

# 백엔드와 데이터 분석 병행 3개월 학습 로드맵

## 목차

- [학습 목표 및 조건](#)
- [주차별 커리큘럼 \(12주\)](#)
- [1주차: JavaScript & Python 기초](#)
- [2주차: JavaScript ES6 & Python 중급](#)
- [3주차: React 도입 & Pandas 입문](#)
- [4주차: React Hooks & Pandas 심화](#)
- [5주차: Next.js 기본 & SQL 기초](#)
- [6주차: Next.js API/라우팅 & SQL 심화](#)
- [7주차: Node.js/Express & EDA 프로젝트](#)
- [8주차: Supabase 시작 & 데이터 시각화](#)
- [9주차: Supabase 보안\(RLS\) & 데이터 분석 프로젝트](#)
- [10주차: 배포 준비\(Next.js/Vercel\) & SQL-Jupyter 연동](#)
- [11주차: 통합 프로젝트 개발](#)
- [12주차: 리뷰 및 최종 정리](#)
- [일일 학습 시간 분배](#)
- [병행 학습 팁](#)
- [결과물 및 프로젝트](#)

## 학습 목표 및 조건

사용자는 **하루 4시간 이상, 주말 5시간**의 학습 시간을 확보할 수 있다. 따라서 매주 평일 4시간(예: 오전 2시간, 오후 2시간), 주말 각 5시간 학습을 가정했다. 최종 목표는 **실무 수준의 프로젝트 결과물**을 만들 수 있도록 실습 중심으로 학습하는 것이다. 로드맵은 JavaScript 기반의 웹 개발(Next.js/React/Node.js/Supabase)과 Python 기반의 데이터 분석(Pandas+SQL)을 병행 학습하도록 설계했다. 핵심 기술로는 **REST API, React Hooks(useEffect 등), Express 라우팅, SQL 외래키, Supabase RLS(행 수준 보안)** 등이 있으며, 이를 학습 일정에 맞춰 반드시 다룰 예정이다. 매주 일정 시간을 짧은 복습 및 정리에 할애해 이해도를 높인다.

## 주차별 커리큘럼 (12주)

### 1주차: JavaScript 기초 & Python 기초

- **웹 개발(프로그래밍 기초):** JavaScript 기본 문법(변수 `var/let/const`, 자료형, 연산자, 제어문, 함수) 학습. 브라우저에서의 JS 실행 환경(DOM, console) 이해. 간단한 계산기나 할 일(ToDo) 입력 예제 실습.
- **데이터 분석(파이썬 입문):** Python 기본 문법(변수, 자료형, 연산자, 제어문, 함수 정의) 학습. Python 인터프리터 및 **Jupyter Notebook** 설치·사용. 간단한 산술 및 문자열 처리 실습.
- **핵심 개념:** 변수 스코프(전역/지역), 자료구조(리스트, 딕셔너리) 이해. 개발자 도구(크롬 DevTools) 활용법.
- **실습 및 복습:** JavaScript 코드와 Python 코드를 비교하며 짧은 스크립트 작성. 주말에 학습한 내용을 정리하고 간단 문제 풀이.

### 2주차: JavaScript ES6 & Python 중급

- **웹 개발(심화 JS):** ES6+ 문법(화살표 함수, 템플릿 리터럴, 객체/배열 구조 분해, 클래스) 학습. `npm` 과 모듈 개념 이해, 간단한 Node.js 환경 설치.

- **데이터 분석(Python 자료구조):** Python 자료구조(집합, 튜플, 반복문, 조건문 심화) 학습. 파일 입출력과 예외 처리 기본. Pandas 설치 및 기본 사용법 연습.
- **핵심 개념:** 비동기 기초(setTimeout/Promise 개념), 변수 호이스팅. Python의 반복 및 제어 흐름.
- **실습 및 복습:** JavaScript ES6 기능을 사용하여 배열 조작 예제. Python으로 CSV 파일 읽기 연습. 주중 배운 개념 정리 및 짧은 퀴즈 응시.

### 3주차: React 도입 & Pandas 입문

- **웹 개발(React 소개):** React 프로젝트 셋업( `create-react-app` 또는 Next.js 초기 페이지 생성). JSX 문법과 컴포넌트 구성 이해. `props` 와 `state` 기초 개념 학습.
- **데이터 분석(Pandas 기초):** Pandas 소개 및 데이터프레임 생성. CSV/Excel 파일 불러오기( `read_csv` 등). 데이터 조회 및 기초 통계 확인.
- **핵심 개념:** React 컴포넌트 생명주기(클래스 vs 함수형), **useState** 훅. Pandas의 DataFrame, Series 개념. React Hook의 부수효과 처리 예: `useEffect` 훅 ① .
- **실습 및 복습:** React 컴포넌트 트리 예제 구현(제목, 내용 등). Pandas로 샘플 데이터 EDA(기초 통계량, head/tail). 짧은 코딩 숙제와 주말 복습.

### 4주차: React Hooks & Pandas 심화

- **웹 개발(React Hooks):** `useState` , `useEffect` 등을 활용하여 컴포넌트 상태 관리와 부수효과 처리. 간단한 API 호출 예제(가짜 JSON API fetch) 구현. React Router 또는 Next.js 기본 라우팅 개념 소개.
- **데이터 분석(Pandas 심화):** 데이터 필터링, `groupby` 집계, 결측치 처리(드롭/채우기) 실습. Matplotlib/Seaborn으로 기본적인 데이터 시각화(막대/선 그래프) 학습.
- **핵심 개념:** **useEffect**를 통한 렌더링 후 작업 처리 ① . 데이터 전처리(정제, 결합). React 컴포넌트 간 통신.
- **실습 및 복습:** Todo 리스트 같은 React 앱 구현(항목 추가/삭제 기능). 공공 데이터(예: 코로나 통계) 불러와 Pandas로 간단 분석. 중간 복습 및 노트 정리.

### 5주차: Next.js 기본 & SQL 기초

- **웹 개발(Next.js 입문):** Next.js 프로젝트 생성 및 폴더 구조 이해. Pages 기반 라우팅, 정적 생성(SSG)과 서버 사이드 렌더링(SSR) 개념 소개. 간단한 **Next.js 페이지**(홈, 소개) 제작. **Next.js**는 React 기반 **풀스택 웹 앱 프레임워크**로, 추가 기능과 최적화를 제공한다 ② .
- **데이터베이스(SQL 기초):** 관계형 데이터베이스(RDB) 개념 소개. SQL `SELECT` , `WHERE` , `ORDER BY` 등 기본 쿼리 학습. **Primary Key(기본키)** 개념, 기본적인 테이블 생성 실습.
- **핵심 개념:** Next.js의 **파일 기반 라우팅**과 페이지 프리렌더링. SQL 데이터 조회 및 간단 조인(익히기 전). Prisma나 Supabase CLI를 이용한 DB 연결 기초.
- **실습 및 복습:** Next.js로 정적 페이지 배포 연습. SQLite나 Supabase 데이터베이스에 샘플 테이블 생성하고 간단 쿼리 실행. 배운 SQL 문법 복습.

### 6주차: Next.js API/라우팅 & SQL 심화

- **웹 개발(Next.js API 라우트):** Next.js에서 `pages/api` 를 이용한 **REST API** 엔드포인트 구현. 간단한 GET/POST 엔드포인트 작성. Express.js 개념 소개(필요 시). **REST API**는 모든 요청을 상태에 독립적으로 처리하며 서버는 각 요청만으로 완전한 응답을 반환한다 ③ .
- **데이터베이스(SQL 심화):** SQL **JOIN**(INNER, LEFT)으로 여러 테이블 연결하기 실습. **Foreign Key(외래키)** 개념 학습 및 제약조건 적용 ④ . Group By, Aggregation 함수(MAX, COUNT 등) 활용.
- **핵심 개념:** 서버 간 통신 방식(RESTful 원칙) ③ . Express 라우팅: 애플리케이션의 엔드포인트가 클라이언트 요청에 응답하는 방식을 정의한다 ⑤ . SQL JOIN으로 데이터 결합 ⑥ , 관계 무결성 유지.
- **실습 및 복습:** Next.js API를 통해 예제 데이터를 반환하는 API 개발. Express로 간단한 라우팅 서버 생성 실습. 데이터베이스에서 두 테이블 JOIN하여 결과 조회. 주간 학습 내용 정리.

## 7주차: Node.js/Express & EDA 프로젝트

- **웹 개발(Node.js/Express):** Node.js 환경 설정(Node Version Manager 이용). Express.js로 웹 서버 구성, RESTful API 엔드포인트 작성(예: /users, /posts). **Express**는 Node.js를 위한 가볍고 유연한 웹 애플리케이션 프레임워크로, 견고한 API를 빠르게 구축할 수 있게 돕는다 7. 라우팅과 미들웨어 개념 숙지.
- **데이터 분석(EDA 프로젝트):** 데이터 분석을 위한 작은 프로젝트 진행(예: 영화/음악/학생 데이터 등). Pandas를 활용해 데이터셋을 불러오고 EDA(기초 통계, 시각화) 수행. 보고서 형태로 주요 발견 사항 기록.
- **핵심 개념:** Express 라우팅 설정 및 미들웨어 사용 5 7. 데이터 탐색(EDA) 방법론: 이상치 탐색, 변수 간 관계 파악. Git을 이용한 버전 관리 시작.
- **실습 및 복습:** 간단한 CRUD(생성/읽기/수정/삭제) API 개발 및 테스트. Pandas로 대용량 CSV 처리 실습. GitHub에 코드 푸시 및 주말까지 복습.

## 8주차: Supabase 시작 & 데이터 시각화

- **웹 개발(Supabase 입문):** Supabase 가입 및 프로젝트 생성. PostgreSQL 테이블 생성, 인증(Auth) 설정 실습. Supabase API(Key)를 이용해 Next.js/Node.js에서 DB에 데이터 읽기/쓰기.
- **데이터 분석(시각화 심화):** Matplotlib과 Seaborn을 이용한 심화 시각화(히스토그램, 박스플롯, 산점도 등). 데이터 전처리 실전: 결측치 대체, 파생 변수 생성.
- **핵심 개념:** Supabase 사용법: 데이터베이스 연결과 인증 관리. 데이터 시각화 라이브러리 활용.
- **실습 및 복습:** Supabase로 사용자 테이블 만들고 간단한 쿼리 수행. Next.js에서 Supabase Client로 데이터 조회 예제. 분석 프로젝트에 고급 차트 추가. 배운 내용 문서화.

## 9주차: Supabase 보안(RLS) & 데이터 분석 프로젝트

- **웹 개발(Supabase RLS 및 응용):** Row Level Security(RLS) 개념과 정책 생성 실습. Supabase Auth와 RLS를 결합하여 사용자별 데이터 접근 제어. **Supabase 문서**에 따르면, 세밀한 권한 규칙이 필요할 때는 Postgres RLS만한 것이 없고 8, RLS는 데이터 보화를 위한 깊은 방어 역할을 한다 9.
- **데이터 분석(종합 프로젝트):** 실제 문제 해결 프로젝트(예: 판매 데이터 분석, 유저 행동 분석). Pandas와 SQL(또는 SQLAlchemy)을 병행하여 데이터 정제, 모델링, 인사이트 도출. 간단한 예측 모델(회귀 분석) 가능 시도.
- **핵심 개념:** RLS 정책 설정 방법과 Supabase 연결. 모델링 설계 및 결과 시각화.
- **실습 및 복습:** Supabase에서 테이블에 RLS 활성화 후 정책 적용. 데이터 분석 결과를 Markdown/Jupyter 리포트로 작성. 주간 리뷰 및 코드 정리.

## 10주차: 배포 준비(Next.js/Vercel) & SQL-Jupyter 연동

- **웹 개발(배포 준비):** Next.js 애플리케이션을 **Vercel**에 배포. Vercel은 Next.js 제작사가 개발한 서버리스 플랫폼으로, **Next.js 앱을 프로덕션에 배포하는 가장 쉬운 방법**을 제공한다 10. GitHub와 연동하여 자동 배포 설정. Node/Express 서버는 Heroku나 Supabase Edge Functions 활용.
- **데이터 분석(SQL 연동):** Jupyter Notebook에서 SQL 데이터베이스 연동 실습(pandas `read_sql`, SQLAlchemy). 분석한 결과를 대시보드 형태로 표현(예: Streamlit 기초).
- **핵심 개념:** 배포 파이프라인: Git→Vercel→공개 URL 확보. 데이터 파이프라인: Python과 SQL 연동.
- **실습 및 복습:** Next.js 앱 배포 후 실제 접속 테스트. Python에서 Supabase DB에 쿼리하여 데이터 시각화. Docker 환경 검토(선택).

## 11주차: 통합 프로젝트 개발

- **웹 개발 & 데이터 분석:** 그동안 배운 내용을 종합하여 **포트폴리오 수준의 프로젝트** 개발. 예를 들어, Next.js/React 프론트엔드와 Express/Supabase 백엔드를 조합한 웹 애플리케이션에 데이터 분석 기능을 추가. (예: 데이터 조회 API + Pandas 처리 후 결과 표시)
- **핵심 개념:** 전체 시스템 아키텍처 설계. 모듈화된 코드 작성, Git으로 협업 워크플로우 체득.
- **실습:** 개인 프로젝트 구현 및 중간 점검. 동료나 멘토 리뷰받기.

## 12주차: 리뷰 및 최종 정리

- **전체 복습:** 12주간 학습한 기술과 프로젝트를 돌아보고 부족한 부분 보완. 주요 코드를 문서화. 최종 프로젝트 발표 자료(포트폴리오 페이지) 작성.
- **핵심 개념 점검:** REST API와 HTTP, React 주요 Hook(useState/useEffect), Express 라우팅, SQL JOIN/제약, Supabase RLS 및 배포 워크플로우 등. 필요 시 추가 학습 계획 수립.
- **실습:** 프로젝트 마감, GitHub 리포지토리 정리 및 배포 검증. 후속 학습 방향 결정(예: 더 복잡한 프로젝트, 팀 협업, 추가 프레임워크 학습 등).

## 일일 학습 시간 분배

- **평일(월~금):** 하루 4시간. 예) 오전 2시간, 오후 2시간으로 분리.
- **방법1:** 월/수/금은 웹 개발(프론트/백엔드) 집중, 화/목은 데이터 분석 집중. 각 세션 종료 후 30분 복습.
- **방법2:** 매일 오전에는 웹 개발, 오후에는 데이터 분석으로 고정하여 공부.
- **주말(토~일):** 각 5시간. 예를 들어, 토요일 5시간을 **웹 프로젝트/심화 실습**에 투입하고, 일요일 5시간을 **데이터 분석 프로젝트**에 투입. 또는 한쪽 주제를 심화하고 다른 날은 복습 및 프로젝트 마무리.
- **주간 복습 시간:** 매주 금요일 또는 일요일 마지막 1시간은 그 주 배운 내용을 정리·요약. 팀 동료나 온라인 커뮤니티에 질문하며 취약점 보완.

## 병행 학습 팁

- **교차 학습:** 백엔드와 분석을 번갈아 가며 학습한다. 예) 요일별로 주제를 분리하거나, 매일 오전/오후를 나눠 특정 분야 집중. 이렇게 하면 각각의 피로를 줄이고 집중도를 높일 수 있다.
- **실습 중심:** 이론보다 코딩으로 학습하자. 예를 들어, 백엔드 실습 시 데이터를 직접 불러와 처리하고, 분석 시에는 실제 웹 API에서 데이터 받아 분석하는 식으로 연계 학습.
- **핵심 개념 정리:** 공부할 때마다 핵심 개념과 용어를 정리한다. (예: REST API의 무상태성 ③, React useEffect의 역할 ①, Express 라우팅 정의 ⑤, 외래키와 참조 무결성 ④, RLS의 중요성 ⑧ ⑨ 등)
- **교류와 협업:** 질문/토론이 가능한 스터디나 동료를 구해 정기적으로 지식을 공유한다. 코드 리뷰로 부족한 부분을 채우는 것도 도움이 된다.
- **주간 목표 설정:** 매주 시작 시 달성 목표(예: “Next.js로 페이지 3개 완성” 또는 “매출 데이터 분석 보고서 작성”)를 정하고, 주말에 결과물로 확인한다.

## 결과물 및 프로젝트

최종적으로 **실무와 유사한 웹 애플리케이션과 데이터 분석 프로젝트**를 완성한다. 예를 들어:

- **웹 애플리케이션 예시:** Next.js/React 프론트엔드와 Express/Supabase 백엔드를 이용해 간단한 게시판이나 포트폴리오 사이트 구축. 사용자 인증(Supabase Auth)과 데이터 CRUD, RESTful API 연동 등을 구현. (배포는 Vercel)
- **데이터 분석 프로젝트 예시:** 실제 공개 데이터를 이용한 분석 리포트(예: 공공 데이터 EDA, 머신러닝 예측 모델 구현) 작성. 분석 결과를 시각화하여 정리하고, 웹 앱과 연계하여 차트로 보여주어도 좋다.

모든 학습 과정은 **실습 기반**이다. 각 주차마다 미니 프로젝트나 과제를 수행하고, 매주 짧게 코드와 결과를 정리한다. 이렇게 하면 3개월 후에는 웹서비스 개발부터 데이터 분석까지 통합된 프로젝트 수준의 산출물을 얻을 수 있다.

**참고:** React `useEffect`는 컴포넌트 렌더링 이후 특정 작업을 수행하도록 해주는 훅이다 ①. Express 라우팅은 “애플리케이션의 엔드포인트(URI)가 클라이언트 요청에 응답하는 방식”이다 ⑤. REST API는 각 요청이 독립적으로 처리되는 무상태 아키텍처를 따른다 ③. Pandas는 Python 기반의 고성능 데이터 분석 라이브러리이다 ⑪. Supabase의 RLS는 Postgres 고유의 기능으로, “세밀한 권한 규칙”에 탁월한 보안책으로 추천된다 ⑧ ⑨.

1 리액트(react.js) useEffect란 무엇인가? - RARRIT NOTE

<https://rarrit.github.io/react/til/react-useeffect/>

2 Introduction | Next.js

<https://nextjs.org/docs>

3 What is RESTful API? - RESTful API Explained - AWS

<https://aws.amazon.com/what-is/restful-api/>

4 SQL FOREIGN KEY Constraint | GeeksforGeeks

<https://www.geeksforgeeks.org/foreign-key-constraint-in-sql/>

5 Express routing

<https://expressjs.com/en/guide/routing.html>

6 SQL Joins

[https://www.w3schools.com/sql/sql\\_join.asp](https://www.w3schools.com/sql/sql_join.asp)

7 Express - Node.js web application framework

<https://expressjs.com/>

8 9 Row Level Security | Supabase Docs

<https://supabase.com/docs/guides/database/postgres/row-level-security>

10 Pages Router: Deploy to Vercel | Next.js

<https://nextjs.org/learn/pages-router/deploying-nextjs-app-deploy>

11 pandas - Python Data Analysis Library

<https://pandas.pydata.org/>