

웹 서버 & 웹 어플리케이션 서버

Web Server & Web Application Server



웹 서버 (Web Server)



클라이언트 (Client)

사용자 또는 브라우저 응용프로그램을
뜻하기도 함



정적 콘텐츠 (Static Contents)

사전에 제작되어, 더 이상 어떤 처리도 하지 않고
단순히 반환되는 모든 종류의 데이터들을 의미

HTML페이지, 단순 텍스트파일, 이미지 파일,
비디오 파일 등



동적 콘텐츠 (Dynamic Contents)

사용자의 요청에 따라 결과값이 달라지는 콘텐츠

모바일 앱 또는 온라인상 내 위치 기반의
지역 날씨 콘텐츠, 쇼핑물의 장바구니 등



서버 (Server)

요청을 받아 알맞은 기능이나 데이터를 제공

서버 프로그램이 실행되는 컴퓨터

웹 서버 *Web Server* 란?

Web Server & Web Application Server

- 클라이언트는 서버에 HTML 문서를 요청
- 웹 서버는 사용자가 요청한 HTML 문서를 웹 브라우저에게 전달
- 웹 브라우저는 웹 서버로부터 전송 받은 HTML 문서를 읽어 들인 후 해석
- HTML 문서를 보여주기 위해 필요한 img, CSS, JS 같은 리소스들에 대해 URL을 추출하며 동시에 웹 서버에 여러 개의 리소스를 요청하고 웹 서버는 이 여러 개의 요청을 받아 들인 뒤 하나로 합쳐서 그 결과를 브라우저에 전송

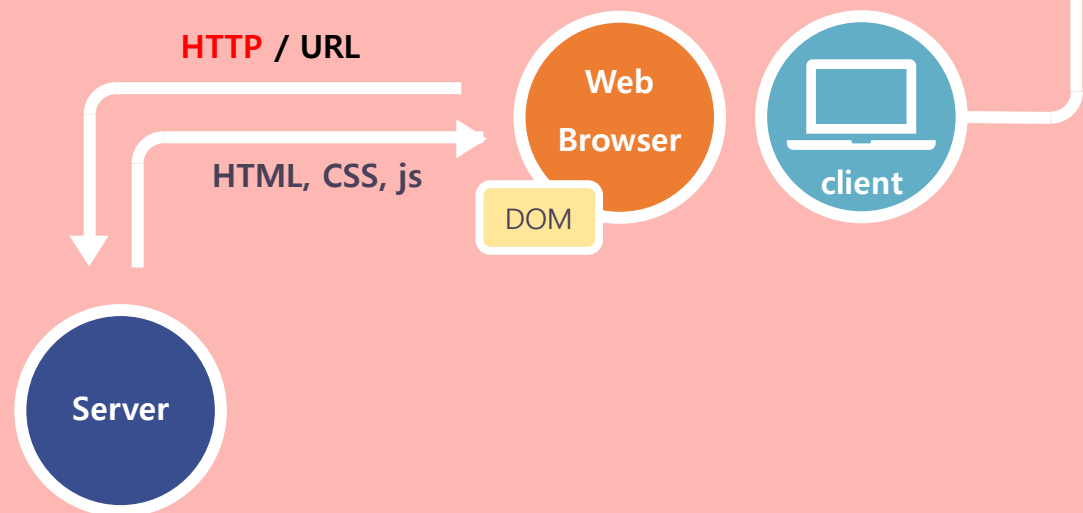
Static Pages

HTML : 하이퍼텍스트 마크업 언어

인터넷에서 웹을 통해 보여지는 사이트의 기본적인 언어로, 가장 기초가 되는 마크업, 규칙에 따라 정해진 태그, 속성값으로 이루어진 언어

DOM : 문서 객체 모델

HTML 문서에 접근할 수 있는 API로, 객체로 이루어져 스크립트나 CSS같은 언어들이 DOM 구조에 접근하여 커스텀 할 수 있게 만들어주는 연결 부분 역할을 갖는다



웹 서버 (Web Server) 란

- 웹 서버는 소프트웨어(Software)를 보통 말하지만, 웹 서버 소프트웨어가 동작하는 컴퓨터를 말함
- 웹 서버의 가장 중요한 기능은 클라이언트 (Client)가 요청하는 HTML 문서나 각종 리소스를 전달하는 것
- 웹 브라우저가 요청하는 리소스는 컴퓨터에 저장된 정적(static)인 데이터이거나 동적(dynamic)인 결과가 될 수 있습니다.

웹 서버와 웹 브라우저

- URL 주소가 HTTP로 시작되는 이유는 HTTP를 사용한다는 의미
- HTTP란 인터넷상의 데이터를 주고받기 위한 프로토콜(규약, 규칙)
- 클라이언트와 서버도 미리 정해 놓은 규칙을 가지고 통신을 해야만 제대로 통신가능

웹 서버 *Web Server* 소프트웨어의 종류

Web Server & Web Application Server

SOFTWARE

01



02



03



CONTENTS

아파치

오픈소스 소프트웨어로 대부분의
운영체제에서 설치 및 사용 가능

01

엔진엑스

동시접속 처리에 특화된 웹 서버

02

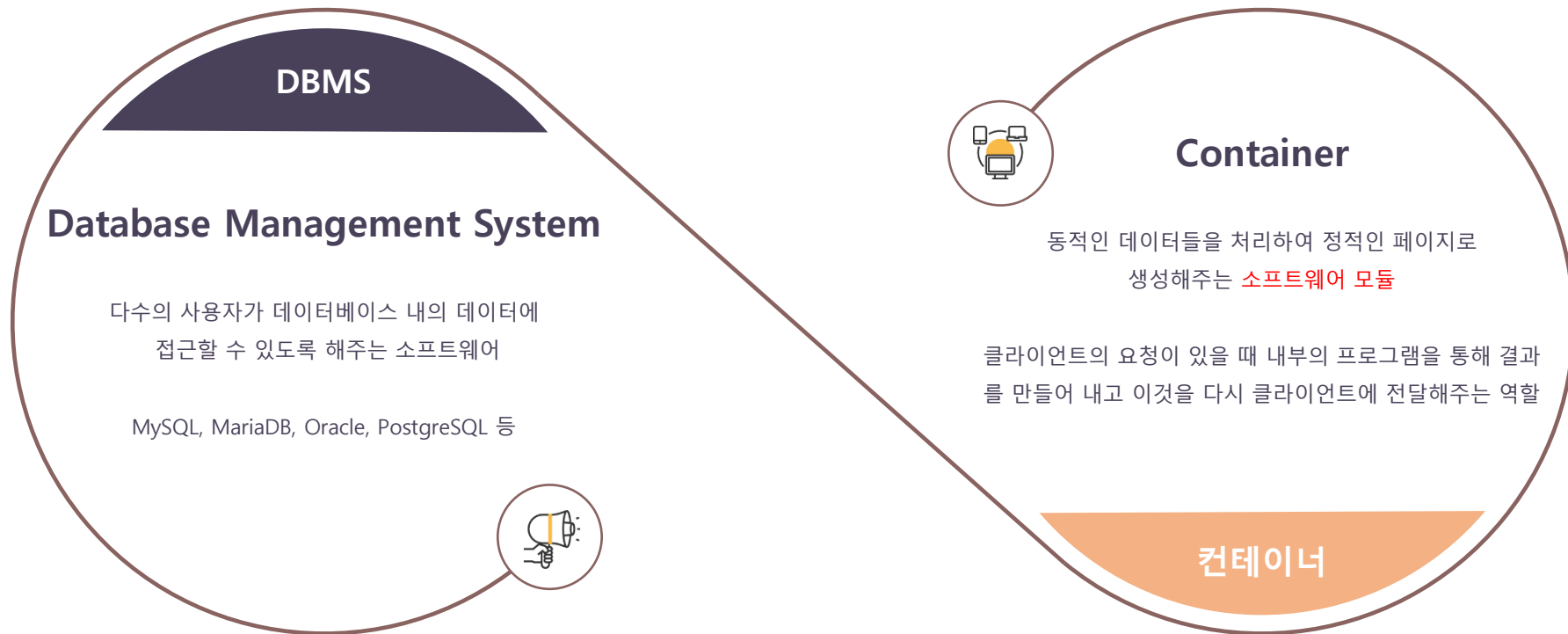
Internet Information

Services

윈도우 OS에서만 사용 가능
ASP 스크립트 언어 사용 가능

03

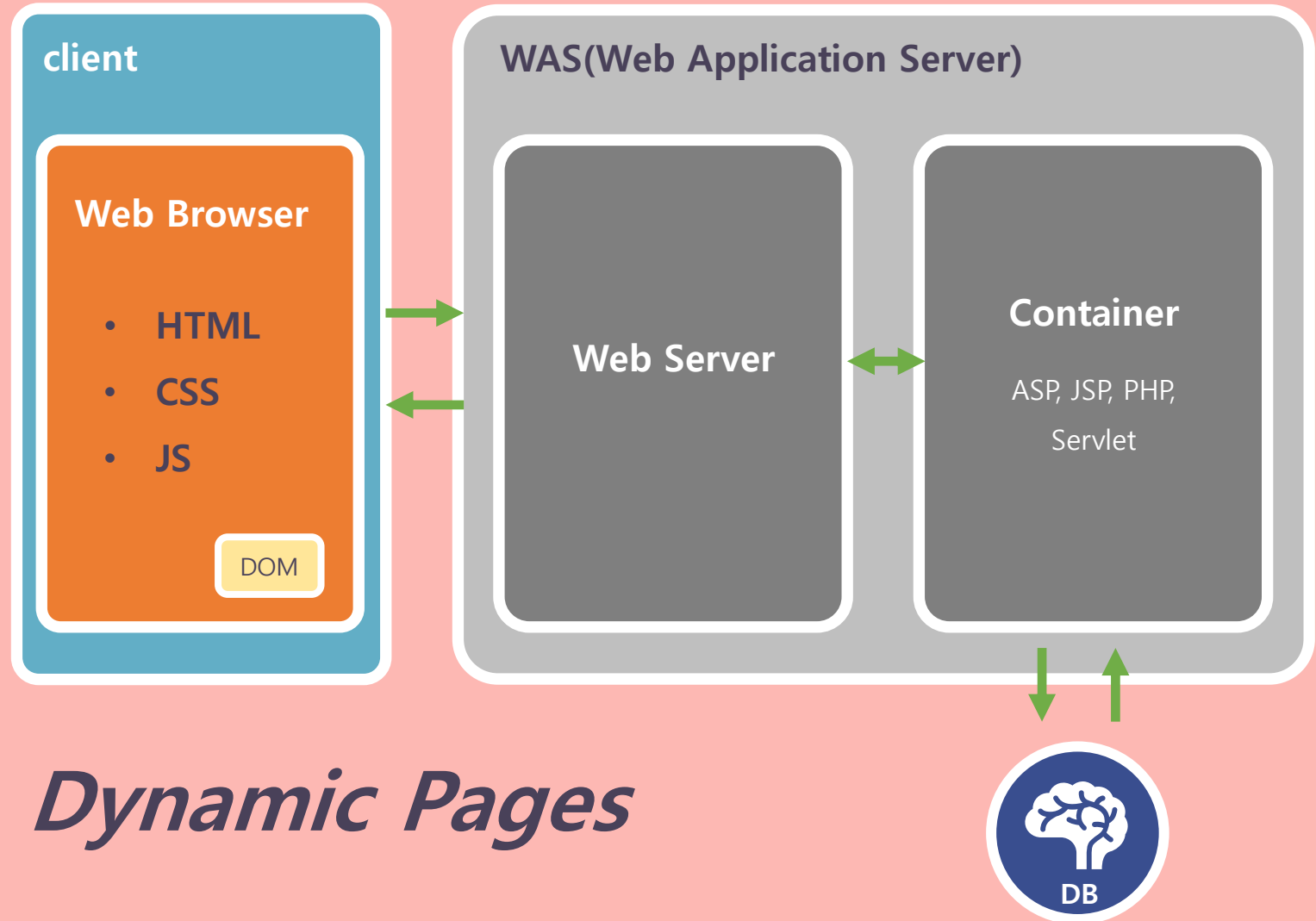
웹 어플리케이션 서버 (*Web Application Server*)



웹 어플리케이션 서버 **WAS** 란?

Web Server & Web Application Server

- 웹 서버는 클라이언트의 요청을 받아 컨테이너로 전송
- 웹 컨테이너는 JSP, Servlet 등 구동 환경을 제공하여 동적 데이터 처리 및 데이터 베이스 접속이 필요한 작업을 실행
- 데이터 베이스를 거쳐 처리된 결과값을 받아 클라이언트로 전송



웹 어플리케이션 서버 (Web Application Server)

- 웹이 정적인 기능을 넘어 동적인 기능이 필요 해지며 동적인 기능은 프로그래밍 기능으로 구현 가능
- DBMS와 연관되어 웹이 복잡해짐에 따라 보다 복잡한 프로그래밍 기능을 요구하게 됨
- WAS는 프로그램의 동적인 결과를 웹 브라우저에게 전송하는 역할
- WAS또한 자체적으로 웹 서버 기능이 내장되어 있으며 초창기에는 성능 부실했으나 현재는 WAS만으로 충분하다

WAS의 주요 기능

- 프로그램 실행 환경, DB 접속 기능 제공
- 여러 개의 트랜잭션(논리적 작업단위) 관리
- 비즈니스 로직 수행
- 비즈니스 로직 : 유저의 눈에 보이지는 않지만, 유저가 바라는 결과물을 올바르게 도출할 수 있게 짜여진 코드 로직

웹 어플리케이션 **WSA** 소프트웨어의 종류

Web Server & Web Application Server

01

SOFTWARE



Apache
Tomcat



CONTENTS

톰캣

동적인 데이터 처리 가능 DB연결,
데이터 조작, 다른 응용프로그램
과 상호 작용에 용이

01

02



웹 로직

기존 및 클라우드 환경에서 어플리케이션을 개발 및 배포하기 위한 최고의 어플리케이션 서버

02

03



제우스

J2EE의 full Spec
국내기업으로 유지보수 및 장애
시 문제해결 용이

03

Web Server와 WAS의 차이점



웹 서버가 필요한 이유

웹 서버가 정적 요청을 처리 하면 WAS의 요청/응답 과정이 생략되어 서버의 부담을 줄일 수 있음

WAS가 필요한 이유

- 웹 페이지에 존재하는 동적 콘텐츠를 처리
- 웹 서버만을 이용한다면 동적으로 동작해야 하는 콘텐츠의 결과를 웹 서버에 모두 미리 로드해야 하는 자원낭비가 발생
- WAS를 통해 클라이언트의 요청에 맞는 콘텐츠를 그때 그때 제공 가능

기능을 분리하는 이유

- 두개의 서버 프로세스가 별개의 작업을 처리하게끔 하여 상대적으로 바쁜 WAS에 가해지는 부하를 줄임
- 물리적으로 두 서버를 분리하여 보안 강화 및 유지보수 용이
- 대용량 처리를 필요로 하는 웹 서비스의 경우 웹 서버 하나에 여러 개의 WAS를 분산하여 연결 가능

웹 어플리케이션의 발전 흐름

Web Server & Web Application Server

CGI 방식

초기 전통적인 어플리케이션 기술
각각의 클라이언트 요청에 대하여 독립적인 별도의 프로세스가 생성

JSP

스크립트 방식의
웹 어플리케이션 기술
HTML과 자바 코드를 분리시킴으로써 웹 어플리케이션 개발 시 콘텐츠 변경 유용

오픈소스 프레임워크

스트러츠(Struts), 스프링(Spring),
아이바티스(iBatis) 등의 프레임워크



서블릿 방식

요청과 응답 기반 서비스에 실행되는
자바 기반의 모듈
서블릿은 자바 가상 머신(JVM)이라는 다일 프로세스 환경에서 클라이언트의 요청에 따라 다중 스레드로 동작

MVC

컴포넌트 기반의
웹 어플리케이션 기술
프리젠테이션 로직과 비즈니스 로직이 분리되는 방식의 패턴

SOA

(서비스 지향 아키텍처)

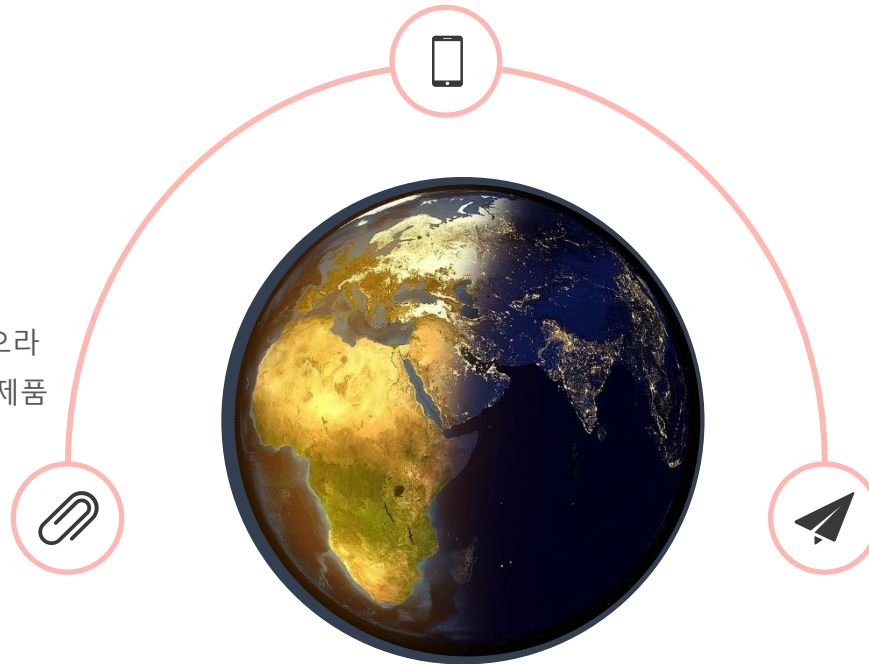
시스템과 어플리케이션 프로세스를 기업 서비스에 맞게 재설계하는 개념

퓨전 미들웨어

웹 포털, 비즈니스 인텔리전스 소프트웨어, 오라클 협업 스위트 등 다수 컴포넌트를 통합한 제품

BEA 웹 로직 서버

무정이지(Hot-Swappable)
시스템다운 없는 유지보수 및 업그레이드



감사합니다!

Web Server & Web Application Server