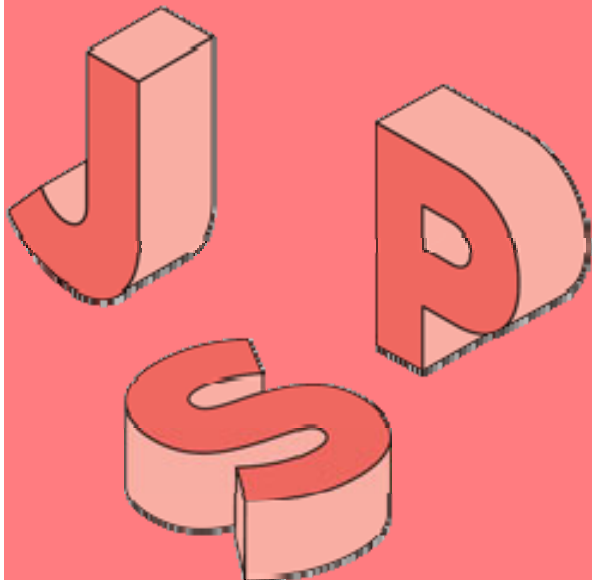


# Chapter 01

## 웹 프로그래밍의 이해



# 목차

1. 웹의 개요
2. 네트워크와 인터넷의 개념
3. 웹 프로그래밍의 구조
4. 새로운 웹 개발 트렌드

# 학습목표

- 인터넷과 웹의 차이점을 이해한다.
- 웹 프로그래밍의 구조와 대표적인 언어를 살펴본다.
- 백엔드와 프론트엔드의 개념을 살펴본다.
- 새로운 웹 개발 트렌드를 살펴본다.

# Section 01

## 웹의 개요

# 웹이란?

- 월드 와이드 웹(World Wide Web)
  - 줄여서 WWW 또는 웹이라고 함
  - 웹은 인터넷에서 운영되는 서비스 중 하나임
    - 따라서 웹 자체가 인터넷을 의미하는 것은 아님
  - 웹의 개발 목적
    - 연구자들이 흩어져 있는 정보를 손쉽게 정리하고 공유하기 위함

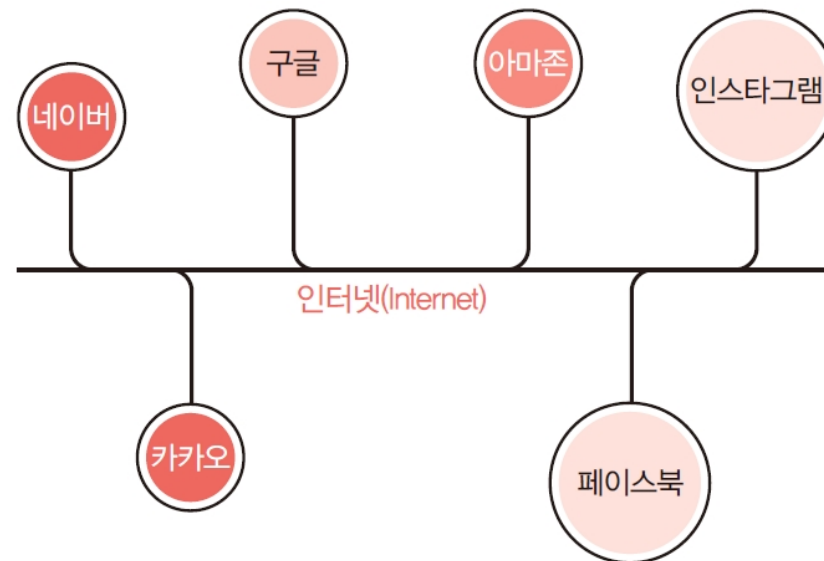


그림 1-1 인터넷과 월드 와이드 웹

# 웹이란?

## ■ 웹의 특징

- 인터넷은 컴퓨터 네트워크망을 의미하고, 웹은 인터넷 서비스 중 하나를 의미함
- 인터넷상의 정보를 하이퍼텍스트 방식과 멀티미디어 환경에서 검색할 수 있게 해주는 정보 검색 시스템을 의미함
- HTTP 프로토콜을 사용하며, HTML 문서를 연결하여 다양한 콘텐츠를 제공함
- 많은 사람들이 정보를 쉽게 공유하고 접근할 수 있도록 하는 것을 목적으로 함

# 웹의 동작 구조

## ■ HTML을 중심으로 한 웹의 동작 과정

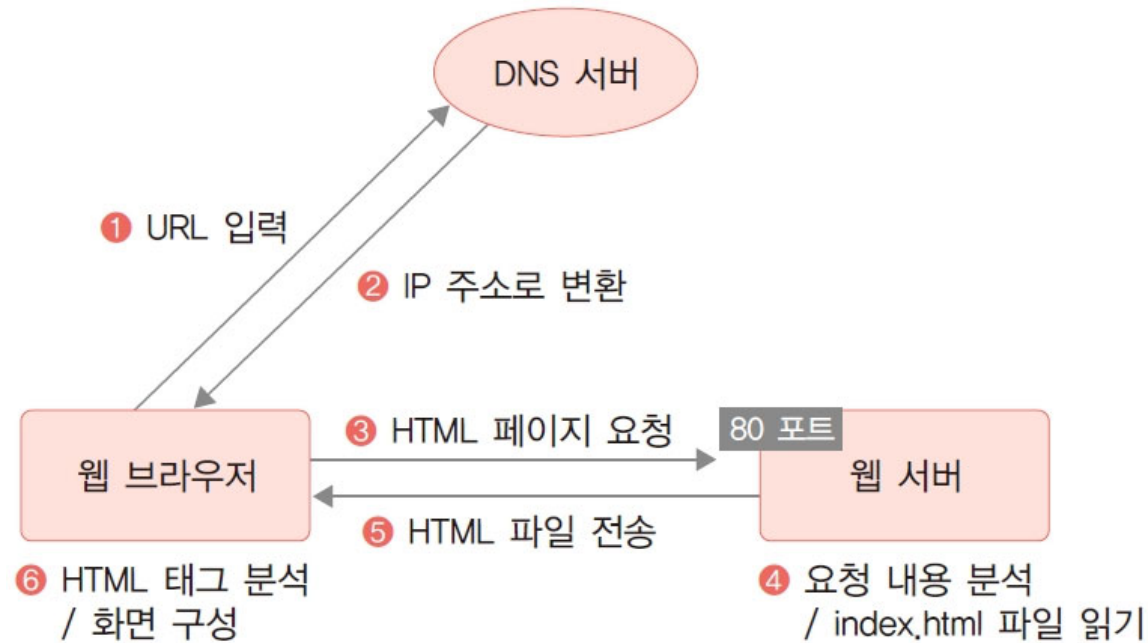


그림 1-2 웹의 동작 구조

## Section 02

# 네트워크와 인터넷의 개념



# 네트워크

## ■ 네트워크(Network)

- 사전적 의미 : 전선이나 혈관, 통로 등으로 이루어진 망형 조직
- 가장 많이 쓰이는 의미 : 컴퓨터와 컴퓨터를 연결해주는 망

## ■ 프로토콜(Protocol)

- 네트워크를 구축하기 위한 컴퓨터 간의 연결 규격
- TCP/IP : 여러 프로토콜 중 가장 널리 쓰이며 인터넷에서도 사용되고 있는 프로토콜

# 네트워크

- TCP/IP(Transmission Control Protocol/Internet Protocol)
  - 컴퓨터 간에 통신할 수 있도록 만든 프로토콜의 종류 중 한 가지
  - 이 프로토콜에 기반하고 있는 것이 인터넷임
  - TCP/IP가 인터넷의 기반 프로토콜로 자리 잡은 이유
    - 하드웨어, 운영체제, 접속 매체와 관계없이 동작할 수 있는 개방형 구조이기 때문

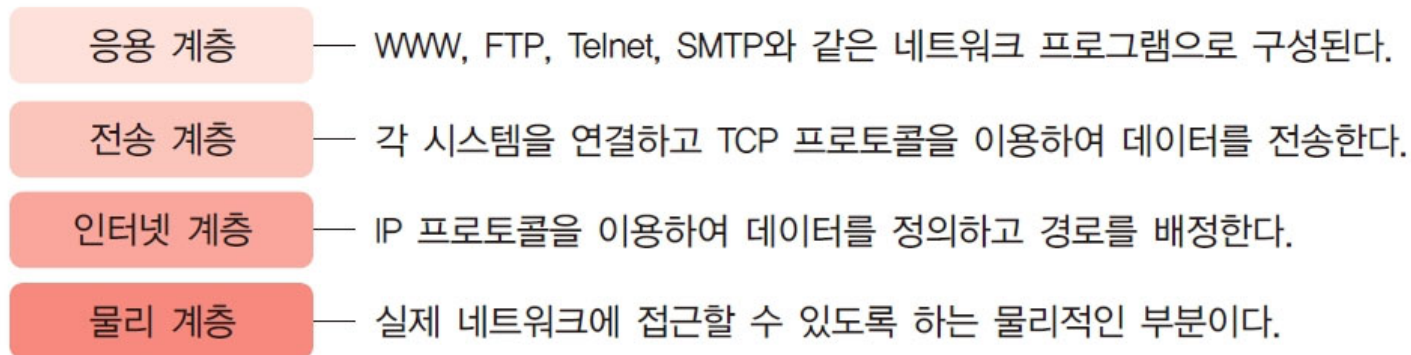


그림 1-3 TCP/IP의 4계층 구조

# 네트워크

## ■ OSI 7계층

- 네트워킹을 위한 물리적 장비에서부터 실제 서비스를 제공하기 위한 애플리케이션에 이르는 단계까지를 계층화한 것을 의미함
- 계층화를 통해 상위 레벨에서는 하위 레벨에서 구현한 내용을 모르더라도 표준화된 인터페이스를 통해 네트워크 시스템을 개발·운영할 수 있음

## ■ TCP/IP의 특징

- TCP/IP는 이러한 OSI 7계층을 좀 더 단순화하여 4계층으로 정의한 것임
- **TCP** : 데이터 흐름 관리, 데이터 정확성 확인 등의 역할을 수행함
- **IP** : 데이터(패킷)를 목적지까지 전송하는 역할을 담당함
- TCP/IP는 개방형 구조로, 특정 운영체제나 하드웨어에 영향을 받지 않고 근거리와 원거리 모두 데이터를 전송할 수 있음

# 네트워크

## ■ IP 주소(IP Address)

- TCP/IP로 연결된 네트워크에서 각각의 컴퓨터를 구분하기 위해 사용하는 주소
- IP 주소는 숫자로 구성되어 있음
- '123.123.123.123'과 같이 4개로 구분되며, 10진수를 사용함
- 사용할 수 있는 IP 주소 범위

표 1-1 IP 주소의 구분과 범위

구분	범위	네트워크 수	사용 목적 / 네트워크당 주소
클래스 A	1.0.0.0 ~ 127.0.0.0	128	대형 통신망 / 16,777,214개( $2^{24}-2$ 개)
클래스 B	128.0.0.0 ~ 191.255.0.0	16,384	중형 통신망 / 65,534개( $2^{16}-2$ 개)
클래스 C	192.0.0.0 ~ 223.255.255.0	2,097,152	소형 통신망 / 256개( $2^8-2$ 개)
클래스 D	224.0.0.0 ~ 239.255.255.255	—	멀티 캐스트용
클래스 E	240.0.0.0 ~ 255.255.255.255	—	실험 목적 / 배포 중지

# 인터넷

- internet과 Internet의 차이점
  - internet은 내부 네트워크를 의미함
  - Internet은 전세계가 하나로 연결된 네트워크를 의미함
  - 즉 인터넷(Internet) 자체는 네트워크 인프라임
  - 우리가 알고 있는 웹, 이메일 등은 인터넷 기반의 서비스인 셈

# 인터넷

## ■ 인터넷 기반 서비스

- 웹 이외에도 이메일, FTP, Telnet, DNS 등이 있지만 일반적으로 사용하는 인터넷 기반 서비스로는 웹이 절대적임
  - 이메일은 네이버, 다음, 구글과 같은 웹 기반의 메일 서비스를 이용하는 경우가 많음

표 1-2 대표적인 인터넷 기반 서비스

서비스 이름	기능	프로토콜	포트
웹www	웹 서비스	HTTP/HTTPS	80/443
이메일Email	이메일 서비스	SMTP/POP3/IMAP	25/110/143
FTP	파일 전송 서비스	FTP	21
Telnet/SSH	원격 로그인 서비스	TELNET/SSH	23/22
DNS	도메인 이름 변환 서비스	DNS	53

# 인터넷

## ■ 프로토콜(Protocol)

- 네트워크에 연결된 컴퓨터들 간의 통신 규약을 의미함
- [표 1-2]의 서비스는 모두 TCP/IP를 사용하며, TCP/IP 4계층 중에서도 응용 계층에 해당하는 프로토콜임
- 즉 물리적인 네트워크 연결은 TCP/IP를 사용하지만, 응용 계층에 서는 서비스마다 별도의 규격이 필요함

## ■ 포트(Port)

- 네트워크 서비스를 제공하기 위한 일종의 출입문임
- 하나의 컴퓨터에서 여러 개의 네트워크 서비스를 제공하는 경우 이들을 구분하기 위한 목적으로 사용함
  - 예시) 은행에서 입출금, 대출, 신용카드, 보험 등 업무에 따라 창구를 구분해놓기
- 포트와 프로토콜이 일치하지 않는 경우 정상적인 네트워크 서비스를 이용할 수 없음

# 인터넷

## ■ 도메인 네임 시스템(Domain Name System)

- 컴퓨터의 IP 주소는 네트워크에서 컴퓨터를 구분해주는 주민번호와 같은 개념
- 인터넷 주소의 형태 : **호스트(컴퓨터) 이름(www) + 도메인 이름(hanbit.co.kr)**
- 도메인 이름은 규칙에 따라 붙여지며, 도메인을 관리하는 기관에 일정 비용을 지불하고 구매해야 함
- 도메인 이름을 통한 웹 사이트 접속도 겉으로는 보이지 않지만 내부적으로 도메인 이름을 IP 주소로 변환하는 서비스를 이용함

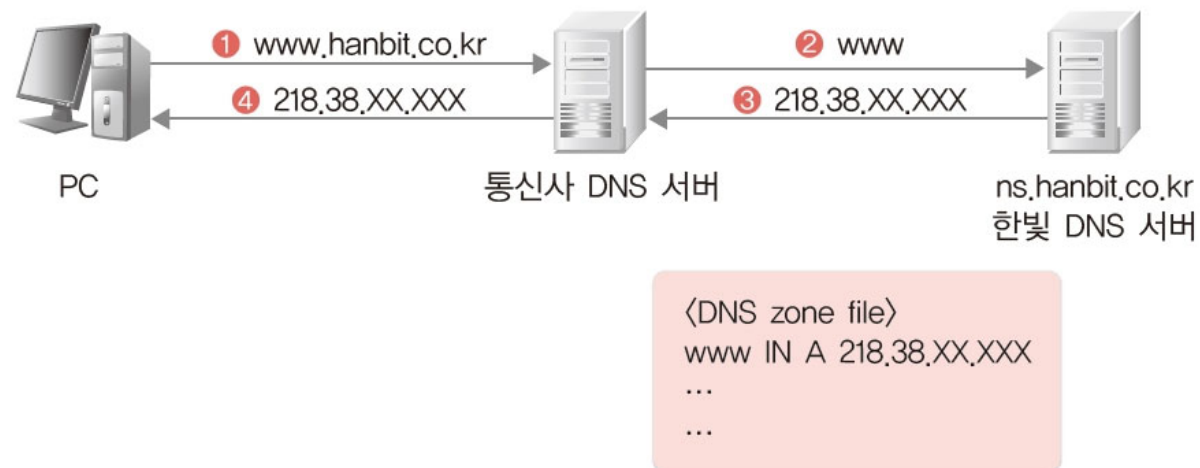


그림 1-4 DNS 동작 구조



## Section 03

# 웹 프로그래밍의 구조

# 클라이언트-서버 구조

## ■ 클라이언트-서버 구조

- 웹 서비스 제공을 위해서는 서버 컴퓨터와 서버에서 동작하는 프로그램이 필요함

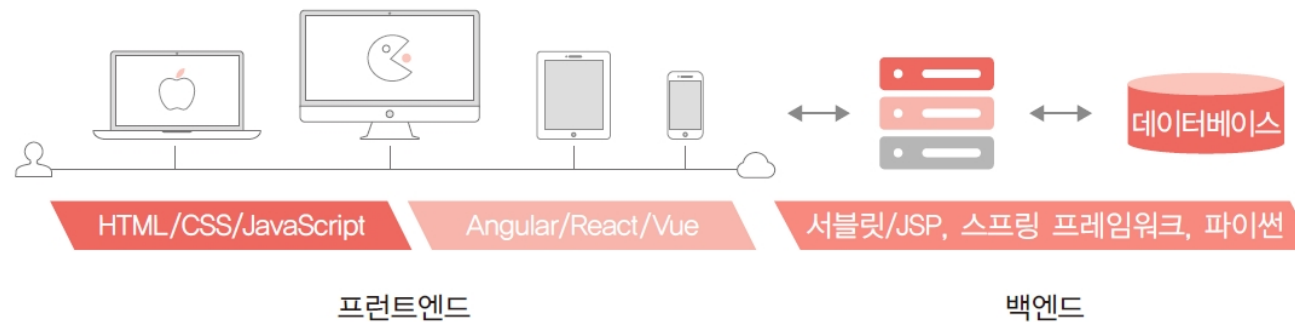


그림 1-5 웹 프로그래밍의 구조

- **클라이언트** : 웹 서비스를 이용하는 사용자
- **서버** : 웹 서비스를 제공하기 위한 서비스 공급자
- **프론트엔드** : 클라이언트 중심의 프로그래밍(HTML, CSS, 자바스크립트) 영역
- **백엔드** : 서버를 구성하며 서비스를 제공하기 위한 서버 쪽 프로그래밍(웹 서버 소프트웨어, 웹 프레임워크, 데이터베이스) 영역

# HTML/CSS/자바스크립트

- HTML(HyperText Markup Language)
  - 웹 문서의 구조를 정의하고 콘텐츠를 표현하는 기본 마크업 언어
  - 웹을 통해 콘텐츠를 제공하려면 HTML을 사용해야 함
  - HTML은 태그라고 하는 간단한 표기법으로 표현하고자 하는 정보에 의미를 부여함

```
<h2>안녕하세요?</h2>  

```

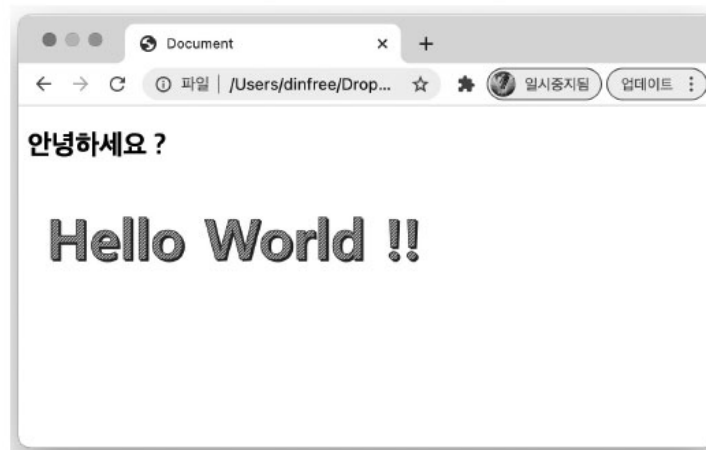


그림 1-6 HTML 태그를 사용한 실행 결과

# HTML/CSS/자바스크립트

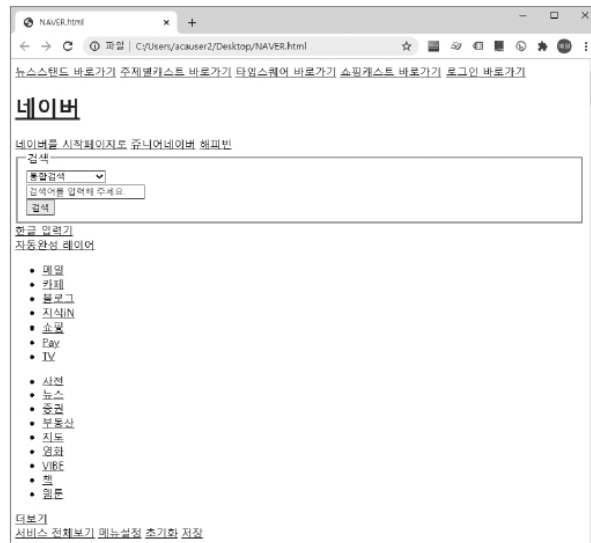
## ■ CSS(Cascading Style Sheet)

- HTML 문서에 레이아웃과 디자인을 포함한 시각적 요소를 정의하기 위한 규격
- HTML은 콘텐츠의 내용과 구조(레이아웃)를 정의하고  
CSS에서 콘텐츠의 위치, 정렬, 글 자 크기, 여백, 배경 색상 등을 정의함
- 디자인을 적용할 HTML 요소를 선택터(Selector)로 지정하고 원하는 디자인 속성을 부여하는 형식

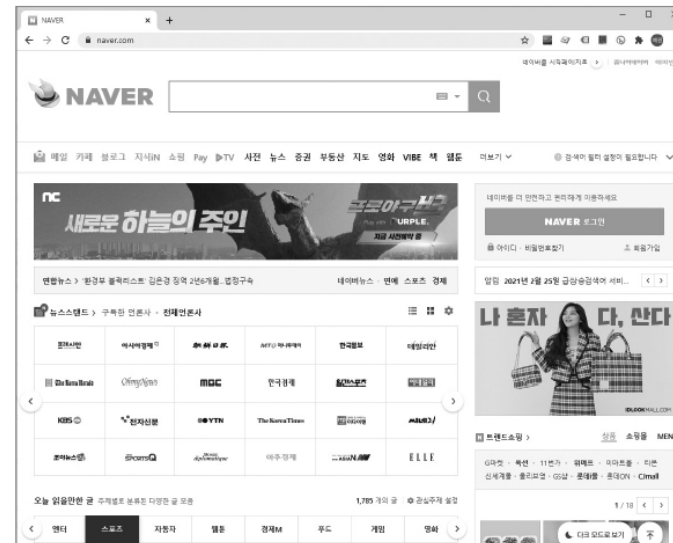
# HTML/CSS/자바스크립트

## ■ 자바스크립트(Javascript)

- HTML 문서에서 동적으로 변하는 콘텐츠를 표현하기 위해 이벤트 처리를 하거나 서버와 연결하여 데이터를 가지고 오는 등의 역할을 하는 프로그램 언어
- HTML 문서에서 이벤트를 서버 연결 없이 동적으로 처리할 수 있는 기술임
  - 즉 서버에 새로 접속하지 않고도 화면 정보가 변하는 기능을 함
- 대표적인 자바스크립트 라이브러리(프레임워크)
  - 앵귤러(Angular.js), 뷰(Vue.js), 리액트(React.js) 등



(a) HTML로만 이루어진 화면



(b) CSS, 자바스크립트가 적용된 화면

그림 1-7 HTML, CSS, 자바스크립트가 적용된 실제 화면 예시

# 백엔드 중심 개발과 프론트엔드 중심 개발

## ■ 백엔드 중심 개발

- 전통적인 웹 개발 모델이며, 서버에서 모든 것을 담당하는 방식임
- 특히 자바 서블릿/JSP는 가장 인기 있는 백엔드 개발 기술임

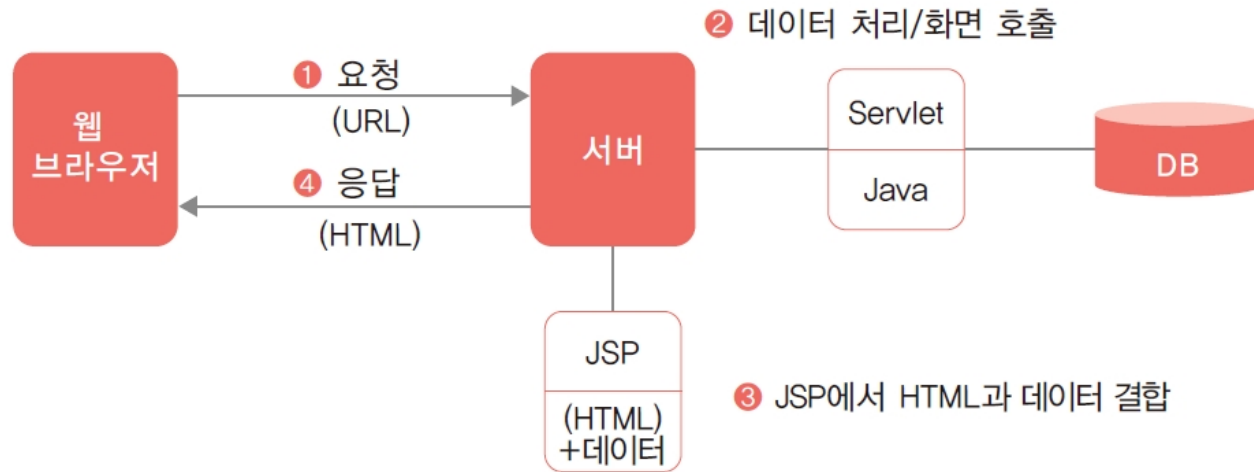


그림 1-8 백엔드 동작의 개요

# 백엔드 중심 개발과 프론트엔드 중심 개발

## ■ 백엔드 중심 개발의 장점

- 서비스 연동에 필요한 다양한 서버 환경에 대응할 수 있음
- 검색 엔진 최적화에 유리함
- 기술이 안정적이고 검증됨
- 기존에 개발된 시스템이 많고 레거시 시스템은 오랫동안 유지됨

## ■ 백엔드 중심 개발의 단점

- 모바일 네트워크의 속도가 느리며 이용에 많은 비용을 지불해야 함
- 서버에 화면 갱신을 요청할 경우 모든 데이터가 다시 전송되어야하므로 모바일 환경에는 부적합함
- REST API와 클라우드 인프라가 보편화되면서 기존의 대규모로 서버를 구축하는 모놀리식 아키텍처 방식보다는 소규모 서버를 연동하는 MSA 방식이 확산되고 있음

# 백엔드 중심 개발과 프론트엔드 중심 개발

## ■ 프론트엔드 중심 개발

- 클라이언트에서 HTML을 가지고 있거나 서버에서 HTML만 받아오고 서버로부터 화면 구성에 필요한 데이터만 자바스크립트로 받아와 데이터와 화면을 조합해 보여줌
- 이러한 방식을 CSR(Client Side Rendering)이라고도 함

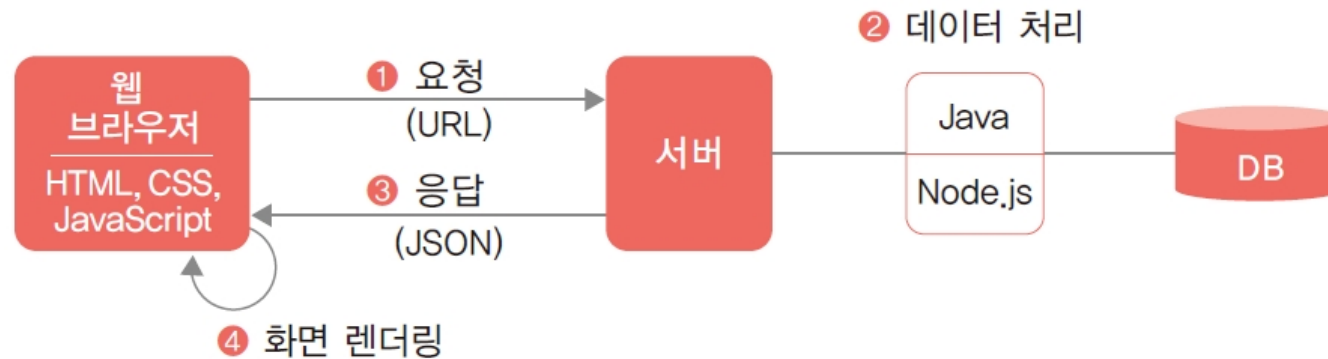


그림 1-9 프론트엔드 동작의 개요



# 백엔드 중심 개발과 프론트엔드 중심 개발

## ■ 프론트엔드 중심 개발의 장점

- 필요한 부분의 데이터만 갱신이 가능하기 때문에 서버로부터 매번 갱신된 전체 화면을 받아올 필요가 없음
- 실시간 데이터 갱신이 자유로움
- SPA(Single Page App), PWA(Progressive Web App) 등의 구현에 적용할 수 있음
- React.js, Vue.js 등 다양한 라이브러리/프레임워크를 사용할 수 있음

## ■ 프론트엔드 중심 개발의 단점

- 프론트엔드 중심의 개발이라 하더라도 데이터 제공을 위한 서버는 필요함
- 데이터 제공 서버는 주로 REST API로 개발되기 때문에 백엔드 작업은 당연히 존재함
- 프론트 엔드 중심 개발을 하더라도 콘텐츠가 검색 엔진에 노출될 수 있도록 검색 엔진 최적화를 위해 SSR(Server Side Rendering)을 접목하기도 함

## Section 04

# 새로운 웹 개발 트렌드

# 새로운 웹 개발 트렌드

## 1. 자바 언어의 대안 등장

- 오라클과 구글의 자바 라이선스 문제가 오라클의 승리로 끝남
- 구글은 새로운 안드로이드 개발 언어로 젯브레인스에서 개발한 코틀린(Kotlin) 언어를 추가하면서 안드로이드 개발 트렌드는 점점 코틀린 쪽으로 이전하고 있음
- 코틀린은 최신 프로그래밍 언어의 특징을 모두 가지는 모던 프로그래밍 언어 (Modern Programming Language)임

# 새로운 웹 개발 트렌드

## 2. 백엔드 개발 및 운영 환경의 변화

- 스프링 프레임워크가 등장하게 됨
- 클라우드 기반의 서버 운영이 보편화되면서 전통적인 모놀리식 아키텍처 중심의 서버 운영 전략이 소규모 분산 서비스 운영 방식인 MSA로 전환되기 시작함
- 프론트엔드 중심의 개발이 가속화되면서 서버 프로그램이 보다 간결한 형태로 전환되고 Node.js, 파이썬을 이용한 서버 프로그램 개발이 늘어나고 있음
- 특히 REST API 형태의 개발이 주를 이루게 되면서 백엔드에서 화면을 다루어야 하는 경우는 점점 줄어들고 있음
- 구글의 파이어베이스(Firebase)를 비롯한 클라우드 기반의 서비스 인프라가 보편화되어 필요한 데이터를 REST API 형태로 손쉽게 제공받을 수 있는 서버리스 환경도 보편화되고 있음

# 새로운 웹 개발 트렌드

## 3. 프론트엔드 중심 개발로의 이동

- 웹에서 모바일로 이동하면서 데이터를 서버에서 조합(바인딩Binding)한 다음 화면을 구성(렌더링Rendering)하는 서버 사이드 렌더링(SSR)은 방식에서 여러 문제가 부각되기 시작함
  - 예시1) 모바일에서 인터넷을 사용하려면 데이터 요금을 부담해야 하고 유선에 비해 상대적으로 속도도 느림
  - 예시2) 인터넷 접속이 항상 보장되지 않으며, 실시간으로 데이터 갱신은 필요함
- 이러한 새로운 요구는 서버로부터 데이터만 수신하고 클라이언트(웹 브라우저)에서 데이터를 조합한 다음 화면을 구성하는 클라이언트 사이드 렌더링(CSR) 방식으로 개발을 전환하게 되는 계기가 됨

# 새로운 웹 개발 트렌드

## 4. 스프링 프레임워크

- 스프링 프레임워크(Spring Framework, 이하 스프링)는 상용화할 수 있는 수준의 대규모 서비스를 누구라도 쉽게 개발할 수 있도록 해줌
- 자바 기반의 대표적인 백엔드 개발 프레임워크로 국내에서는 절대적인 위치에 있음
- 전자정부 프레임워크 역시 스프링 기반임
- 스프링을 쉽게 사용할 수 있게 하는 스프링 부트(Spring Boot)의 경우 JSP 사용에 제약이 있음