
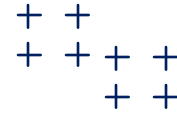
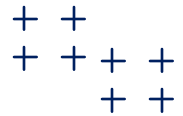


전북대 
소중해유(You)

Javascript 기초 문법 및 기본 구조 이해



전북대학교
SOFTWARE중심대학사업단

SW중심대학



목차

1 / Javascript란?

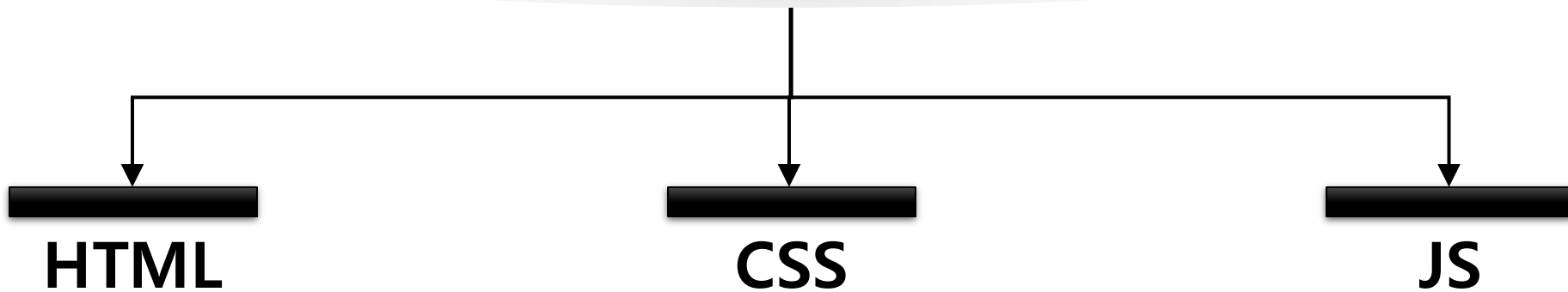
2 / 백엔드 서비스의 구조

3 / 웹서비스의 구성

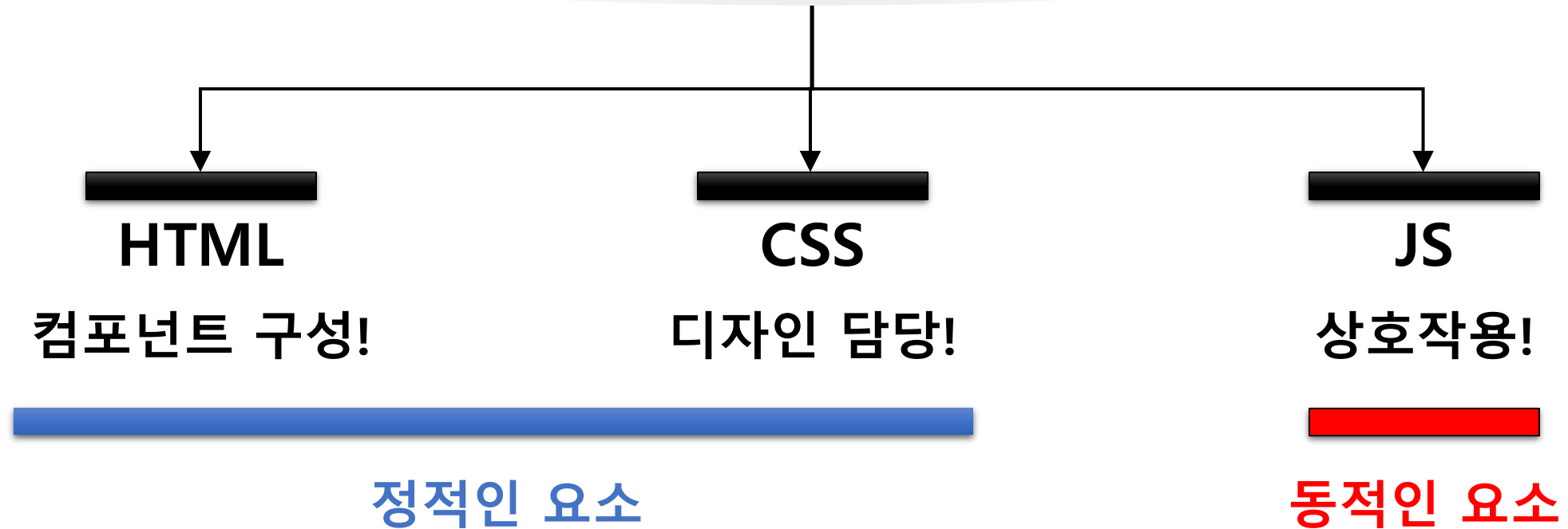
| Javascript란?

Javascript \neq Java

웹페이지의 기본 구성 요소



웹페이지의 기본 구성 요소



Javascript란?

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Sample Page</title>
</head>
<body>
</body>

<script>
</script>

<style>
</style>
</html>
```

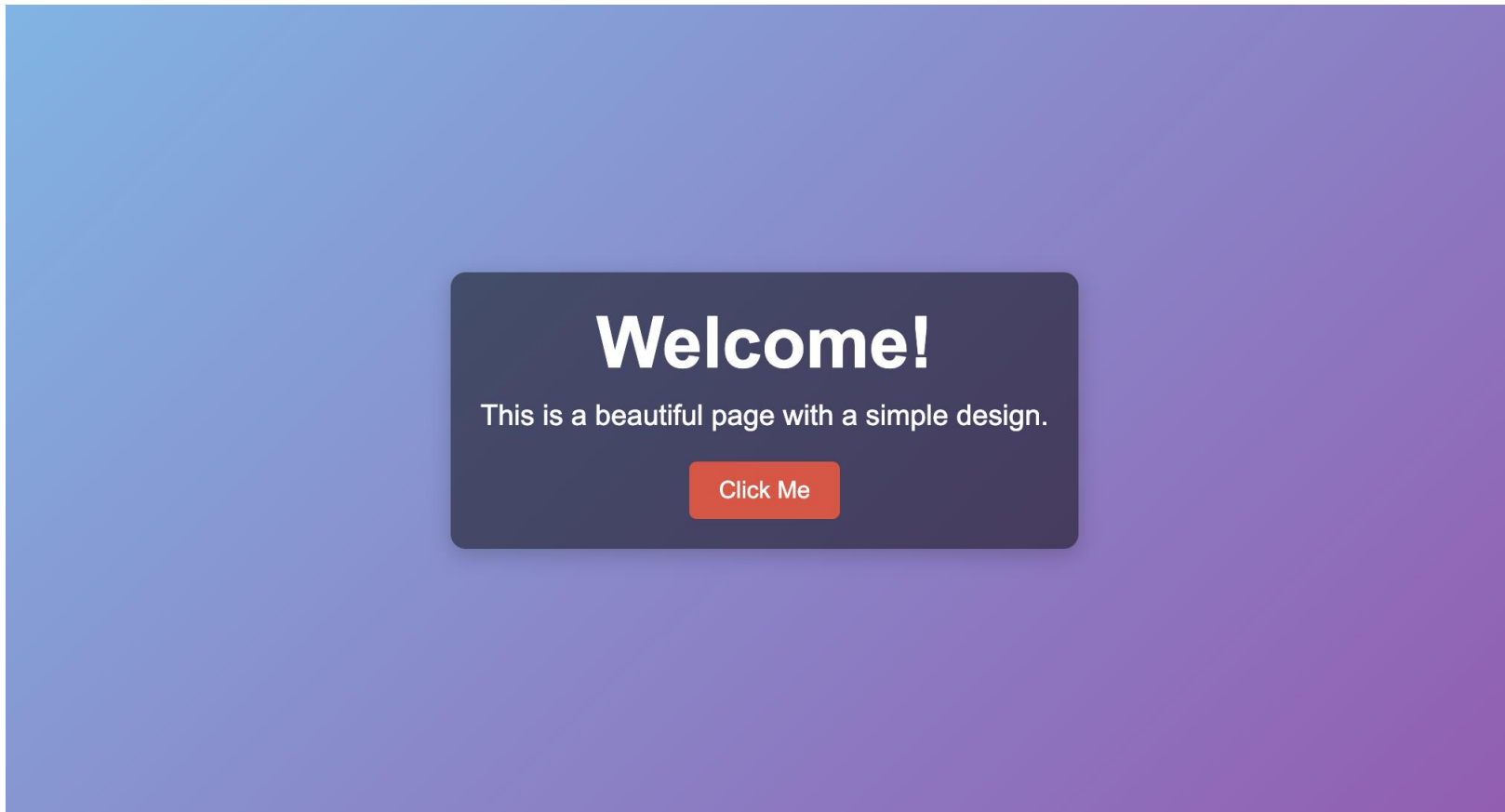
* Sample HTML File

HTML Component가 들어가는 부분: "컴포넌트 생성 담당!"

Javascript가 들어가는 부분: "컴포넌트의 동작 담당!"

Style sheet (CSS)가 들어가는 부분: "컴포넌트의 디자인 담당!"

Javascript란?



Sample Page

Javascript란?

```
<style>
  body {
    font-family: 'Arial', sans-serif;
    margin: 0;
    padding: 0;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    background: linear-gradient(135deg, #71b7e6, #9b59b6);
    color: #fff;
  }
  .container {
    text-align: center;
    padding: 20px;
    background: rgba(0, 0, 0, 0.5);
    border-radius: 10px;
    box-shadow: 0 4px 15px rgba(0, 0, 0, 0.2);
  }
  h1 {
    margin: 0;
    font-size: 3rem;
  }
  p {
    margin: 10px 0 20px;
    font-size: 1.2rem;
  }
  .button {
    padding: 10px 20px;
    font-size: 1rem;
    color: #fff;
    background: #e74c3c;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    transition: background 0.3s ease;
  }
  .button:hover {
    background: #c0392b;
  }
</style>
```

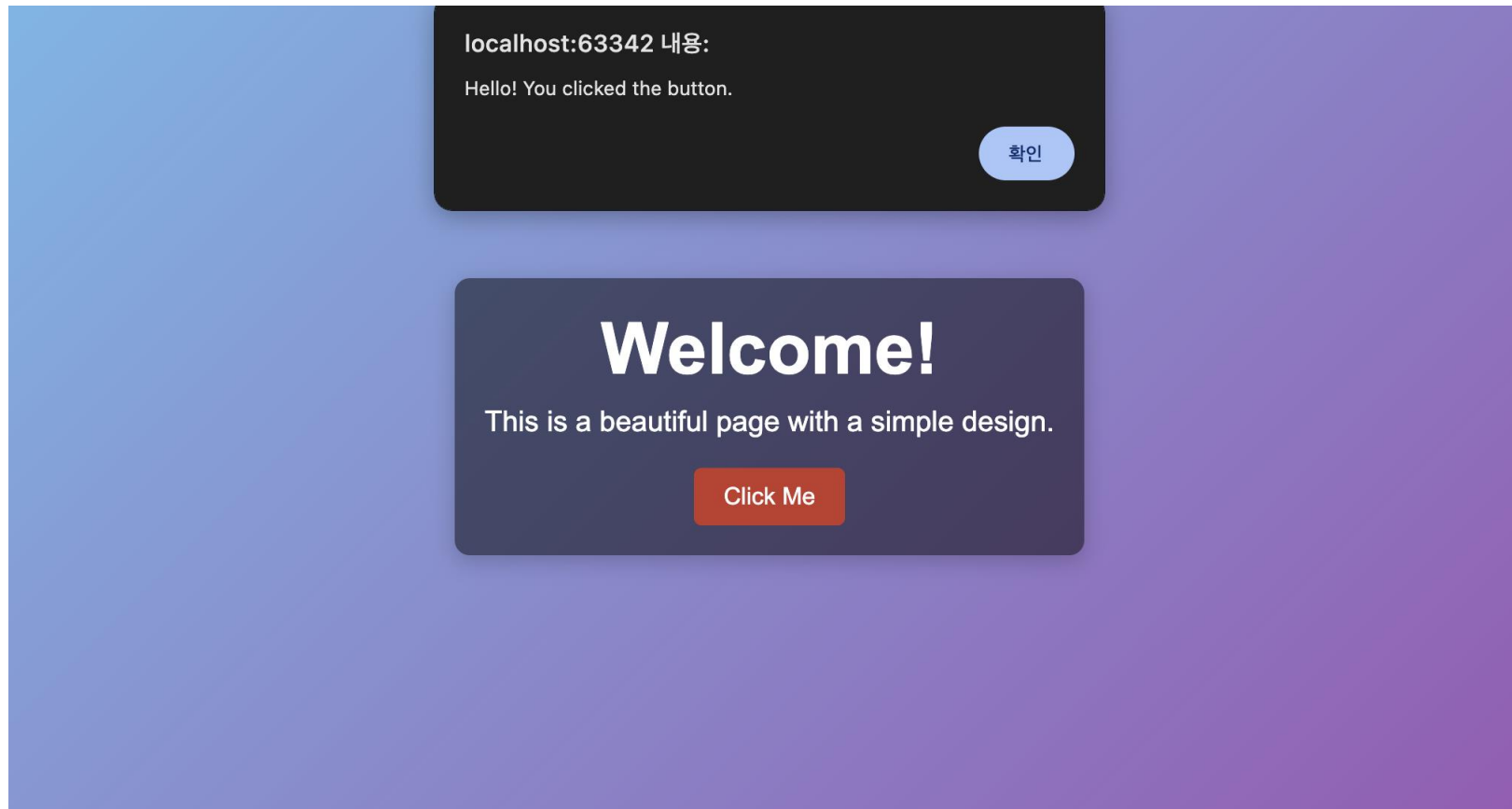
```
<div class="container">
  <h1>Welcome!</h1>
  <p>This is a beautiful page with a simple design.</p>
  <button class="button" onclick="showAlert()">Click Me</button>
</div>
<script>
  function showAlert() {
    alert("Hello! You clicked the button.");
  }
</script>
```


Javascript란?

```
<style>
  body {
    font-family: 'Arial', sans-serif;
    margin: 0;
    padding: 0;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    background: linear-gradient(135deg, #71b7e6, #9b59b6);
    color: #fff;
  }
  .container {
    text-align: center;
    padding: 20px;
    background: rgba(0, 0, 0, 0.5);
    border-radius: 10px;
    box-shadow: 0 4px 15px rgba(0, 0, 0, 0.2);
  }
  h1 {
    margin: 0;
    font-size: 3rem;
  }
  p {
    margin: 10px 0 20px;
    font-size: 1.2rem;
  }
  .button {
    padding: 10px 20px;
    font-size: 1rem;
    color: #fff;
    background: #e74c3c;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    transition: background 0.3s ease;
  }
  .button:hover {
    background: #c0392b;
  }
</style>
```

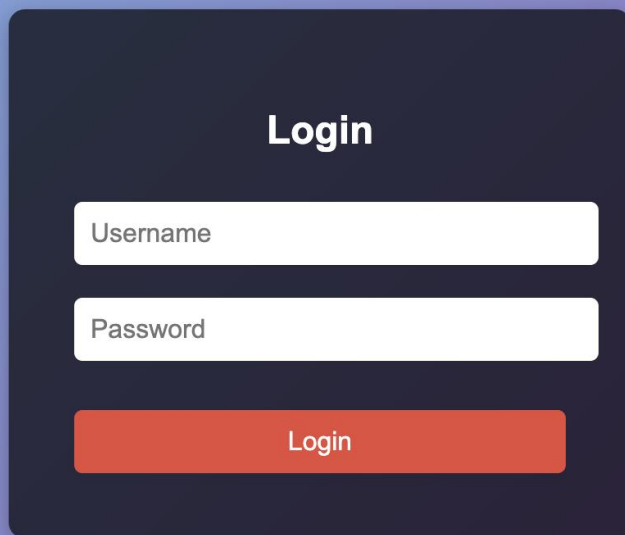
```
<div class="container">
  <h1>Welcome!</h1>
  <p>This is a beautiful page with a simple design.</p>
  <button class="button" onclick="showAlert()">Click Me</button>
</div>
<script>
  function showAlert() {
    alert("Hello! You clicked the button.");
  }
</script>
```

Javascript란?



Sample Page

JavaScript의 역할



Login

Username

Password

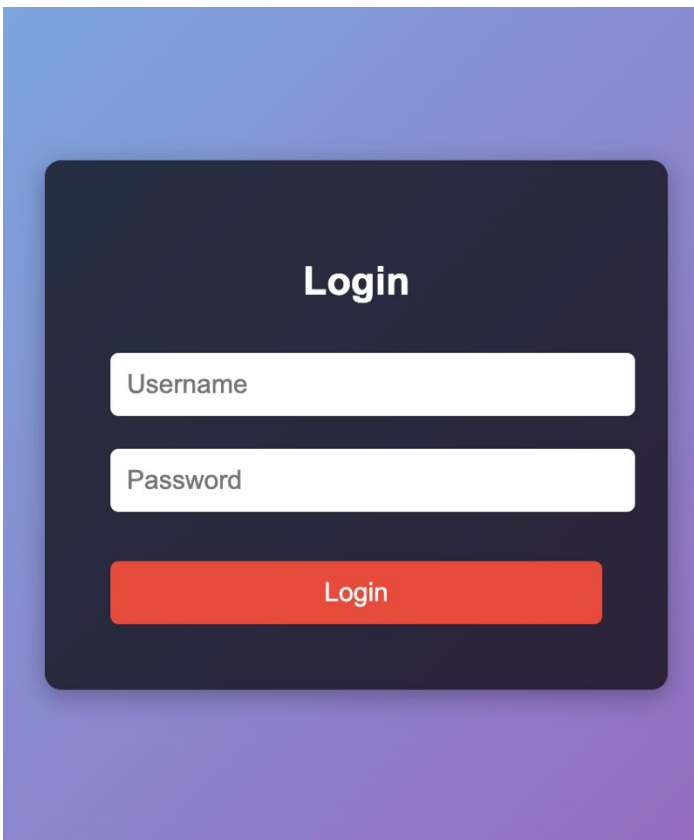
Login

1. 입력 여부 판단

2. 클릭시 이벤트 발생

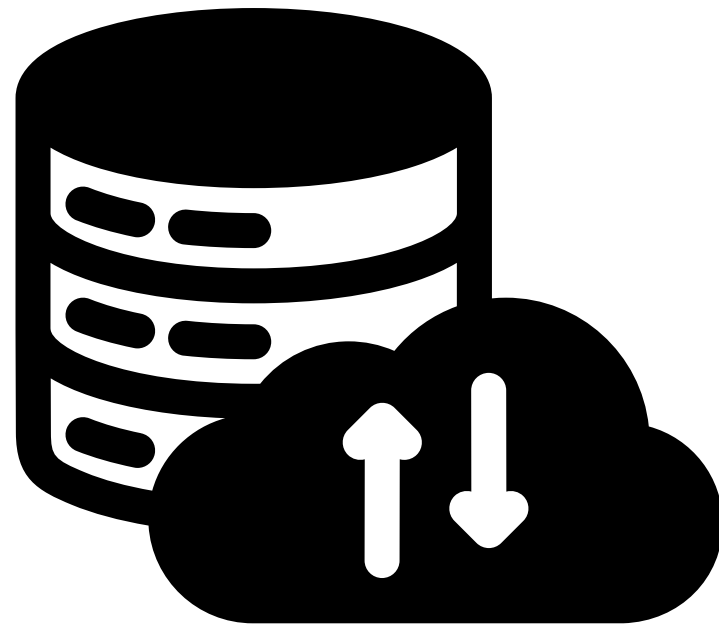
HTML과 연동하여 동작!

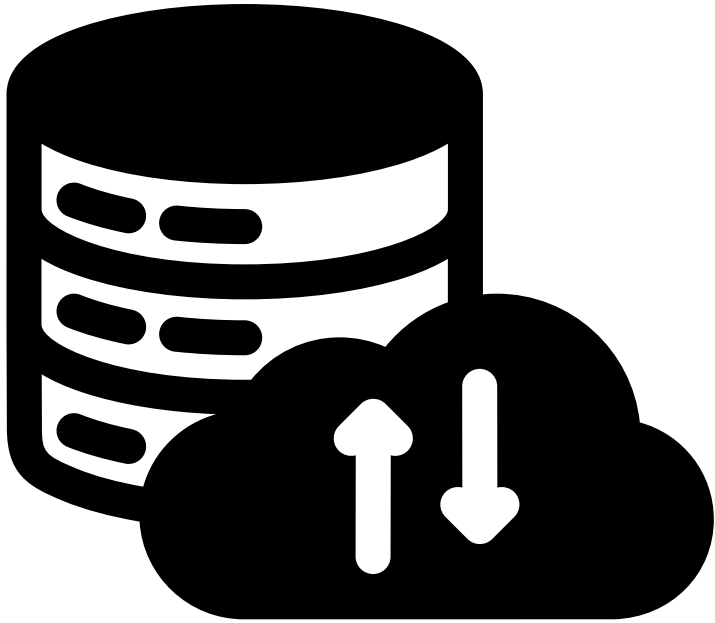
Javascript의 역할



A login form with a dark blue background. It features a title "Login" at the top. Below the title are two input fields: "Username" and "Password". At the bottom is a red button labeled "Login".

서버와 통신 담당!





서버 내부적으로 여러가지 기능 수행!

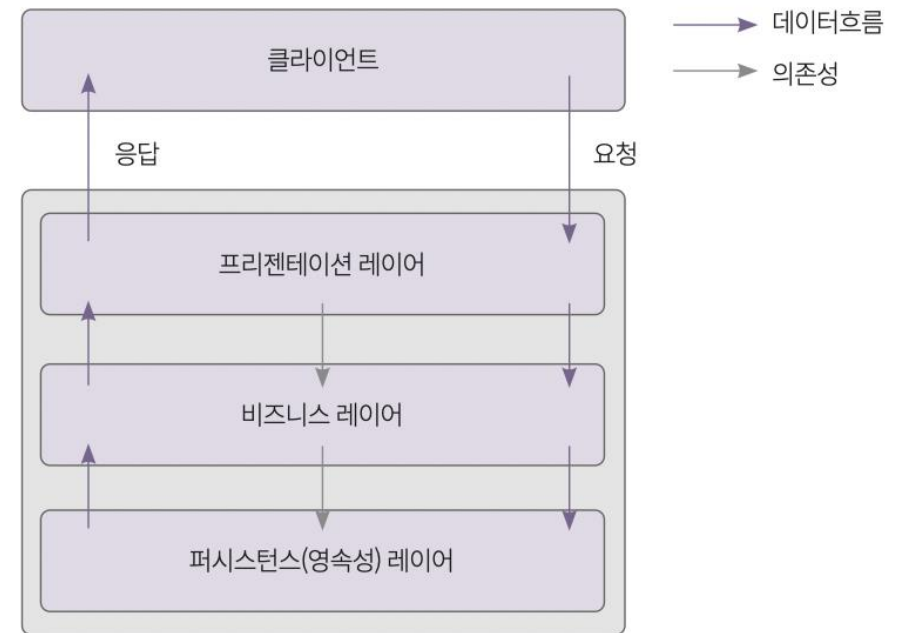
=> 추후 자세히 다룰 예정

- BackEnd Language로서의 역할!
- 로직 처리 => **Business logic**

계층형 아키텍처

- 소프트웨어를 몇 가지 계층으로 나누어 만드는 방식
- 가장 흔한 아키텍처 형태 – 단순함, 이해하기 쉬움
- 각 계층이 논리적으로 분리됨 -> 소규모 애플리케이션에서 많이 채택
- 각 계층은 주어진 역할만을 수행, 계층간 의존성은 단방향임

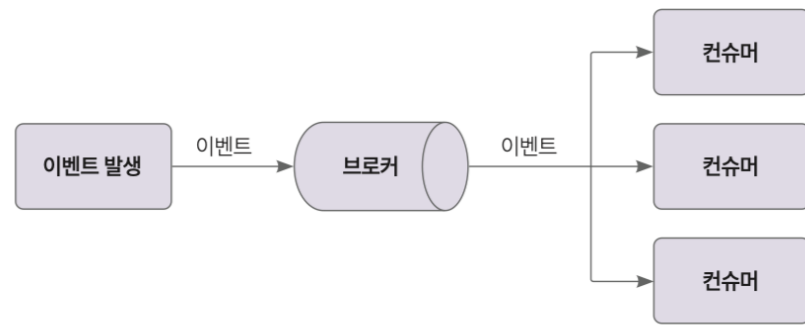
▼ 계층형 아키텍처 예시



이벤트 기반 아키텍처

- 이벤트의 상태 변화에 대응하는 소프트웨어 설계 패턴
- 이벤트 : 시스템에 영향을 주는 상황 의미 (사용자 로그인, 버튼 클릭 등)
- 구성 : 프로듀서 (이벤트 발생시킴) + 컨슈머 (전달하는 브로커 이벤트 수신)
- 모든 요청을 비동기로 처리하므로 확장성이 좋음

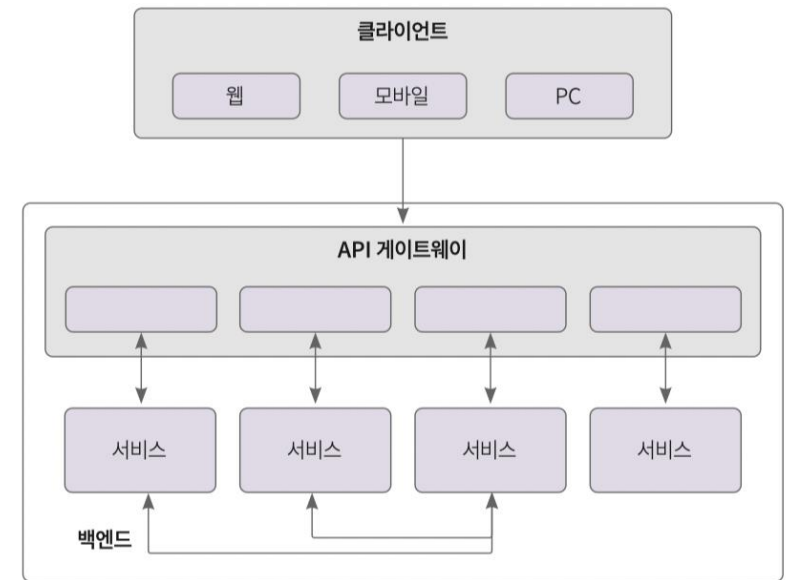
▼ 이벤트 기반 아키텍처 예시



마이크로 서비스 아키텍처

- 시스템을 여러 개의 작은 서비스로 나누어서 관리하는 설계 방식
- 각 서비스는 독립적으로 개발, 배포, 운영됨 - 구조적으로 분리됨
- 서비스 분리 - 독립적으로 업데이트 및 스케일링이 가능함
- 대규모 시스템 구축에 적합
- 각 서비스는 하나의 도메인을 책임지는 형태로 쪼개는 것이 좋음

▼ 마이크로서비스 아키텍처 예시



웹서비스와 백엔드

백엔드 프로그래밍 언어

: 자바, 파이썬, C++, C#, 자바스크립트를 많이 사용

- 파이썬 : 데이터 분석 분야에서 인기 많음
- C++: 성능이 중요한 게임 서버 개발에서 많이 사용
- 자바 : 스프링 기반의 웹서비스에서 많이 사용
- C# : 닷넷 프레임워크, 마이크로소프트 기반 인프라와 함께 많이 사용
- 자바스크립트 : FE, BE에서 모두 인기

대표적인 웹 프레임워크 : express

- Nest JS, Next JS도 급성장 중

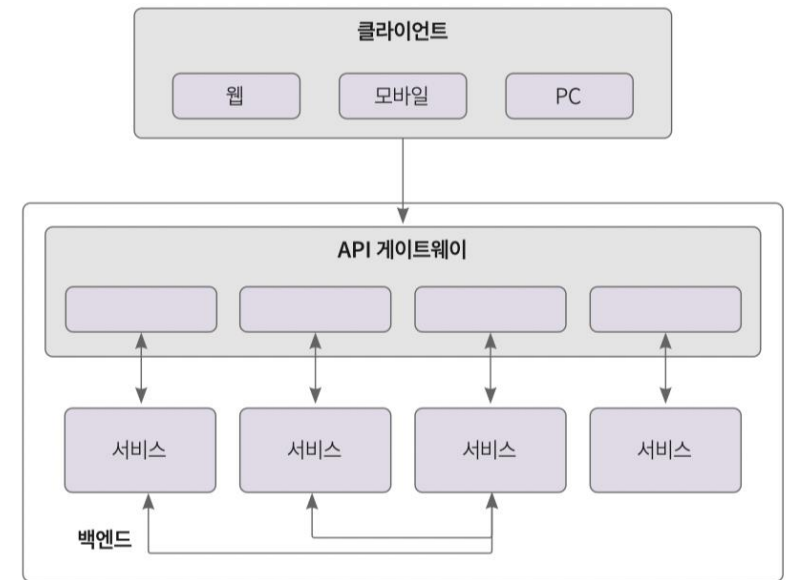
▼ 백엔드 프로그래밍 언어



데이터베이스

- 검색과 축적이 쉽도록 정리된 데이터의 모음
- 일반적으로는 데이터베이스 소프트웨어를 의미
- 크게는 RDB와 NoSQL로 나뉨

▼ 마이크로서비스 아키텍처 예시



데이터베이스 – RDB

- ACID 트랜잭션
 - Atomicity 원자성 / 일관성 Consistency / 격리성 Isolation / 내구성 Durability을 의미
- SQL
 - Structured Query Language의 약자
 - 쿼리(데이터 검색)를 하는 프로그래밍 언어

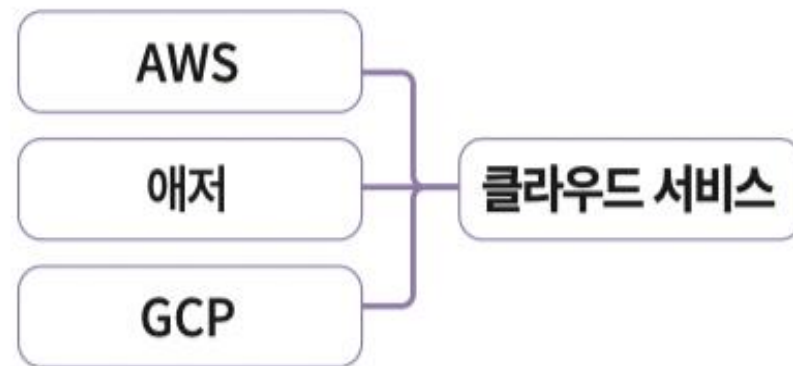
데이터베이스 - NoSQL(Not Only SQL)

- RDB의 문제들을 해결하기 위해 등장
- 클러스터 지원 -> RDB에 비해 스케일 아웃이 쉬움
- 테이블이 아닌 다른 방법으로 데이터 모델링을 함
 - Ex. 키 밸류, 컬럼, 오브젝트
- 키-밸류 스토어 : 다이나모디비, 카우치베이스가 유명
- 도큐먼트 스토어 : 몽고디비
- 와이드 컬럼 스토어 : 테이블과 유사하게 행과 열을 사용
 - 행마다 열의 타입이 다를 수 있음
 - 아파치의 카산드라가 유명

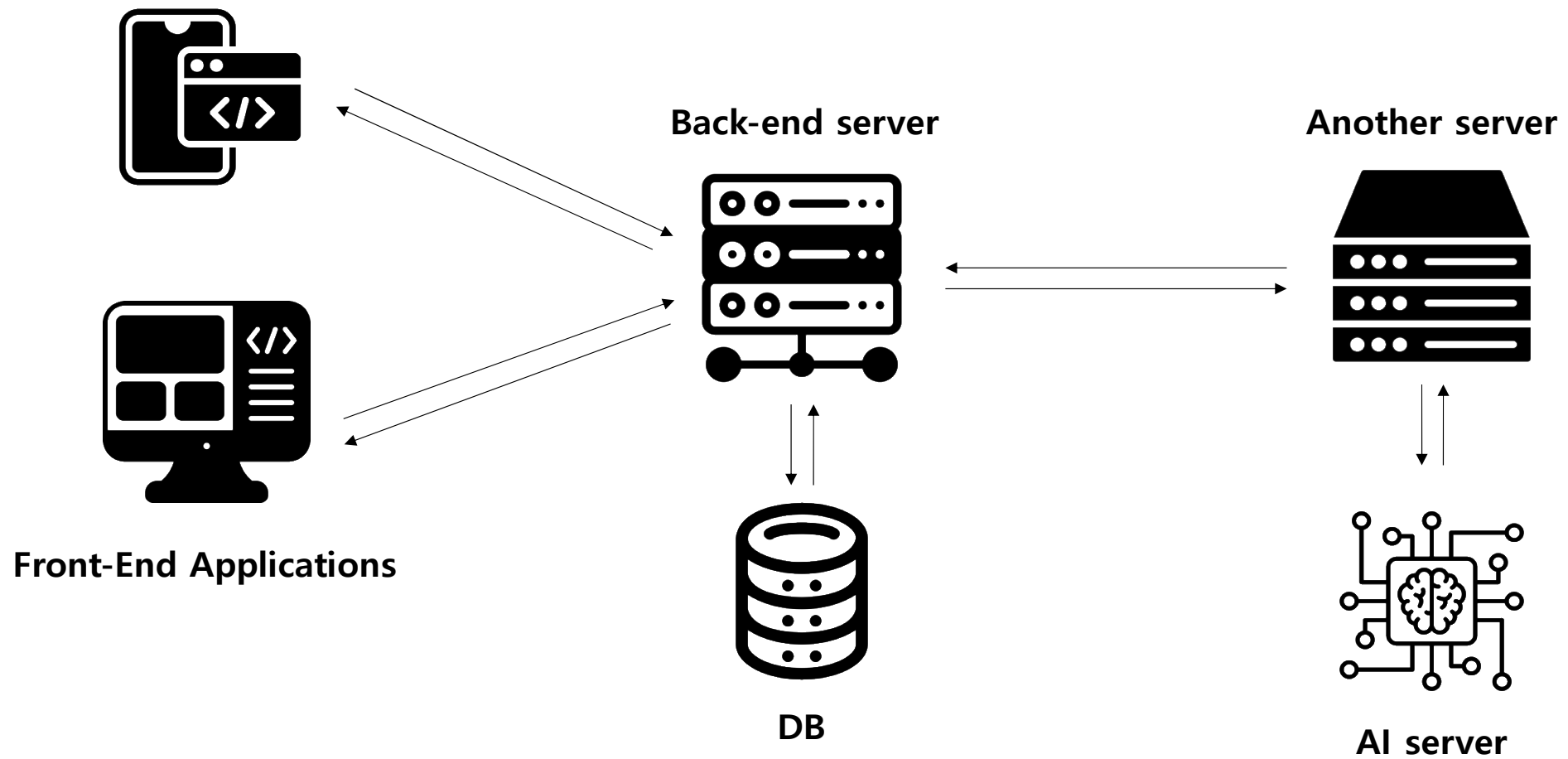
클라우드 서비스

- 이전 : 많은 기업들이 IDC에 서버를 설치한 후 서비스
- 매우 많은 서버의 CPU와 메모리, 네트워크를 나누어서 서비스
- 서버의 자원을 탄력적으로 제공 가능함
- 클라우드를 제공하는 회사 : AWS, AZURE, GCP 등
- 2022년 1분기 기준 3사가 65%의 점유율 가짐
- 사용한 만큼만 요금을 냄

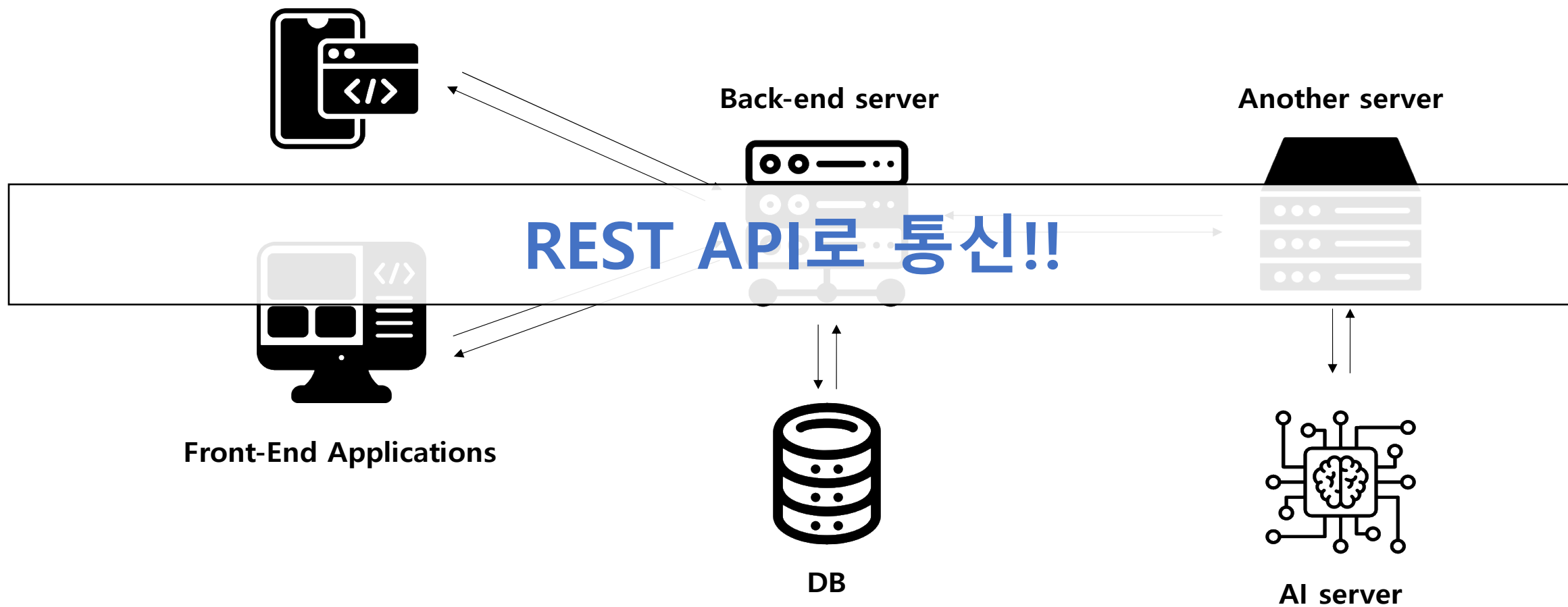
▼ 주요 클라우드 서비스



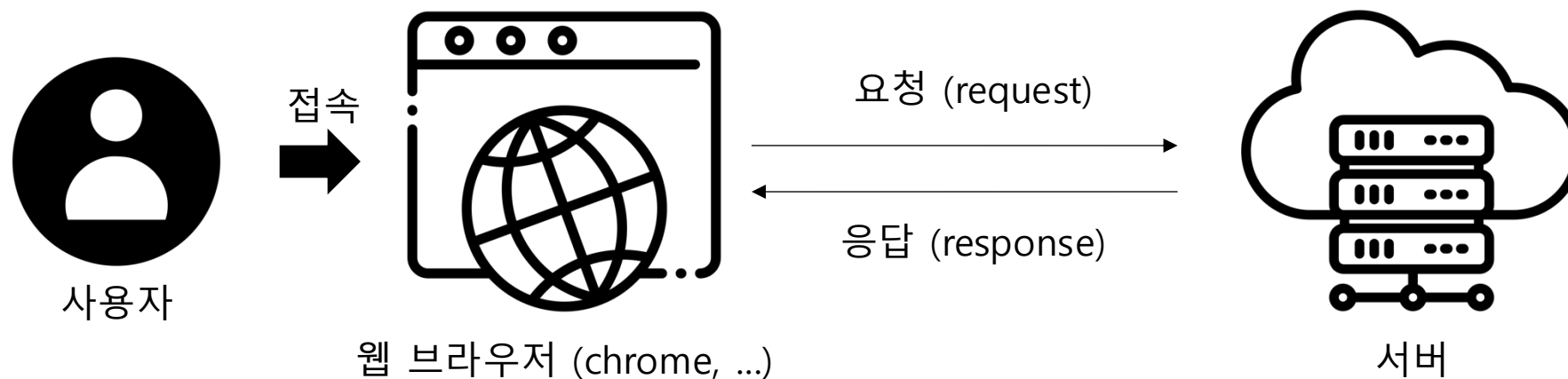
웹서비스의 구성



웹서비스의 구성



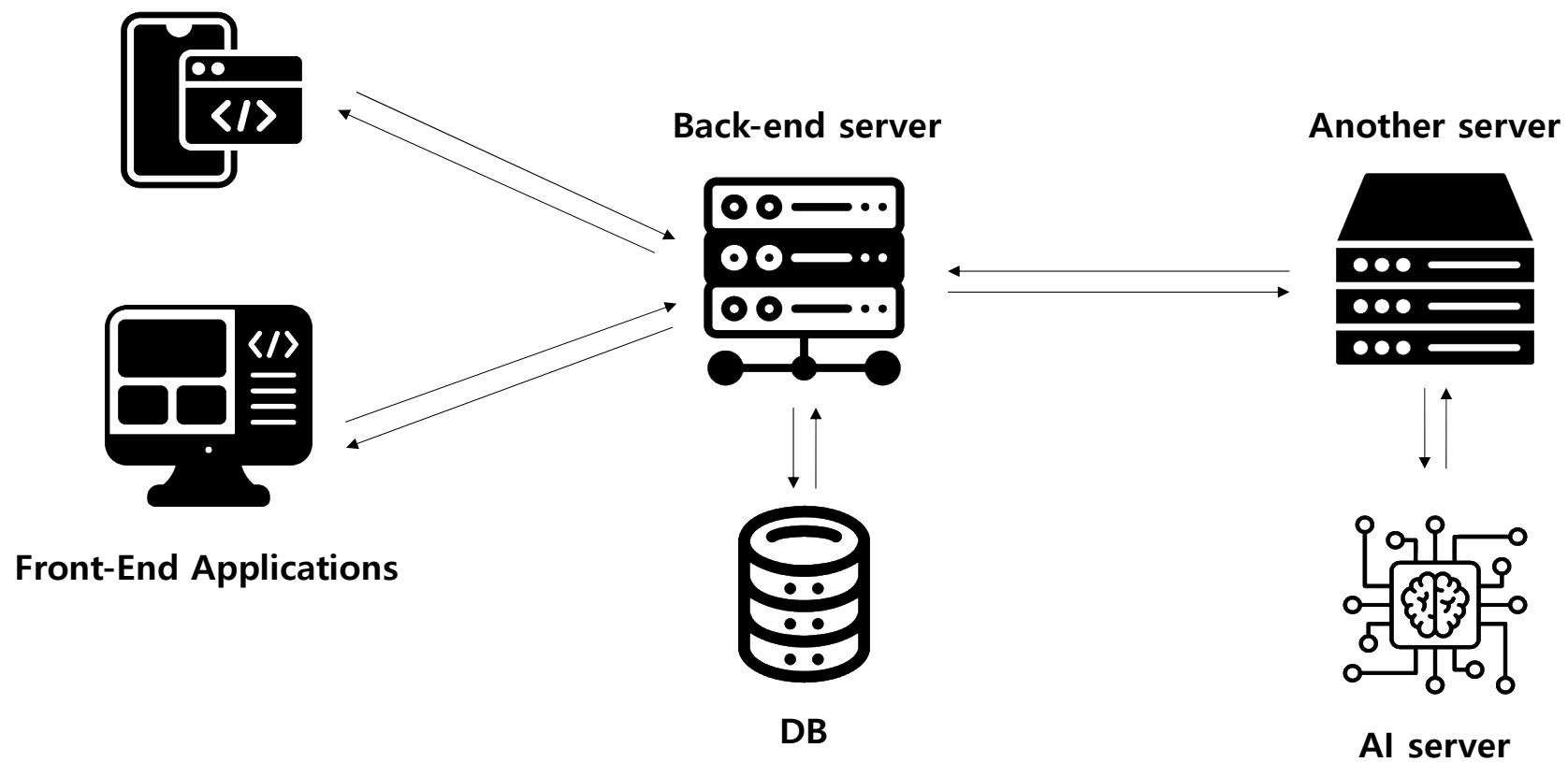
웹서비스의 구성



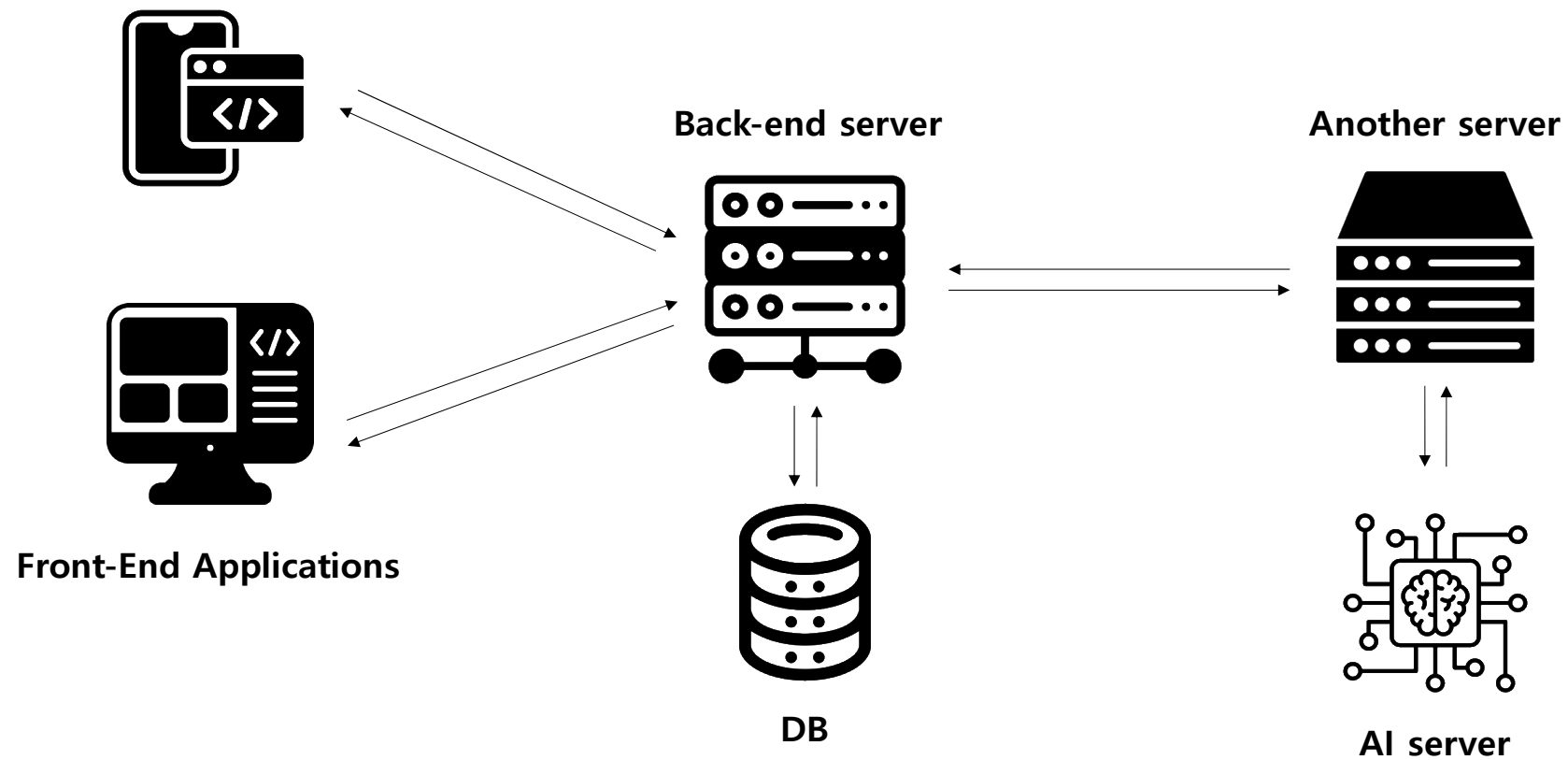
웹 API의 역할

- 서버와 데이터베이스안의 리소스에 접근
 - 데이터베이스의 정보를 누구나 열람하면 곤란하기에, 필요에 의해서만 열람 가능.
 - API는 접근 권한이 인가된 사람에게만 서버와 데이터베이스에 접근하도록 함.
- 모든 요청과 응답을 표준화
 - 아이폰을 쓰던 갤럭시 폰을 사용하든 상관없이 동일한 API를 사용하기 때문에 클라이언트의 요청과 서버의 응답을 하나의 API로 표준화 할 수 있음.

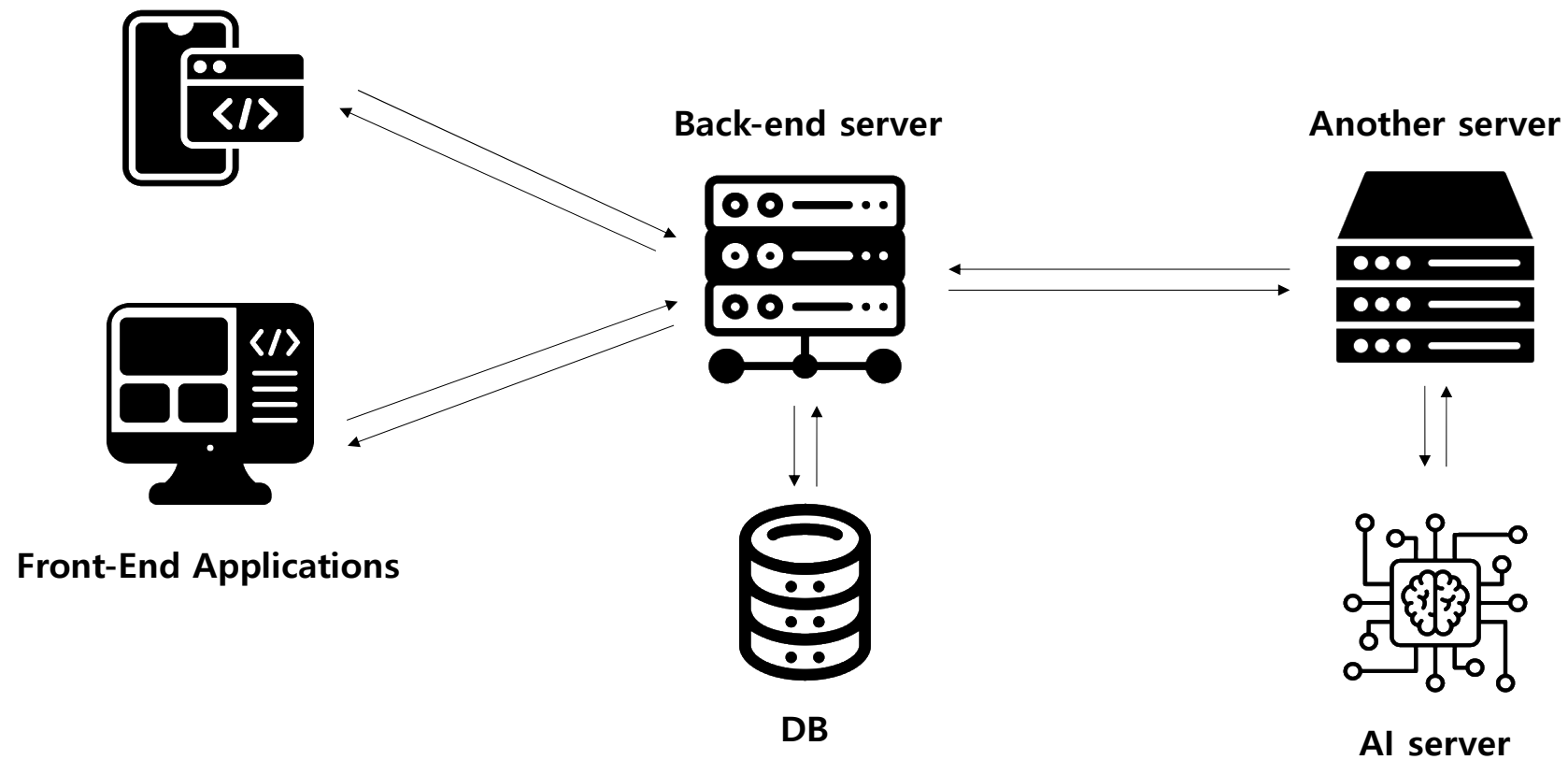
Front-End



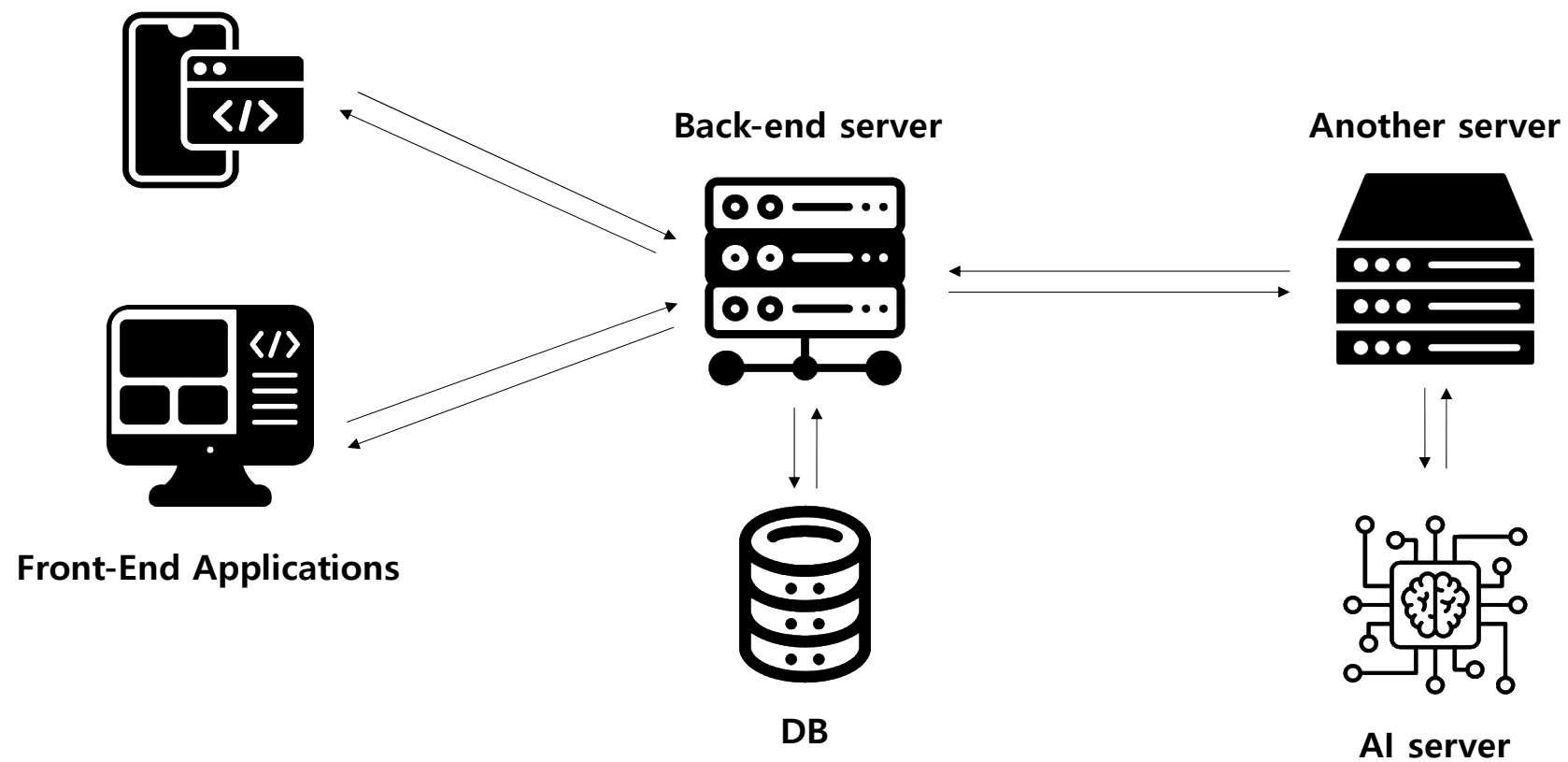
Back-End



Database

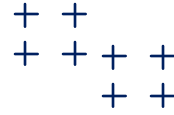


AI Service



Javascript 기본 문법

Javascript 기본 구조



감사합니다.

- 본 온라인 콘텐츠는 2024년도 과학기술 정보통신부 및 정보통신기획평가원의 'SW중심대학사업' 지원을 받아 제작되었습니다.
- 본 결과물의 내용을 전재할 수 없으며, 인용(재사용)할 때에는 반드시 과학기술정보통신부와 정보통신기획평가원이 지원한 'SW중심대학'의 결과물이라는 출처를 밝혀야 합니다.

