**📅 12주 React 마스터 학습 플랜**

**Part 0. React를 위한 필수 JavaScript**

* **Chapter 0.1**: const vs let (변수와 스코프)
* **Chapter 0.2**: 화살표 함수 () => {}
* **Chapter 0.3**: 구조 분해 할당
* **Chapter 0.4**: 배열 고차 함수 (map, filter)
* **Chapter 0.5**: 비동기 통신 (Promise, async/await)

**Week 1 (Part 0)**

* **목표**: React 코드의 90%를 차지하는 JS 문법을 자유롭게 쓸 수 있도록 기본기 다지기
* **실습 과제**:
  + 배열 데이터 필터링/매핑 예제 만들기
  + API 호출 후 결과 출력 (fetch)
* **미니 프로젝트**: JSONPlaceholder API에서 게시글 목록 불러와 콘솔에 출력

**Part 1. React 시작하기 - 기본기 다지기**

* **Chapter 1**: 왜 React인가? (철학과 핵심 원칙)
* **Chapter 2**: JSX (JavaScript와 UI의 만남)
* **Chapter 3**: 컴포넌트의 두 가지 데이터: Props와 State
* **Chapter 4**: 컴포넌트 설계와 재사용성

**Week 2 (Part 1 - Ch 1~2)**

* **목표**: React의 철학과 JSX 문법 이해
* **실습 과제**:
  + Hello, React! 출력 컴포넌트 만들기
  + JSX에서 조건부 렌더링/리스트 렌더링 간단 적용

**Week 3 (Part 1 - Ch 3~4)**

* **목표**: 데이터 흐름과 컴포넌트 재사용성 이해
* **실습 과제**:
  + 버튼 클릭 시 카운터 증가 컴포넌트
  + 재사용 가능한 Button 컴포넌트 설계

**Part 2. 인터랙티브 웹 만들기 - 핵심 기술 마스터**

* **Chapter 5**: 사용자와의 소통: 이벤트 처리
* **Chapter 6**: 데이터 목록 다루기: 리스트와 Key
* **Chapter 7**: 똑똑한 렌더링: 조건부 렌더링

**Week 4 (Part 2 - Ch 5~7)**

* **목표**: 인터랙티브 UI 구현 기초
* **실습 과제**:
  + 입력폼에서 입력값 실시간 반영
  + 조건에 따라 다른 UI 표시
* **주간 프로젝트 (Project 1 시작)**: To-Do 리스트 앱 (추가/삭제/완료 표시)

**Week 5 (Project 1 완성)**

* **목표**: Part 0~2 종합 적용
* **기능 요구사항**:
  + 할 일 추가/삭제
  + 완료 상태 토글
  + 남은 할 일 수 표시
  + CSS Modules 스타일 적용
* **보너스**: 로컬스토리지 저장

**Part 3. 더 나은 React 코드 작성하기 - Hooks 심화**

* **Chapter 8**: 컴포넌트의 생명주기: useEffect
* **Chapter 9**: 거대한 State 관리: useReducer
* **Chapter 10**: 렌더링 최적화: useMemo와 useCallback
* **Chapter 11**: DOM 직접 제어와 값의 유지: useRef
* **Chapter 12**: Props Drilling 회피: useContext

**Week 6 (Part 3 - Ch 8~9)**

* **목표**: 컴포넌트 생명주기와 복잡한 상태 관리 이해
* **실습 과제**:
  + API 데이터 가져오기 & 로딩/에러 처리
  + useReducer 카운터 구현

**Week 7 (Part 3 - Ch 10~12)**

* **목표**: 렌더링 최적화 & 전역 데이터 공유
* **실습 과제**:
  + 배열 연산 성능 최적화
  + 입력창 자동 포커스
  + 다크모드 Context 구현

**Part 4. 실전 애플리케이션 개발**

* **Chapter 13**: 페이지 이동의 구현: React Router
* **Chapter 14**: 외부 세계와의 통신: fetch를 이용한 데이터 페칭
* **Chapter 15**: 컴포넌트 스타일링 (CSS Modules, Styled-Components)

**Week 8 (Part 4 - Ch 13~14)**

* **목표**: 페이지 이동과 API 연동 기초
* **실습 과제**: 다중 페이지 블로그 틀 (홈, 게시글, 소개)

**Week 9 (Part 4 - Ch 15)**

* **목표**: 다양한 스타일링 방법 학습
* **실습 과제**:
  + CSS Modules / Styled-Components 적용
* **프로젝트 2 시작**: 미니 블로그 or 상품 목록 (목록, 상세, 검색)

**Part 5. 실무에서 살아남기 - 모던 상태 관리**

* **Chapter 16**: 서버 상태 관리의 표준: React Query 또는 SWR
* **Chapter 17**: 간편하고 강력한 전역 상태 관리: Zustand 또는 Redux Toolkit

**Week 10 (Part 5 - Ch 16~17)**

* **목표**: 실무 수준의 상태 관리 도구 사용
* **실습 과제**:
  + React Query로 API 데이터 캐싱
  + Zustand로 전역 장바구니 상태 구현

**Part 6. Full-Stack으로의 확장**

* **Chapter 18**: [백엔드] 간단한 REST API 서버 만들기 (Node.js & Express)

**Week 11 (Part 6 - Ch 18)**

* **목표**: 백엔드-프론트엔드 통합 경험
* **실습 과제**:
  + CRUD API 서버 완성
  + React에서 API 연동

**Week 12 (Final Project. Full-Stack CRUD 블로그)**

* **목표**: 배운 내용을 종합한 완전한 서비스 구축
* **구현 내용**:
  + 글쓰기, 읽기, 수정, 삭제
  + React Query + Zustand 적용
  + Styled-Components 스타일링
  + Vercel/Netlify 또는 Docker로 배포
* **보너스**: GitHub에 포트폴리오 업로드

**React 프로젝트 생성 가이드 (NVM + 최신 LTS)**

**0단계: NVM for Windows 설치 (최초 1회만)**

이 단계는 컴퓨터에 NVM을 처음 설치할 때 한 번만 수행한다.

1. **기존 Node.js 제거** **제어판 > 프로그램 추가/제거**에서 컴퓨터에 설치된 모든 Node.js를 삭제한다.
2. **NVM for Windows 설치** [nvm-windows 릴리즈 페이지](https://github.com/coreybutler/nvm-windows/releases)에서 nvm-setup.zip을 받아 설치한다.
3. **최신 LTS 버전 Node.js 설치** 설치 후, **관리자 권한으로 PowerShell을 열고** 아래 명령어를 실행하여 가장 안정적인 최신 LTS 버전을 설치한다.

Bash

nvm install lts

* + **(선택 사항)** nvm list available 명령어로 설치 가능한 전체 버전 목록을 미리 확인할 수 있다.

**1단계: React 프로젝트 생성 (프로젝트마다 수행)**

1. **프로젝트 생성 위치로 이동** 터미널을 열고 프로젝트를 만들 상위 폴더로 이동한다.

Bash

cd C:\sw\ODOC

1. **LTS 버전 Node.js 사용** nvm이 자동으로 최신 LTS 버전을 찾아 사용하도록 설정한다.

Bash

nvm use lts

(nvm ls: 설치된 node 확인)

* + node -v 명령어로 버전이 올바르게 표시되는지 확인한다.

1. **Vite로 React 프로젝트 생성** react-master라는 이름의 프로젝트를 생성한다.

Bash

npm create vite@latest react-master -- --template react

1. **프로젝트 폴더로 이동** 생성된 프로젝트 폴더 안으로 들어간다.

Bash

cd react-master

1. **필요 라이브러리 설치** 프로젝트에 필요한 라이브러리들을 설치한다.

Bash

npm install

1. **개발 서버 실행** 아래 명령어로 개발 서버를 시작한다.

Bash

npm run dev

1. **실행 확인** 웹 브라우저에서 터미널에 표시된 localhost 주소(예: http://localhost:5173)로 접속하여 초기 화면이 나오는지 최종 확인한다.