숭실대학교 Soongsil University

Internet과 Web Programming

숭실대학교 AI융합학부
School of AI Convergence

Contents

1. Network 개요
2. Internet
3. Desktop Application
4. Client/Server Application
5. Client/Server Architecture의 진화
6. Web Application Architecture
7. Web Application 개발 Framework
8. Python 기반 Web 개발 Framework
9. 실습 환경

Network개요

- Network
  - 프로토콜을 사용하여 데이터를 교환하는 시스템의 집합을 통칭
  - 전송 매체로 서로 연결된 시스템의 모음
  - 시스템, 인터페이스, 전송매체, 프로토콜, 표준화로 구성됨

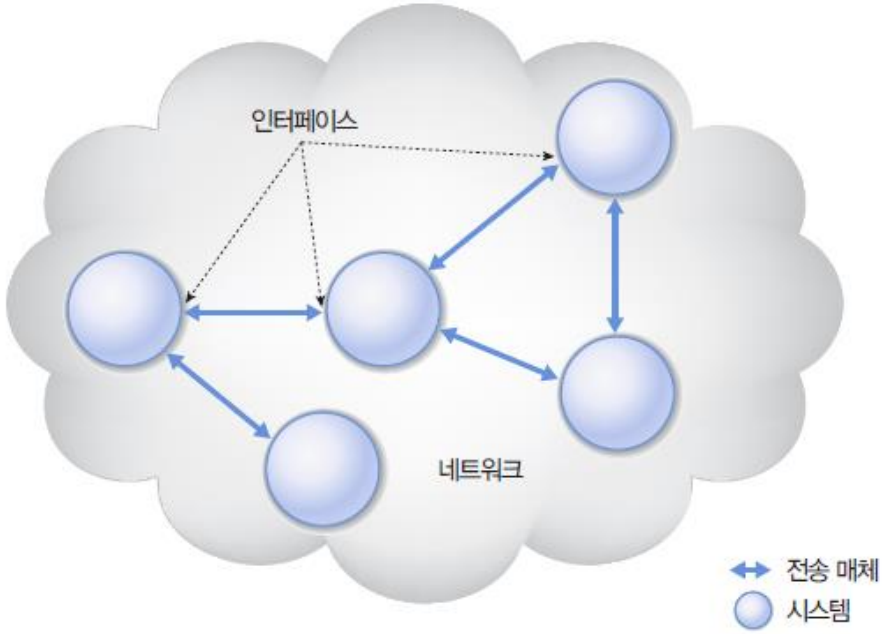


그림 1-1 네트워크의 구성

Network개요

- 시스템
  - 내부 규칙에 따라 능동적으로 동작하는 대상
  - 예: 컴퓨터, 자동차, 커피 자판기, Micro Processor, 운영체제 등
- 인터페이스
  - 시스템과 전송 매체의 연결 지점에 대한 규격
  - 예: RS-232C, USB
- 전송매체
  - 시스템끼리 데이터를 전달하기 위한 물리적인 전송 수단
- 통신 protocol 정의
  - 통신하고자 하는 두 컴퓨터 사이에 정해진 규약에 따라 접속을 하고 데이터를 주고 받도록 하기 위해서 미리 정해 놓은 규약

Network개요

- Internet(인터넷)
  - 전세계의 local network과 시스템들이 유기적으로 연결되어 동작하는 통합 네트워크
  - 공통 기능: IP(Internet Protocol)

- Network 표준화
  - 서로 다른 시스템이 상호 연동해 동작하기 위한 통일된 연동 형식

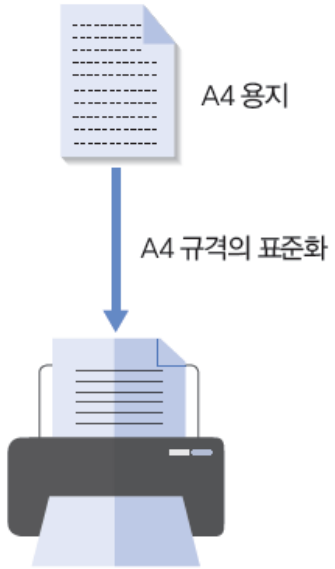


그림 1-2 A4 규격의 표준화

ISO OSI 통신 표준 vs TCP/IP

OSI 7계층	TCP/IP 4계층	
응용 계층	응용 계층	• 네트워크를 사용하는 WWW, FTP, 텔넷, SMTP 등의 응용 프로그램으로 구성.
표현 계층		
세션 계층	전송 계층	• 도착지까지 데이터를 전송 • 각각의 시스템을 연결 • TCP 프로토콜을 이용하여 데이터를 전송
전송 계층		
네트워크 계층	인터넷 계층	• 데이타를 정의 및 경로 지정 • 정확한 라우팅을 위해 IP 프로토콜을 사용 • IP 주소가 위치하는 계층
데이터 링크 계층		
물리 계층	물리 계층	• 물리적 계층 즉 이더넷 카드와 같은 하드웨어

- Protocol과 Interface
  - 프로토콜 : 서로 다른 호스트에 위치한 동일 계층間 통신 규칙
  - 인터페이스 : 같은 호스트에 위치한 상하위 계층 사이의 규칙
  - 서비스 : 하위 계층이 상위 계층에 제공하는 인터페이스

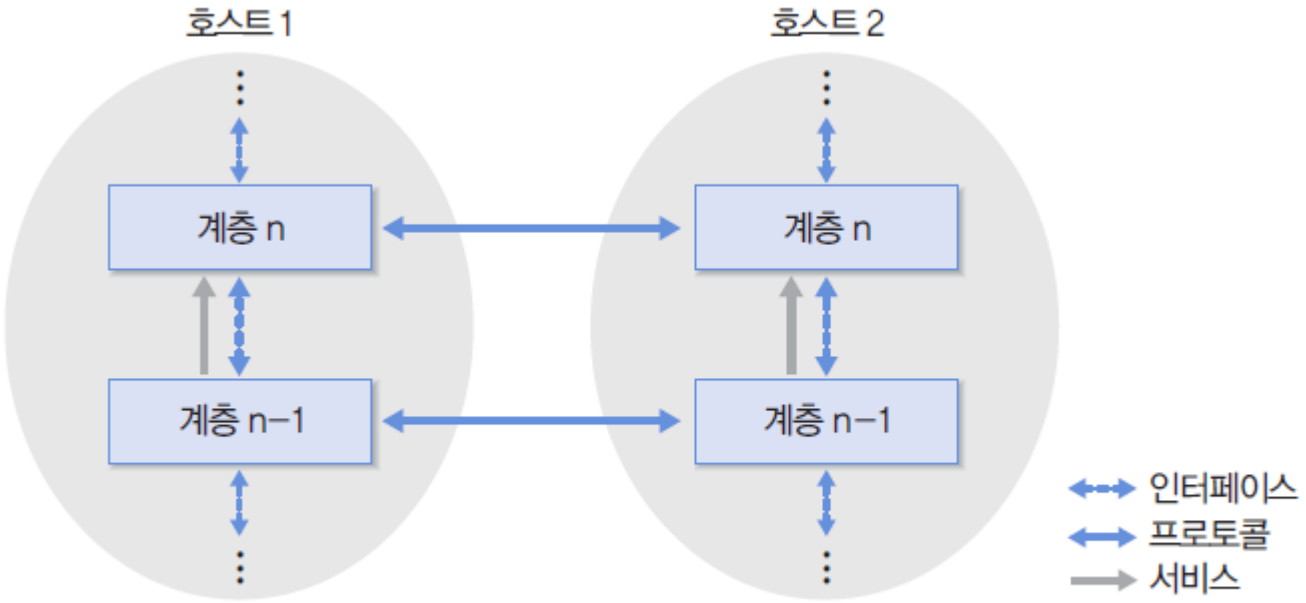


그림 1-5 인터페이스와 프로토콜

- Internet 계층 구조
  - 네트워크 계층(IP 프로토콜), 전송 계층(TCP, UDP 프로토콜)
  - FTP 서비스의 예 [그림 1-6]

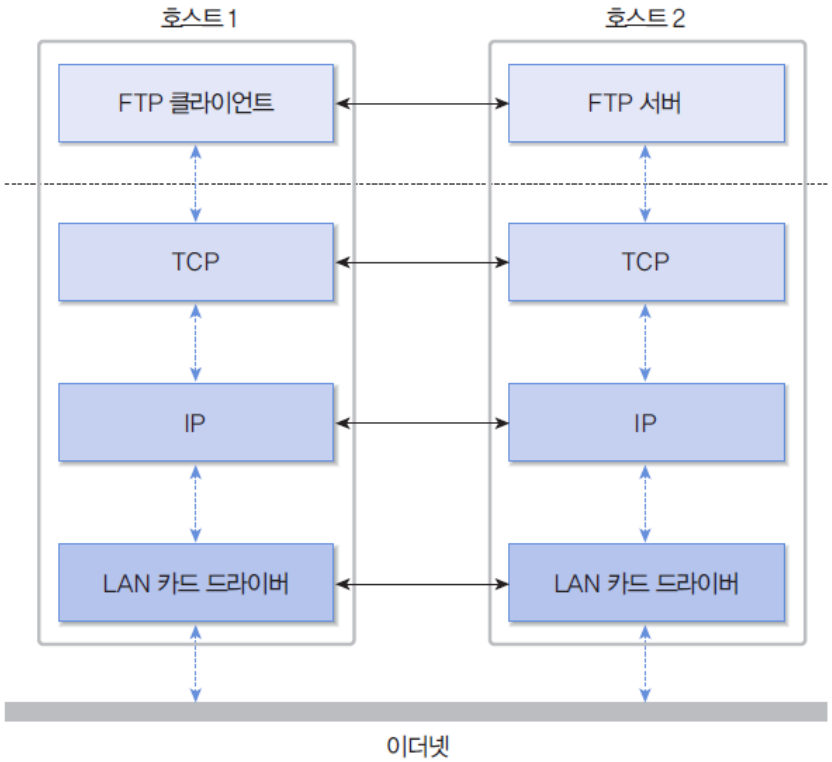


그림 1-6 FTP의 계층 구조

Protocol 예

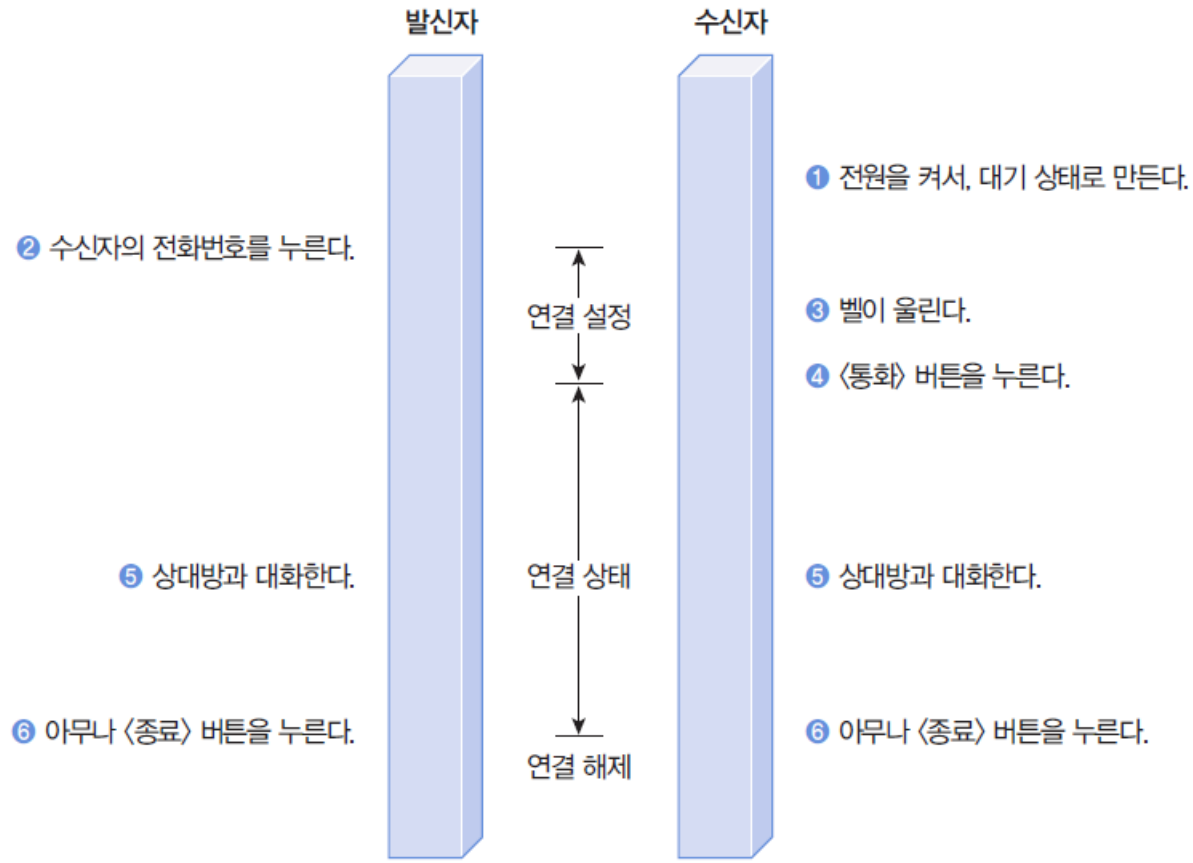


그림 1-8 **전화 연결을 위한 규칙**

IP 주소

- 네트워크에 연결된 컴퓨터를 구분하기 위해 사용
- 32 비트 크기의 주소 체계로, 4개로 구분된 10진수를 사용함.
- IP 주소 부족 문제를 해결하기 위해 IPV6 (128 비트 체계)가 논의됨.
- 211.223.201.30 [그림 1-9]

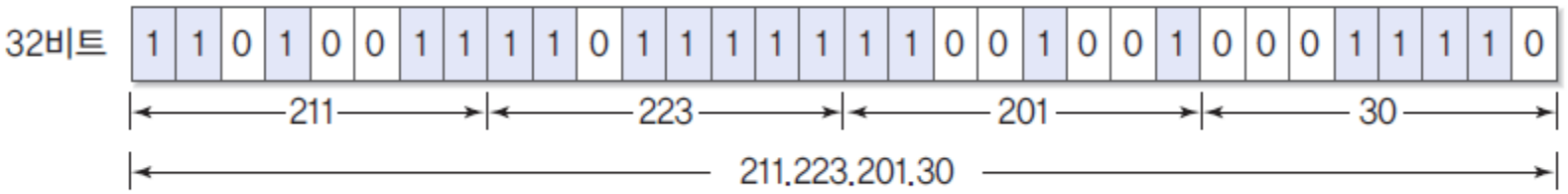


그림 1-9 IP 주소의 표현

호스트 이름

- <호스트>.<단체 이름>.<단체 종류>.<국가 도메인>
- 예: zebra.korea.co.kr

표 1-2 국가 도메인

국가 도메인	해당 국가명
kr	한국
jp	일본
us	미국

표 1-3 단체 종류

단체 종류	기관 성격
co company	회사
ac academy	교육기관
go government	정부 소속 기관

호스트 이름과 IP 주소의 변환

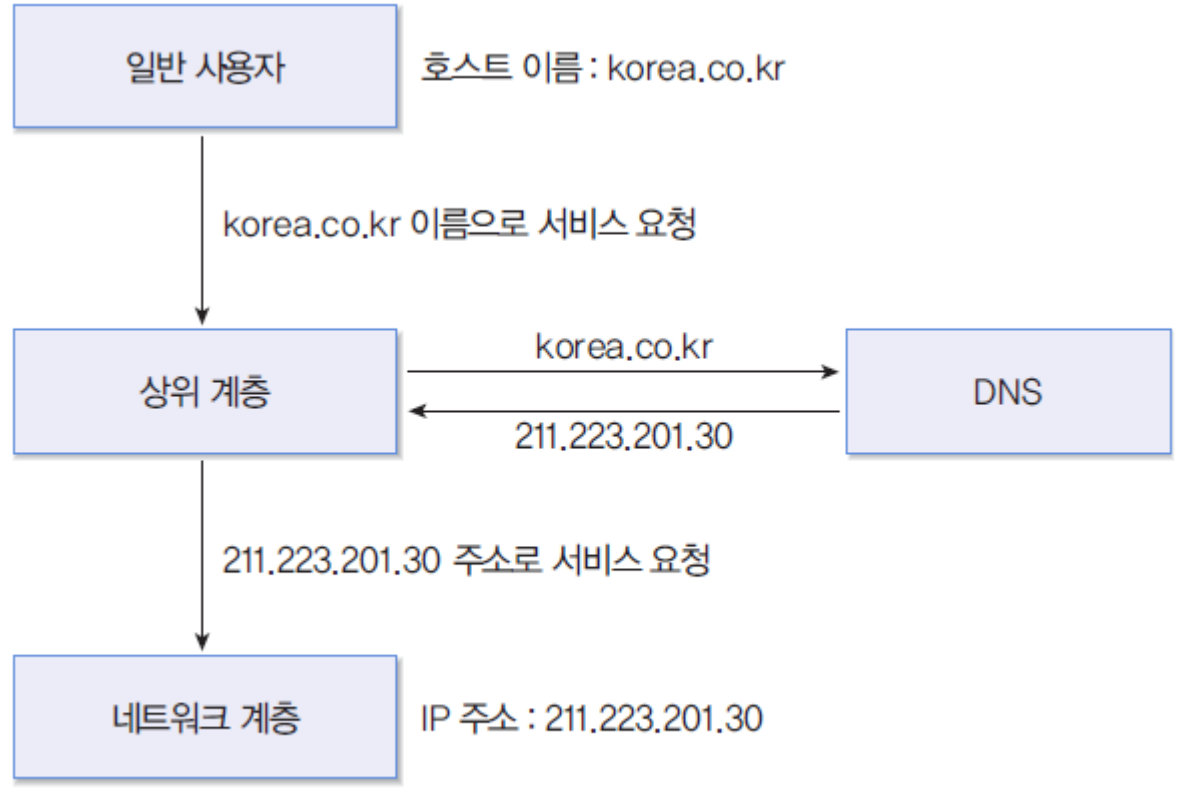


그림 1-11 호스트 이름과 IP 주소의 변환

- 도메인 이름(Domain name)
  - IP 주소를 알기 쉬운 이름으로 바꾼 것
  - DNS(Domain Name System) 서버가 필요함.

DNS 처리 과정

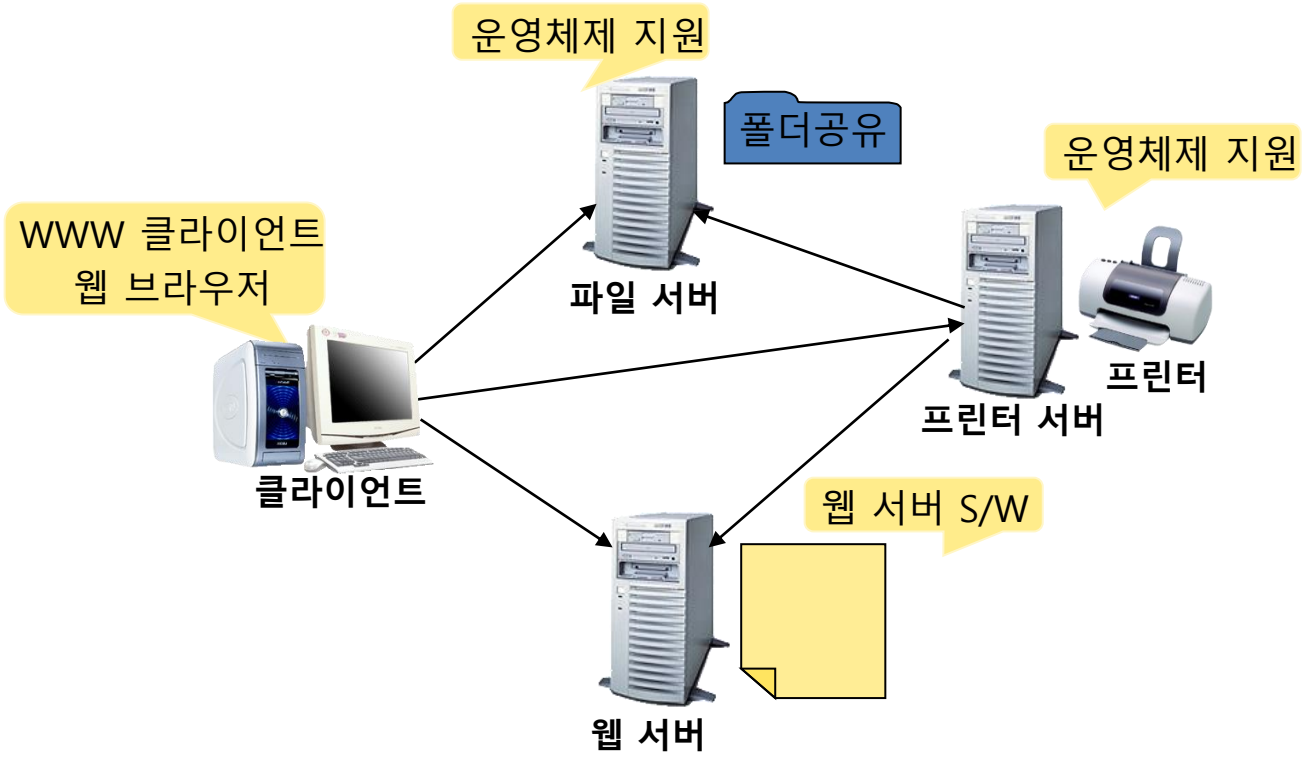
Internet

- Internet과 www
  - 인터넷은 TCP/IP 기반의 네트워크가 전세계적으로 확대되어 하나로 연결된 '네트워크의 네트워크'
  - 인터넷 = www가 아님. www는 인터넷 기반의 서비스 중 하나

이름	프로토콜	포트	기능
www	http	80	웹 서비스
Email	SMTP/POP3/IMAP	25/110/114	이메일 서비스
FTP	ftp	21	파일 전송 서비스
telnet	telnet	23	원격 로그인
DNS	DNS	83	도메인 이름 변환 서비스
News	NNTP	119	인터넷 뉴스 서비스

Internet

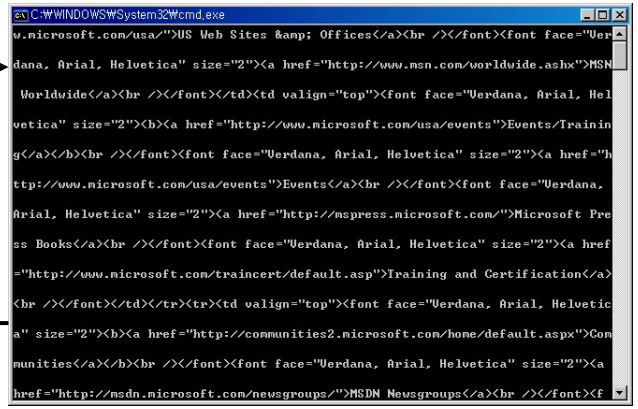
- Web server와 client
  - 서버: 네트워크에서 서비스를 제공하는 컴퓨터
  - 클라이언트: 네트워크에서 서비스를 제공받는 컴퓨터
  - 최근 클라이언트와 서버의 하드웨어적인 구분이 없어지고 있음

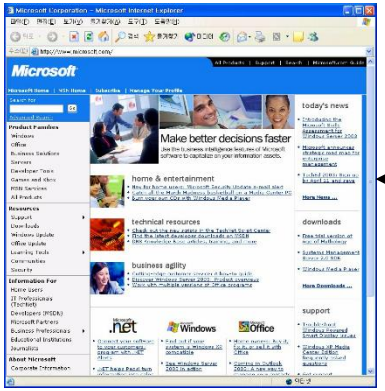


Internet

- HTTP(Hyper Text Transfer Protocol)
  - 프로토콜: 네트워크에 연결된 컴퓨터가 서로 통신(대화)하기 위한 규약
  - HTTP는 www 서비스를 위한 통신 규약
  - 웹 서버와 클라이언트는 HTTP를 이용해 통신
  - HTTP 동작 원리

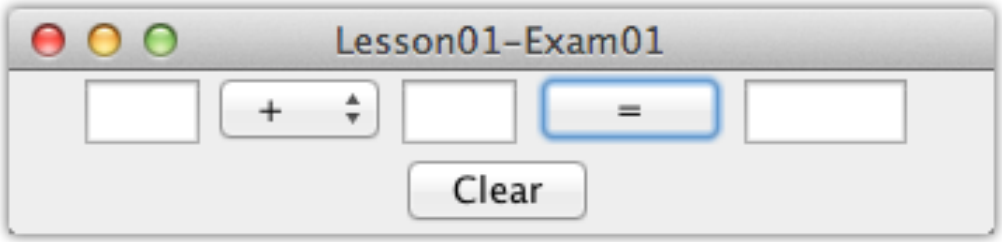
```
c:₩> telnet www.microsoft.com 80
....
GET /index.html
```





Desktop Application

- 계산기 윈도우 애플리케이션 만들기

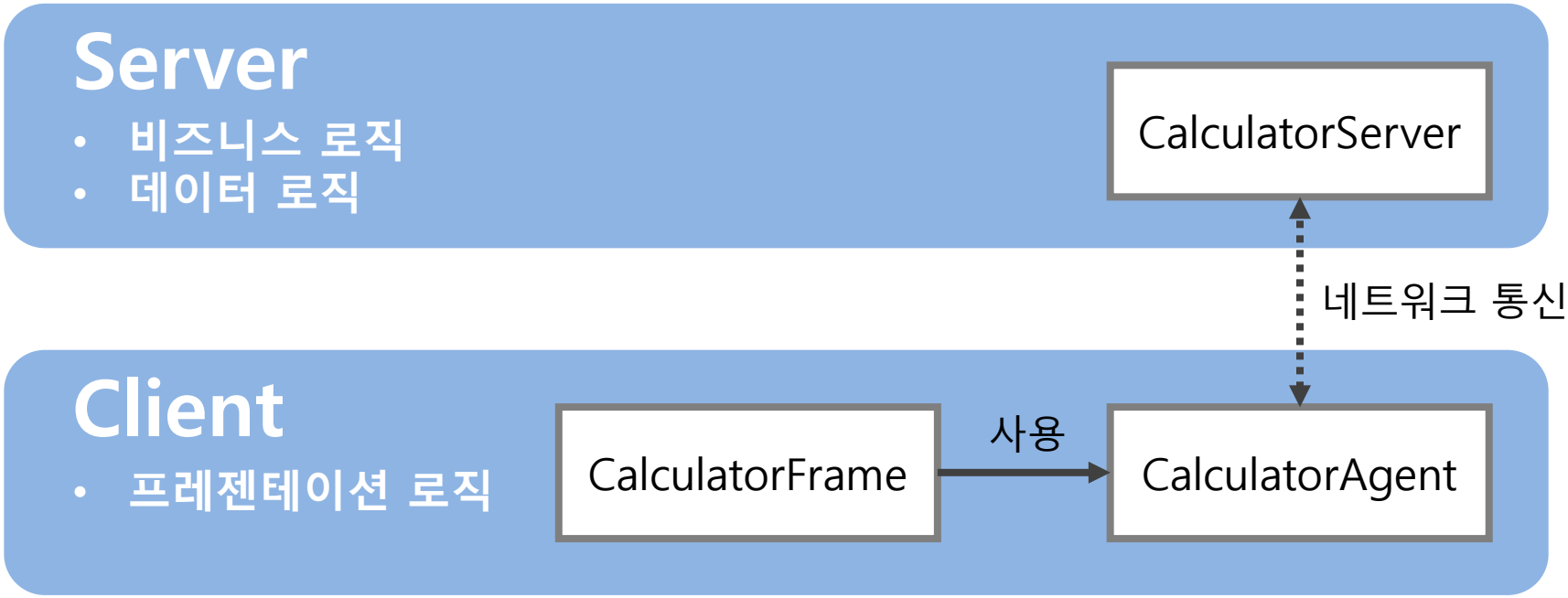


- 프레젠테이션 로직
- 비즈니스 로직
- 데이터 로직

- 특징
  - PC에 설치한 후 실행
  - 사용자 화면 출력(presentation logic), 데이터 보관(data logic), 업무 절차에 따른 일련의 데이터 처리 작업(business logic)을 모두 PC에서 수행 한다
- 문제점
  - 배포가 번거롭고, 보안에 취약하다

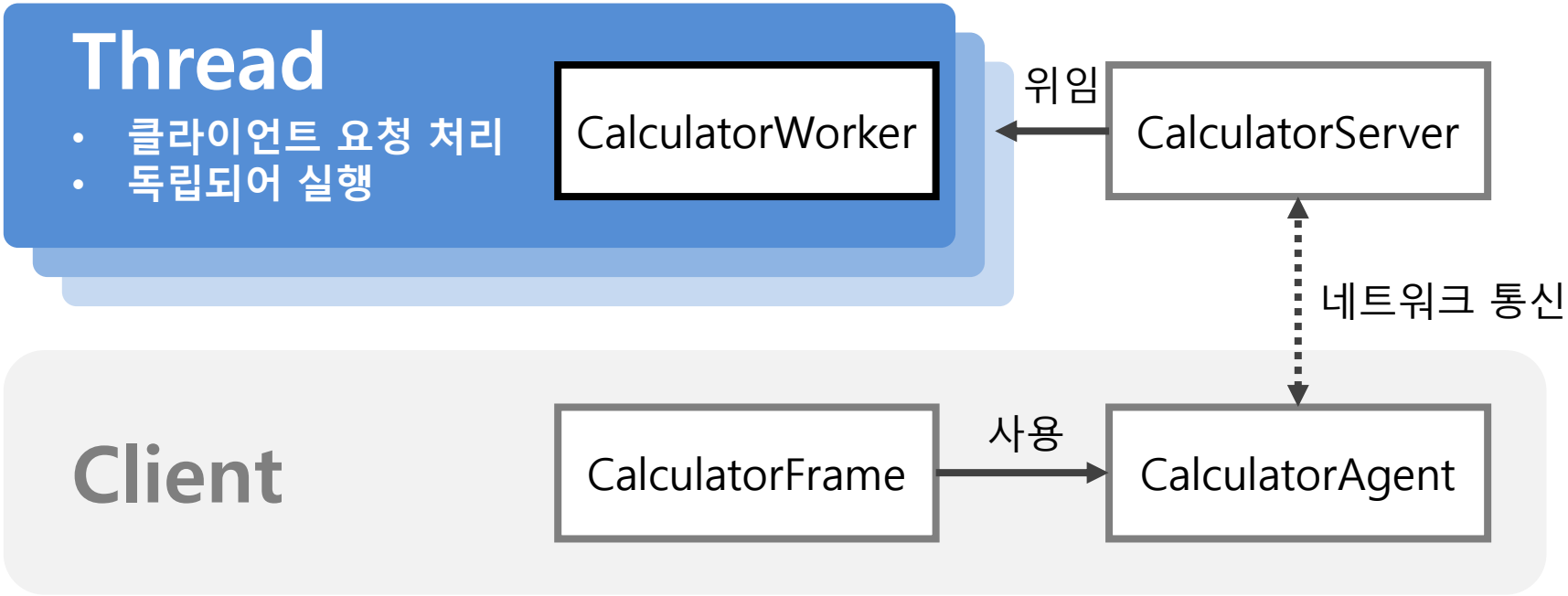
Client/Server Application

- 특징
  - 애플리케이션의 기능을 클라이언트와 서버로 분리한다
    - ✓ 업무 변화에 대응하기 쉽다
    - ✓ 서버 쪽에서 데이터베이스에 접속 → 보안이 강화됨



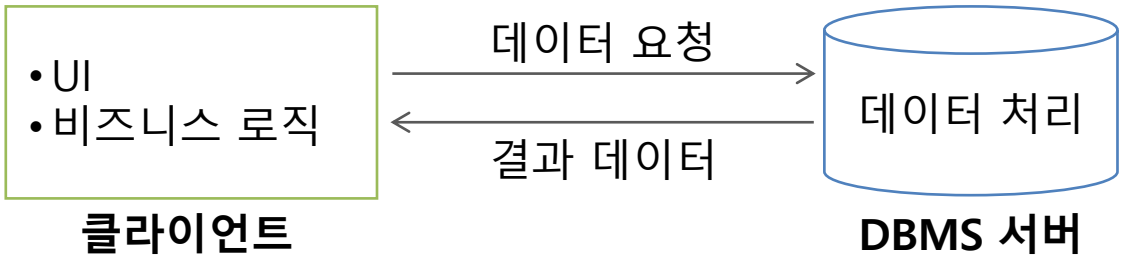
Client/Server Application

- 다중 클라이언트의 요청처리
  - 클라이언트의 요청 처리 부분을 별도의 작업으로 분리한다
  - 분리된 작업은 스레드에 정의한다 (multi-threading 병렬처리)
  - 다중 클라이언트의 요청이 동시에 병행 처리된다

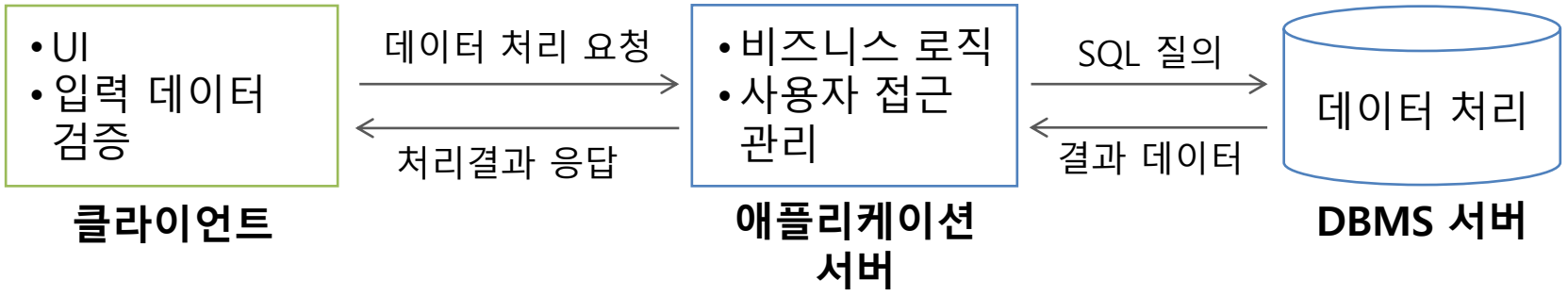


Client/Server Architecture의 진화

- 전통적인 클라이언트·서버 구조
  - 서버는 데이터 처리만 맡는다

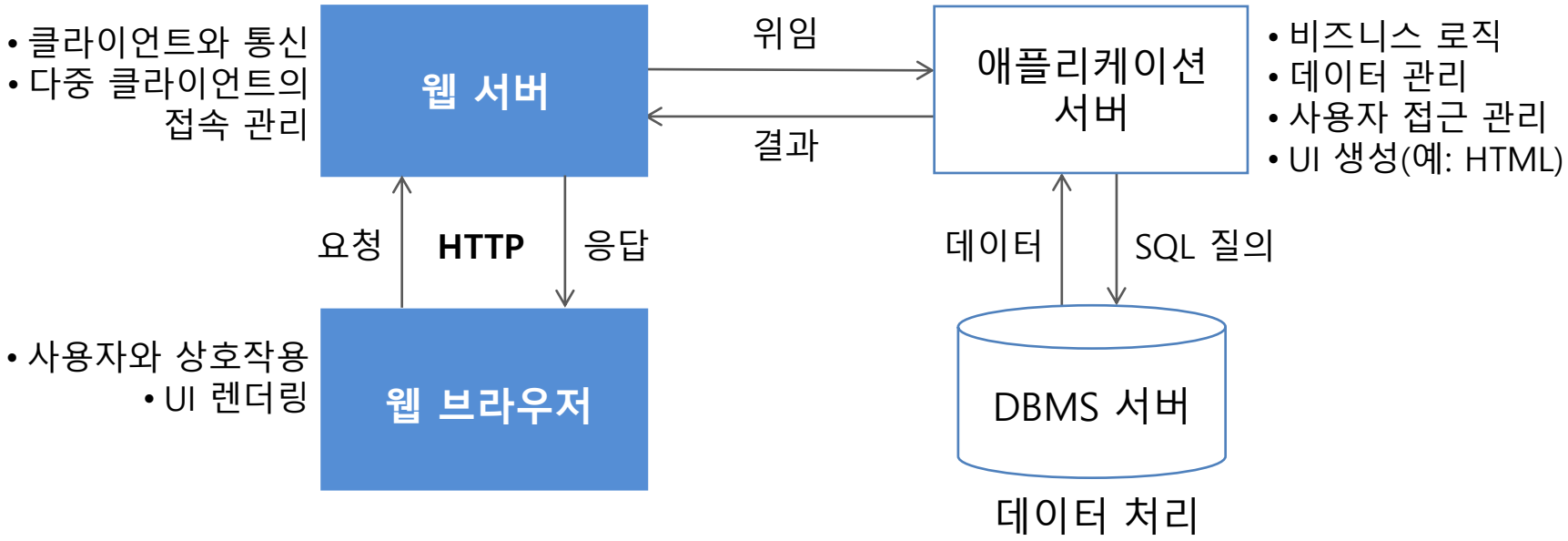


- 클라이언트·서버 구조의 진화
  - 비즈니스 로직을 전문으로 처리하는 서버를 둔다



Web Application Architecture

- 웹 애플리케이션 서버 구조
  - 클라이언트와의 통신은 웹 서버가 전담
    - ✓ 네트워크 및 multi-thread 프로그래밍으로부터 탈출
  - 애플리케이션 서버는 애플리케이션 실행 및 관리에 집중



Web Application Architecture

- 웹 서버
  - 클라이언트와의 통신은 웹 서버가 전담
  - 웹 서버의 기능
    - 리스너 기능 : 클라이언트로부터 접속이 있는지 항상 체크하고 대기
    - 답변 기능 : 요청한 사항을 처리한 후 결과를 클라이언트에 보냄

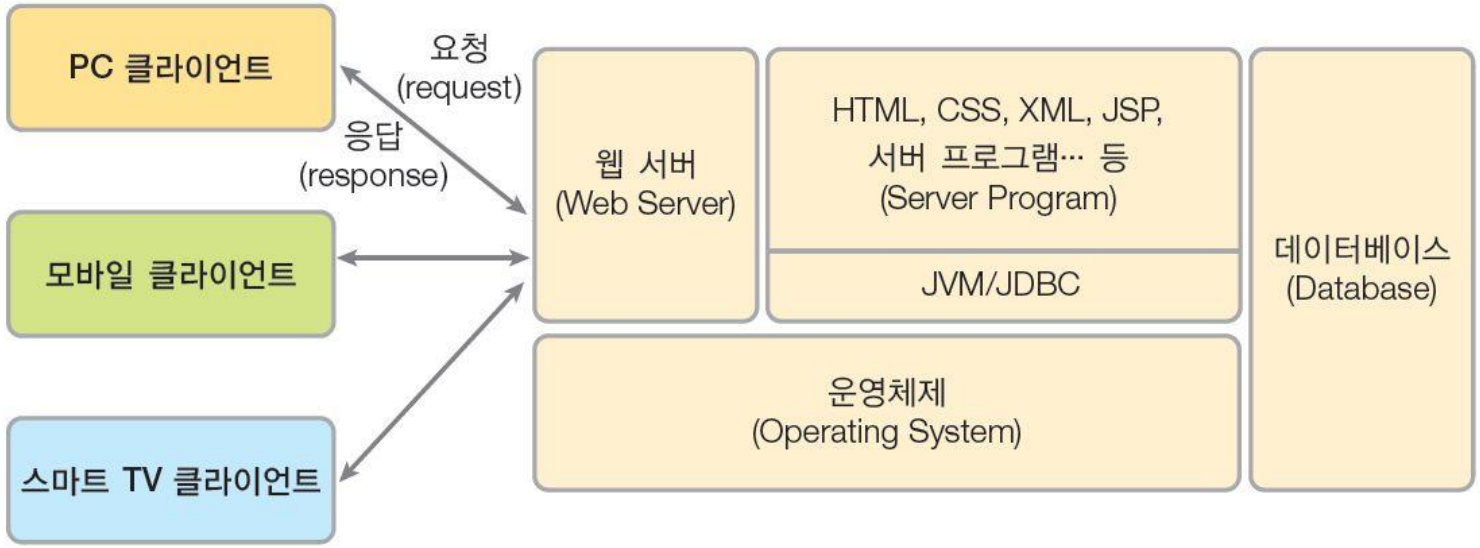
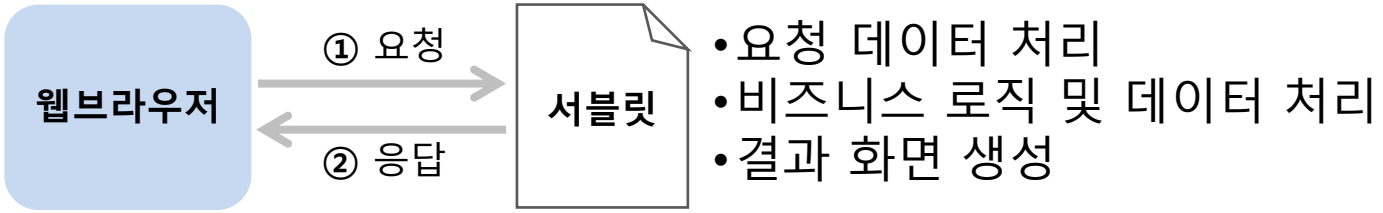


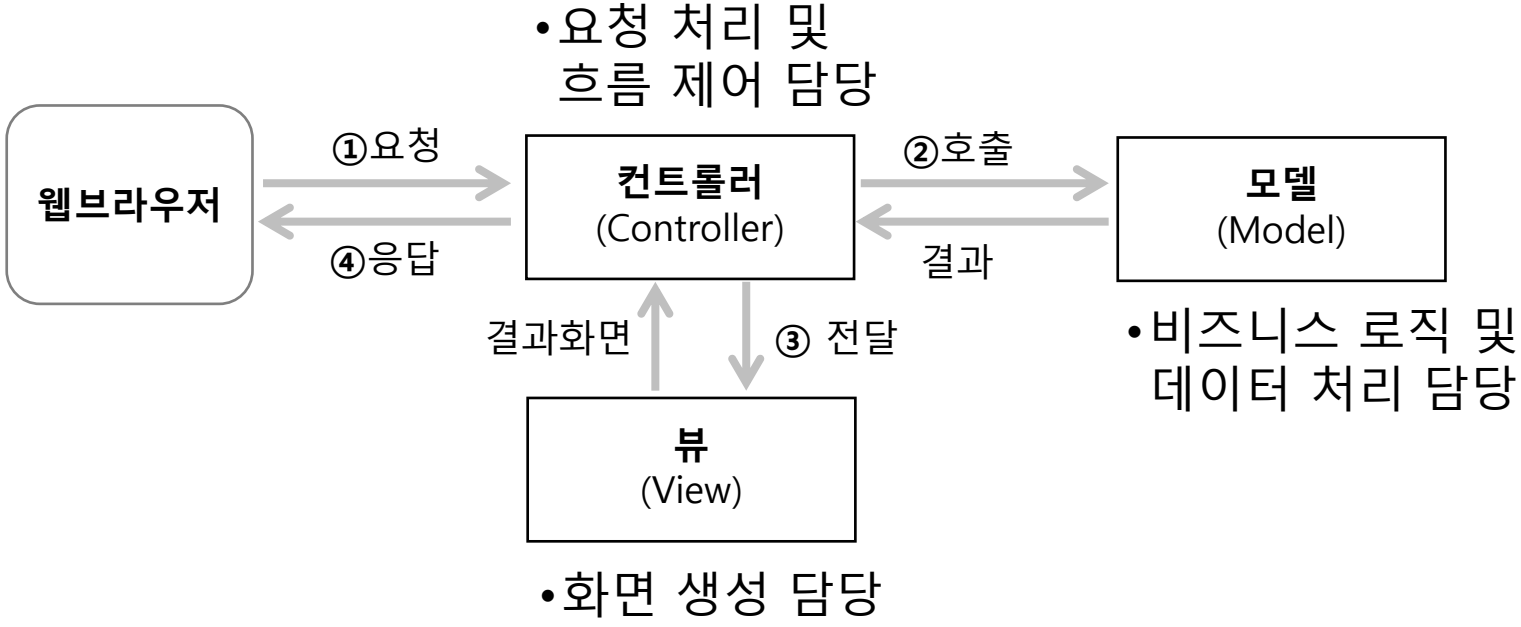
그림 2–2 웹 서버의 서비스 환경

Web Application Architecture

- 올인원(All-In-One) 방식

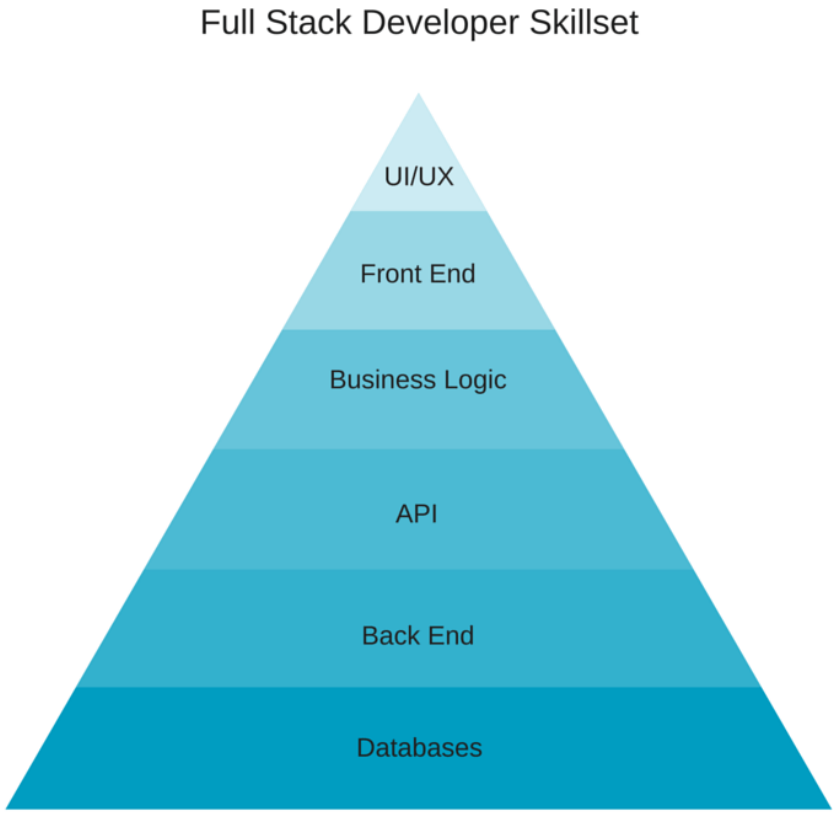


- MVC 아키텍처



Web Application 개발

- Web Full Stack Developer Skillset
  - UI/UX : Web 화면 design
  - API : Frontend와 backend framework 라이브러리



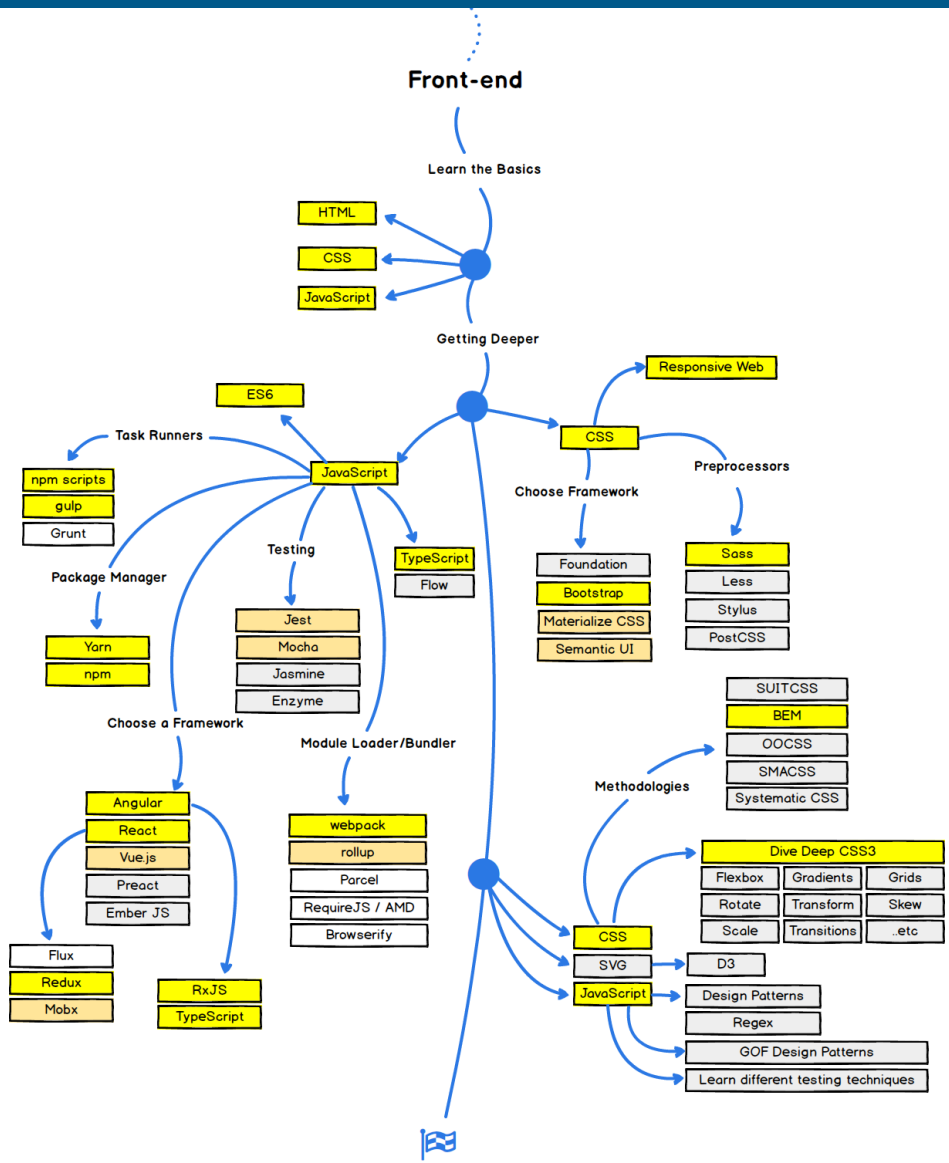
Web Application 개발 Framework - Front-end

- HTML 기술
  - HTML: www 서비스를 표현하기 위해 사용하는 언어
  - www를 통해 서비스하는 모든 내용은 HTML로 표현되어야 함
  - HTML은 텍스트 파일로 정적인 정보만 처리 가능
    → 동적으로 변하는 정보를 처리할 수 없음
  - 동적인 컨텐츠 처리하기 위해 CGI, Fast CGI, PHP, ASP, JSP 등 기술 사용

- Client Script 기술
  - 자바스크립트가 대표적.
  - 웹 브라우저가 스크립트 해석의 주체
  - 웹 브라우저 핸들링은 가능하지만 서버 연동은 불가능

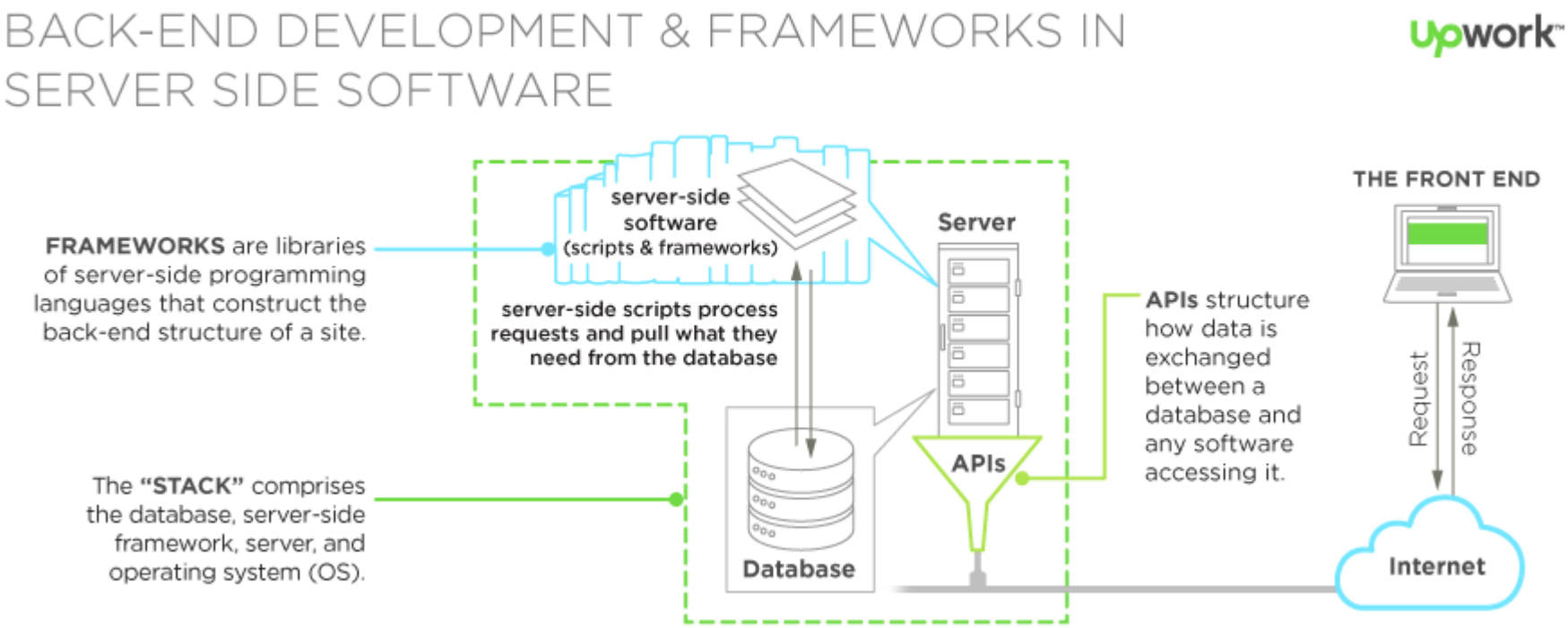
Web Application 개발 Framework - Front-end

- Web Frontend 개발 framework
  - HTML
  - CSS
    - Responsive web
  - JavaScript (ES6)
  - TypeScript
  - JavaScript Framework
    - Angular 2
    - React (flux, redux)
    - Vue.js
  - Package Manager (yarn, npm)
  - Module Loader/Bundler
    - Webpack, Browserify
  - Testing (Jest, Mocha)



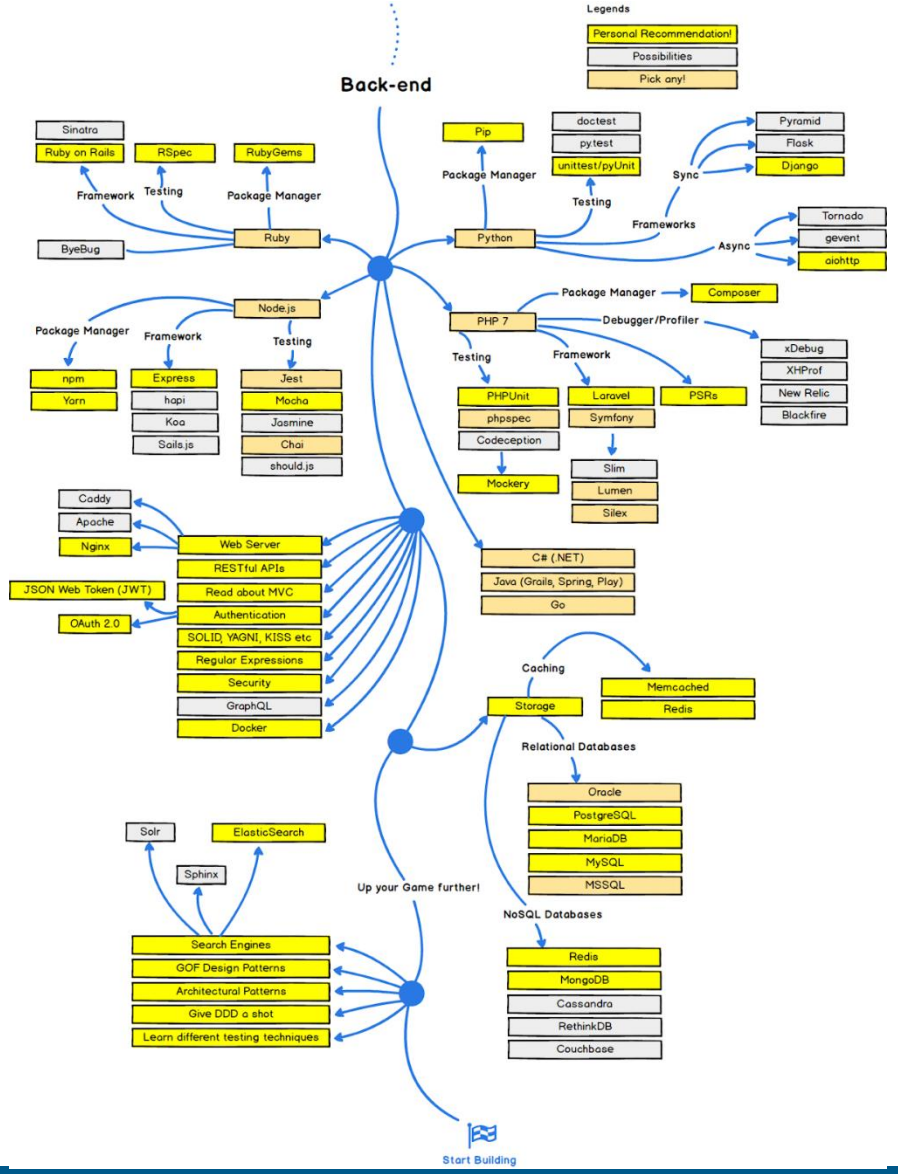
Web Application 개발 Framework – Back-End

- Backend Web Server 어플리케이션 개발 및 운영 환경



Web Application 개발 Framework – Back-End

- Web Backend 개발 framework
  - 개발 Language
    - PHP 7 (Lalavel)
    - Ruby (ruby on rails)
    - Python (Django, Flask)
    - Node.js
    - Java (Spring)
    - C# (.NET)
    - Go
  - Web Server
  - MVC (framework architecture)
  - DBMS(관계형, NoSQL)
  - Restful APIs
  - Authentication & Security
  - Testing



Web Application 개발 Framework

- Popular web application 개발 framework
  - Front-end(web browser)는 javascript 기반 react등이 대세임
    - Angular, jQuery등은 과거보다 인기도가 낮아지는 추세
  - Back-end는 java, python, php가 대세임

Python 기반 Web 개발 Framework

- Full Stack Framework tools
  - HTTP application server 제공하며, database와 연동 library, html template engine, request dispatcher, authentication, ajax 관련 library를 모두 갖고 있는 library 집합을 의미
  - 대표적으로 Django, TurboGears, web2py 등이 있음
- Non full Stack Framework tools
  - HTTP Application server 제공하거나 표준 web application server인 apache등과 연동하여 제공
  - 주로 html template engine 등 필수 library제공하고, 다른 library는 표준 library와 연동 사용함
  - 대표적으로 Bottle, CherryPy, Flask, Hug, Pyramid 등이 있음

수업 실습 환경

- Python 3 install (Anaconda 포함)
- Pycharm Professional IDE tool (Python 언어 기반)
  - Flask, Django framework 제공
  - 학생 License 발급받기 (https://www.jetbrains.com/student)
- Visual Studio Code tool
  - HTML, CSS, JavaScript editor
- MySQL
  - Database 및 Table 생성 등 관리
  - CRUD(Create, Retrieve, Update, Delete) SQL : data 관리 목적