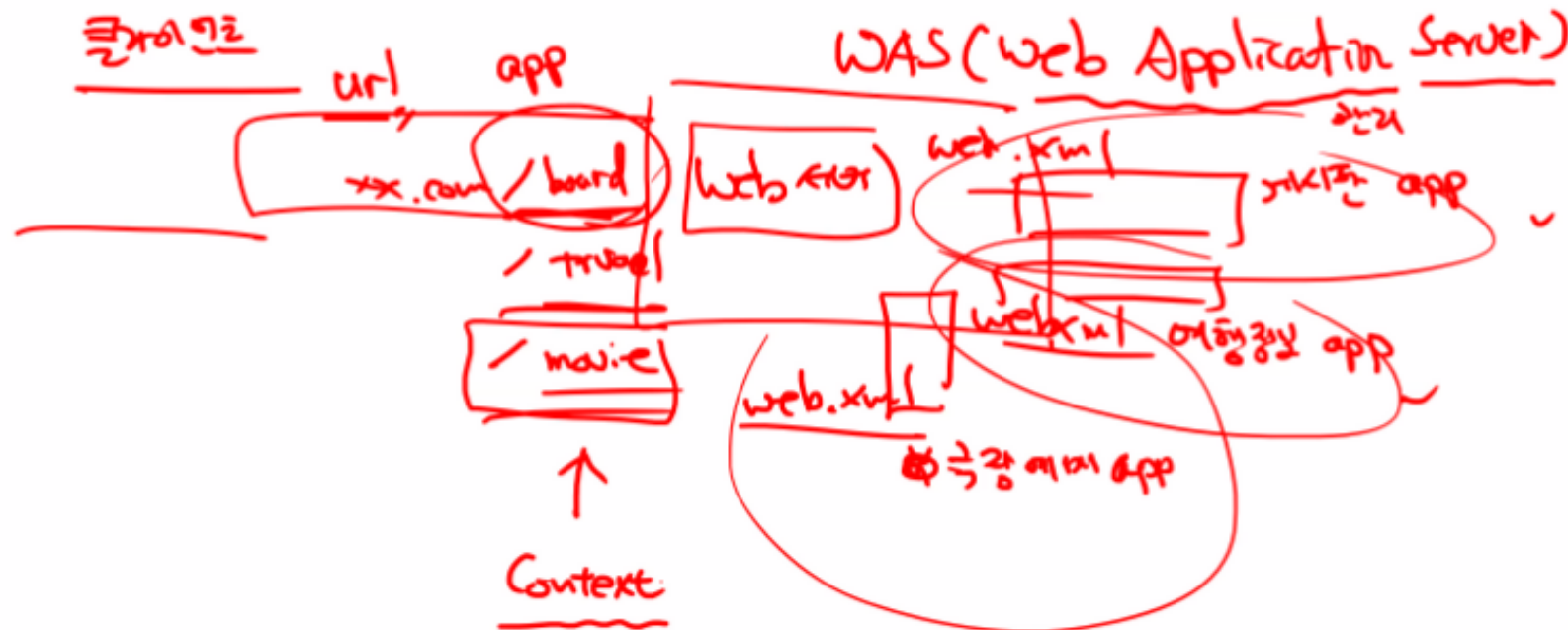
400번대는 프론트
500번대는 백엔드
가 문제

2 웹 어플리케이션

✓ 웹 어플리케이션이란?

- 인터넷을 기반으로 구성된 웹상에서 웹 브라우저를 이용한 클라이언트와 HTML (정적 웹 컴포넌트), JSP/Servlet(동적 웹 컴포넌트) 같은 자원을 가진 웹 서버 간에 동적으로 요청/응답 처리하는 프로세싱

was는 웹서버를 관리하는. 한서버당 1개이상의 was가 있을 수 있다.
 각자 독립적으로 개발 가능하다. 한개의 웹서버에서 다양한 WAS 구동가능.
 각WAS마다 web.xml서술서가 있어서 구분가능하다.
 하지만 클라이언트는 모른다. 물리적으로 어느 것이 구분되어있고 구성되는지모른다
 클라이언트URL만 알고 있는 것. @@@.com/ board /travble/ movie등등
 클라이언트 입장에서 url의 구분 글자만 보고 판별함. 구분글자?를 컨텍스트라고 백엔드에서 불린다.
 앱을 구분하는 용어로 컨텍스트라고함. 기억하세요



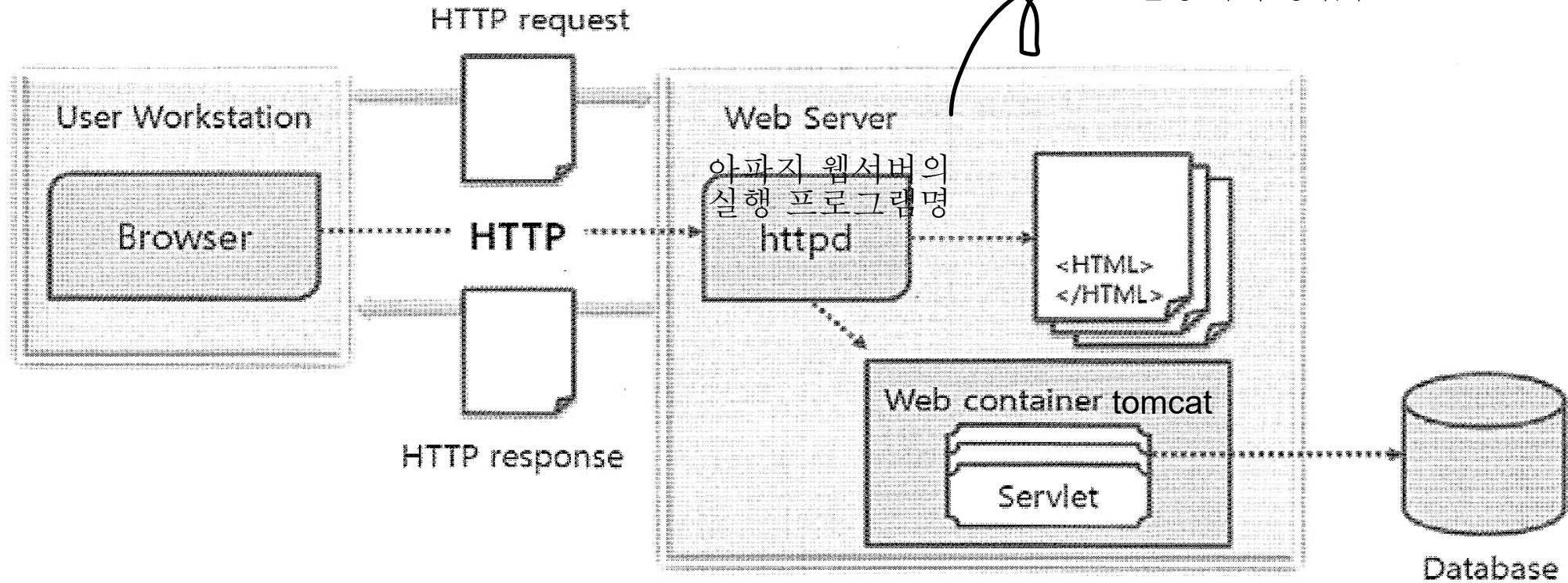
2 웹 어플리케이션

✓ JSP/Servlet

엄격하게 구분하진 않고
웹서버 was 막 분리 구분할 필요는 없음
서로가 당연히 존재해야 하기에.

세부적으로 들어갈때만 유의

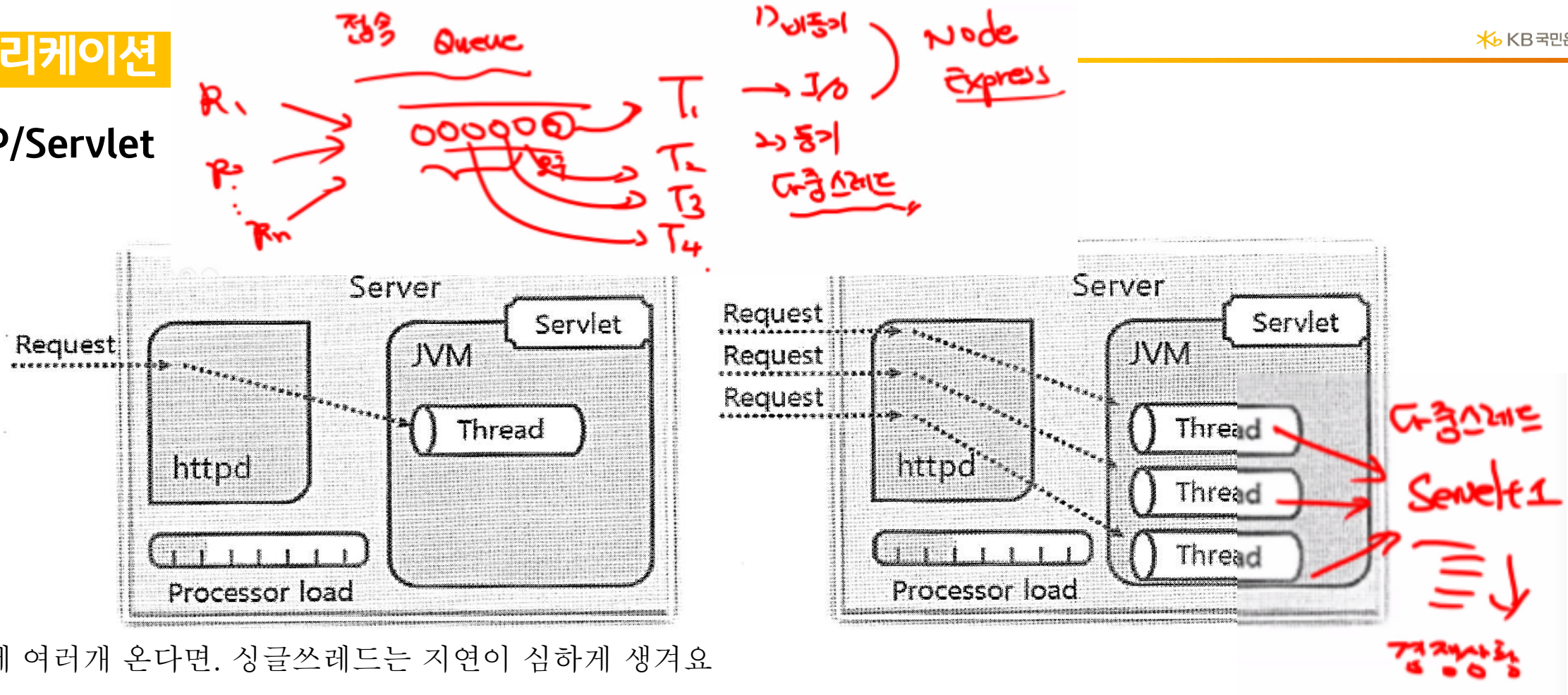
정적페이지는 웹서버가 담당.
바로 읽고 보냄.
동적 페이지는 WAS(WEBcontainer)가
담당 servlet(우리가 만들 코드)
를 통해서 응답함.



데이터베이스
연동 필요할수도

2 웹 어플리케이션

✓ JSP/Servlet



유클이 동시에 여러개 온다면. 싱글쓰레드는 지연이 심하게 생겨요

서버가 접속 큐라는 것은 활용함. 요청이 들어온 순서대로 담아놓고 하나씩 꺼내서 응답을 피함. 첫번째 해결책 비동기모드로 처리. 두번째 해결책 동기식으로 처리하되, 다중쓰레드 사용.

비동기모드로 처리는 NODE express

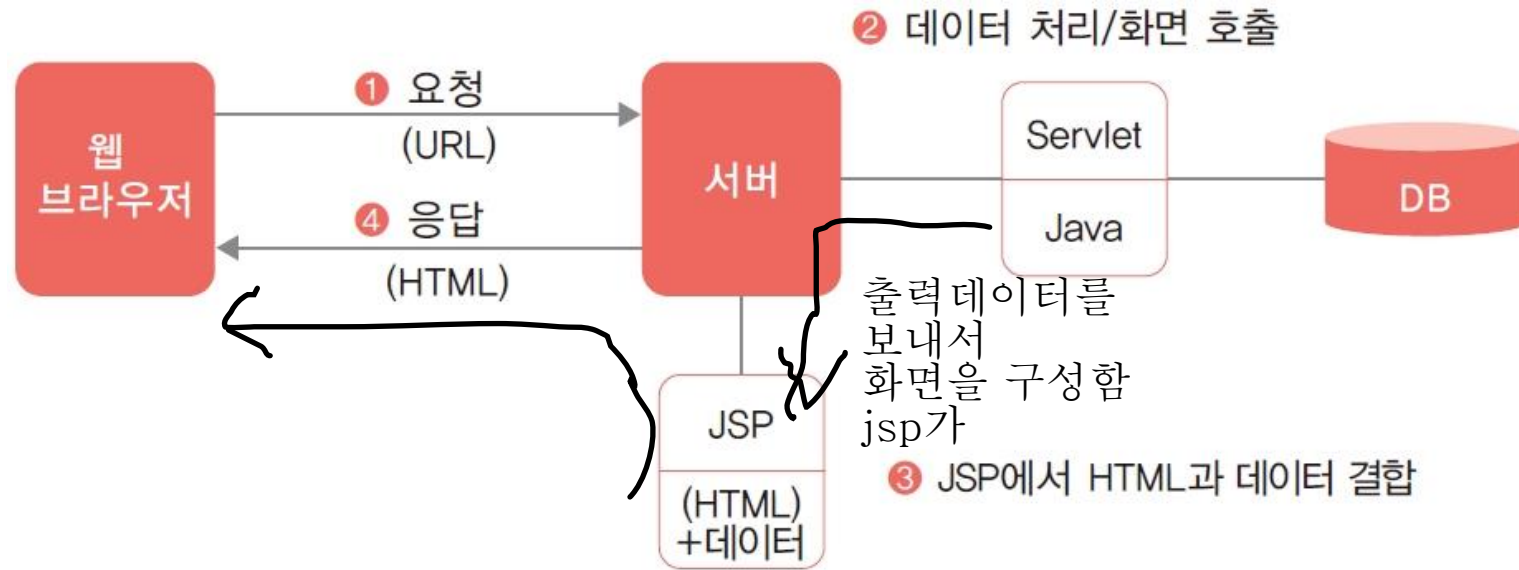
동기식으로 처리할때는 JAVA의 다중쓰레드 사용.쓰레드 갯수만큼 동시처리.

쓰레드 하나당 서블릿하나가 배정될수도. 또한 쓰레드 여러개가 서블릿 하나를 공유할수도. -> 동기화유의.

하지만 프레임워크를 사용하면 잘 처리해줌. 크게 신경쓸필요는 없다. 일반적인 경우에는. 동기화를 직접할 경우는 잘 없다!

2 웹 어플리케이션

✓ 백엔드 동작의 개요



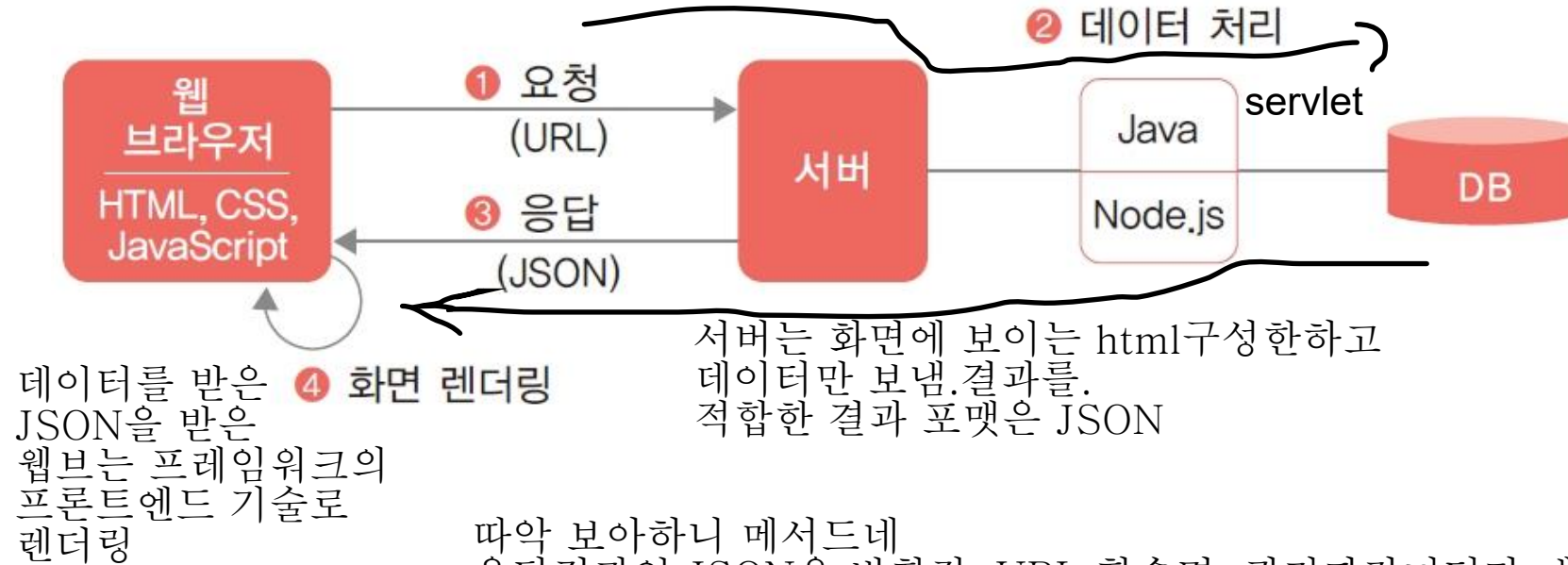
서블릿:
화면과 무관한
비즈니스로직을 처리함.

JSP:
비즈니스로직관심X
결과를 화면에 어떻게
보여줄거냐.
다른말로
html과 데이터를 묶어서
어떻게 보여줄지

전문 프론트엔드 프레임워크와 연동하는 경우

백엔드의 기존 역할에서 JSP기술이 필요없어짐. 해당 파트 없어짐.

✓ 프론트엔드 동작의 개요



따악 보아하니 메서드네
응답결과인 JSON은 반환값. URL 함수명, 쿼리파라미터가 매개변수
메소드 호출과 다르게 없다.

하지만 JSP를 알아야하긴해

3 웹 컴포넌트

정적페이지와 동적페이지가 혼용될수도 있다.

✓ 웹 컴포넌트의 종류

- HTML 컴포넌트 클라이언트에서 보이는
- JSP 컴포넌트 JSP컴포넌트는HTML컴포넌트를 만듦.
- Servlet 컴포넌트 비즈니스로직을 담당

✓ 웹 컨테이너(Web Container) WAS webapp server

- JSP와 Servlet 컴포넌트를 관리하는 역할 ✓
- 웹 컴포넌트는 시작점이 없음
- 이벤트(웹 브라우저의 요청) 방식으로 동작
- 클라이언트의 요청에 따라 웹 컴포넌트를 생성하고 제거하는 메커니즘을 제공
- Tomcat, Resin 등

코드 작성시 실제 main메서드는 TOMCAT이 갖고있기에
우리는 main작성안하고 부품을 만들어 ROMCAT에게 공급
하면돼

✓ Model2 아키텍처(MVC 패턴)

WAS 이론적 배경 끝

