



바이트디그리
React Programming

Syllabus
Fast Campus
2020. 10

00. Byte Degree Description

바이트 디그리란, 직장인 실무 교육을 운영하는 패스트캠퍼스가 산업 전문가들과 협력을 통해 구성한 기술 인증 교육 코스입니다. 디지털 기술의 빠른 변화 속에서 현업 전문가가 설계한 프로젝트를 기반으로 학습하고, 단순히 귀와 눈으로만 익히는 것에서 벗어나 수강생 손으로 직접 구현해보며 해당 기술을 익히고, 프로젝트를 통해 이해한 정도를 검증할 수 있도록 합니다. 수강생이 학습한 내용을 바탕으로 제출하는 프로젝트는 리뷰어에게 1:1 로 피드백을 받으며 일정 수준을 통과해야 바이트 디그리를 받을 수 있습니다. 전통적인 대학 학위보다 적은 시간을 들여 실무에 가장 가까운 기술을 익히고, 프로젝트 결과물을 통해 이해도를 검증하고, 인증받을 수 있습니다.

01. Course Description

바이트디그리 : React Essentials는 [올인원 패키지 : 프론트엔드(React)] 강의 중 React 기술을 익힐 때 필수적으로 학습해야 하는 부분을 재구성하였습니다.

React Essentials 바이트디그리의 경우 HTML/CSS 의 기초 부분을 이미 알고 있다는 가정 하에 시작합니다. JavaScript 기본 문법부터 중고급에 해당하는 비동기처리까지 학습한 후 본격적으로 React를 학습합니다.

1~2주 동안 JavaScript 문법에 대해 학습합니다.

3~5주에는 React 기초 문법에 대해 학습합니다. 해당 파트에서는 React 컴포넌트와 상태관리, 스타일링에 대해 학습하며 Todo 애플리케이션 예제를 만들어봅니다.

5~7주에는 서버연동, 라우터, 리덕스, 리덕스미들웨어 등 React 프로그래밍의 중급 기술들을 학습합니다. 조금 더 규모가 큰 프로젝트를 진행할때 활용할 수 있는 기술들에 대해 배울 수 있습니다.

8~10주에는 TypeScript에 대해 학습하며, TypeScript와 React를 함께 사용하는 방법에 대해 학습합니다. 약타입 언어인 JavaScript와 비교하여 어떤 장점을 가지고 있는지 알아보고, React와 함께 활용할때 어떤 장점을 가지고 있는지 실습을 통해 익혀봅니다.

02. Course Information

1. 모든 과정(강의 수강, 질의 응답, 프로젝트 피드백)은 온라인으로 진행됩니다.
2. 퀴즈 점수(출석 점수), 파트별 미니 프로젝트(3개), 최종 프로젝트(1개) 점수를 종합하여 평가 합니다.
3. 조교를 통한 질의응답과 프로젝트 결과물에 대한 리뷰어의 꼼꼼한 피드백을 진행합니다.

03. Course Objectives

1. JavaScript의 기초를 이해하고 비동기처리에 대한 개념까지 학습합니다.
2. React에 대해서 깊이 이해하고, 다양한 React 기술들을 자유롭게 사용할 수 있도록 하며, TypeScript와 React를 함께 활용하는 방법에 대해 배웁니다.
3. 12주 동안 바이트디그리 : React Essentials를 통해 React 기술을 익히고, 자신이 원하는 웹 애플리케이션을 제작 할 수 있습니다.

04. Course Task

1. 퀴즈 : 매주(15주) 한번씩 퀴즈를 통해 학습 한 내용을 점검합니다. 퀴즈는 객관식, 단답식, O/X 등 다양한 형태로 출제되며 학습한 내용을 복습하는 정도의 수준이 출제됩니다. 10점 만점으로 출제가 되며 5점 이상 득했을 시, 한 주 출석이 인정됩니다. 총 수업 주차(10주) 중 2/3(7주) 출석 인증이 되어야 바이트 디그리 수료 조건을 충족합니다.
2. Mini Project : 총 4개 파트 중 처음 3개의 파트가 끝날 때마다 각각의 파트에 해당하는 Mini Project가 출제됩니다. 해당 파트에 대한 이해도를 점검하는 수준의 프로젝트가 출제됩니다. 총 출제 되는 Mini Project(3개) 중 2개 이상 제출하여야 바이트 디그리 수료 조건을 충족합니다.
3. Final Project : 총 4개 파트 중 마지막 파트 수업이 끝날 때 출제되는 프로젝트입니다. Mini Project와 동일한 형태로 문제가 출제 되지만, 모든 파트에 대한 종합적 이해가 요구되는 수준으로 Mini Project 보다 난이도 있게 출제됩니다. 제출한 프로젝트에

대하여 리뷰어가 정해진 점검표에 맞추어 평가를 진행하고, 최종적으로 통과 하였을 시, 바이트 디그리가 수여됩니다.

05. What You Will Learn

1. 웹 프론트 엔드 기초인 JS 학습
2. 프론트엔드 개발자가 알아야하는 React를 이해하고, 규모가 큰 프로젝트에서 활용할 수 있는 중급 기술에 대해 학습
3. TypeScript를 활용한 React 프로젝트 실습

06. Course Summary

구분	파트	파트 목표	강사	감수자
Part 1	JavaScript 문법	JavaScript 문법부터 정리하며 기본기를 익힙니다. 변수와 상수, 자료형, 조건문, 반복문, 함수, 객체와 클래스 등 JavaScript 언어를 이해하고 비동기처리에 대한 개념까지 학습합니다. 나만의 책 리스트 만들기 프로젝트를 통해 JavaScript를 완성합니다.	이용재	박예준 (우아한형제들)
Part 2	React Component와 Styling	JavaScript를 바탕으로 React에 대해 학습합니다. 함수형 컴포넌트와 클래스형 컴포넌트, 상태관리 등 React의 기본적인 문법과 함께 다양한 컴포넌트 스타일링 방법에 대해 알아봅니다. 멋진 Todo 리스트 애플리케이션을 함께 만들어보며 학습한 내용을 정리하는 시간을 갖습니다.	김민준	안도형 (스튜디오씨드)
Part 3	API 연동 및 Router와 Redux	React 프로그래밍 중 중급 기술들을 학습합니다. API 연동, 라우터, 리덕스, 리덕스 미들웨어 등 서버연동에 필요한 기술들에 대해 알아봅니다. 위 기술들을 사용하는 것과 앞에서 진행했던 프로젝트가 어떻게 다른지 비교해가며 더 효율적으로 개발을 할 수 있는 스킬들을 배울 수 있습니다.	김민준	이용재 (스튜디오씨드)
Part 4	TypeScript와 React	TypeScript 문법에 대해 학습하고, React에서는 어떻게 TypeScript를 활용할 수 있을 지 배워봅니다. 타입 추론이 가능한 TypeScript를 사용하는 경우 사소한 실수를 방지할 수 있고, 코드의 가독성이 높아집니다. 실습을 통해 TypeScript와 React를 함께 사용하는 방법에 대해 익힐 수 있습니다.	고재도 / 김민준	이용재 (스튜디오씨드)

07. Course Schedule

- 총 수업 주차 : 12주 (수업 10주, 최종 프로젝트 2주)
- 총 수업 시간 : 1947 m

주차	기간	수업 시간(m)	특이사항
1	2020. 11. 23 ~ 2020. 11. 29	187	-
2	2020. 11. 30 ~ 2020. 12. 6	204	JavaScript Mini Project
3	2020. 12. 7 ~ 2020. 12. 13	185	-
4	2020. 12. 14 ~ 2020. 12. 20	200	-
5	2020. 12. 21 ~ 2020. 12. 27	181	React Mini Project 1
6	2020. 12. 28 ~ 2021. 1. 3	185	-
7	2021. 1. 4 ~ 2021. 1. 10	182	-
8	2021. 1. 11 ~ 2021. 1. 17	179	React Mini Project 2
9	2021. 1. 18 ~ 2021. 1. 24	194	-
10	2021. 1. 25 ~ 2021. 1. 31	249	-
최종 프로젝트(2주)	2021. 2. 1 ~ 2020. 2. 15	-	Final Project

08. Course Project

- 총 프로젝트 : 4개 (Mini Project 3개 / Final Project 1개)
- 바이트 디그리 수료 기준 : 3개 (Mini Project 2개 제출 / Final Project 통과)

1. Mini Project

a. JavaScript

- 출제자 : 박예준 (우아한형제들 프론트엔드개발)
- 기한 : 2020. 12. 14. 24:00
- 프로젝트 명 : 검색어 자동완성 사이트 구현
- 프로젝트 설명 : 배열 처리, string 자료형 가공, mock API를 활용한 비동기처리를 활용하여 검색어 자동 완성 서비스를 구현해봅니다. study를 입력하는 과정에서 자동 완성 검색어가 아래 나타나도록 자동완성 기능을 추가해봅니다.

b. React Component와 Styling

- 출제자 : 안도형 (스튜디오씨드 프론트엔드개발)
- 기한 : 2021. 1. 4. 24:00
- 프로젝트 명 : 나의 소비패턴 기록 서비스 구현
- 프로젝트 설명 : 투두리스트를 응용하여 나만의 가게부를 만들어봅니다. 오늘의 날짜 표시, 지출 등록 및 삭제, 지출의 총합 표시 등의 기능을 갖는 '나의 소비패턴 기록 서비스'를 구현해봅니다. 배열 렌더링, 상태관리, 커스텀 Hooks 등 리액트 기초파트에서 배웠던 모든 내용을 총 망라하여 만들 수 있습니다.

c. API 연동 및 Router와 Redux

- 출제자 : 박준영 (폴스택 웹 개발자)
- 기한 : 2021. 1. 25. 24:00
- 프로젝트 명 : 리액트 댓글 서비스 구현
- 프로젝트 설명 : 앞 파트에서 다뤘던 등록, 삭제, 수정 기능은 물론 페이지네이션 대댓글, 좋아요, 싫어요 등 추가적인 로직들을 활용하여 댓글서비스를 구현해봅니다. 댓글에 댓글이 있는 경우 원 댓글을 삭제한 후 "삭제된 댓글입니다"로 표현하는 등의 작업을 수행할 수 있습니다.

또한 댓글 기능이 구현된 서버 API와 연동하여 실습을 진행하므로 실무에서 필수적으로 알아야하는 내용들을 경험합니다.

2. Final Project

a. TypeScript와 React

- i. 출제자 : 이용재 (스튜디오씨드)
- ii. 기한 : 2021. 2. 15. 24:00
- iii. 프로젝트 명 : 나의 독서 기록 사이트 구현
- iv. 프로젝트 설명 : Django는 웹사이트를 백엔드부터 프론트엔드까지 매끄럽게 만들 수 있는 웹 프레임워크입니다. 웹사이트의 기본인 회원 관리부터 게시글 보기 및 작성, 데이터베이스 생성과 어드민 페이지 제작을 담은 '장고스타그램' 웹 플랫폼을 Django를 통해 제작합니다.

09. Byte Degree | Web Programming with Python 수료 기준

1. 7회 출석 체크

- a. 총 10주 동안 10번의 출석 체크를 진행하고 2/3 이상(7회 이상) 출석 인증을 해야합니다. (첫 개강일 1주차는 모든 수강생들에게 출석 인증 1회 부여)
- b. 출석 체크는 퀴즈 통과 여부를 통해 인증합니다.
- c. 매주 수업이 끝날 때, 10점 만점의 문제가 출제되며 5점 이상 득할 시, 1회 출석이 인정됩니다.

2. 2개 Mini Project 제출

- a. JavaScript 문법, React Component와 Styling, API연동 및 Router와 Redux 파트에서 총 3개의 Mini Project가 출제됩니다.
- b. 3개의 프로젝트 중 2개 이상의 Mini Project를 제출 하여야 합니다.
- c. 제출된 Mini Project는 리뷰어를 통해 피드백을 진행합니다.

3. Final Project 통과

- a. TypeScript와 React 파트에서 Final Project가 출제됩니다.
- b. 모든 수업이 종료된 이후 2주의 Final Project 출제 기간이 주어집니다.
- c. 리뷰어를 통해 피드백이 진행되고, 리뷰어는 공개된 기준표에 맞추어 Project 통과 여부를 결정하며 피드백을 진행합니다.
- d. 리뷰어의 '통과' 가 이루어진 수강생 중 출석체크, Mini Project 제출 수료 기준을 만족한 수강생에게 Byte Degree가 발급됩니다.

10. Course Curriculum

파트	상세 클립명	클립별 시간
JavaScript 문법	01. 자바스크립트란 무엇인가	3:35:00
	02. Various JavaScript Runtime	2:14:00
	03. 실습을 위한 준비	3:14:00
	04. Expression Statement	5:06:00
	05. Keywords, Reserved Words	3:40:00
	06. Identifier	4:39:00
	07. Comments	2:32:00
	08. 변수와 상수	5:05:00

09. 변수의 유효 범위	12:24:00
10. var 와 호이스팅	7:02:00
11. 자료형2	22:27:00
12. 조건문 - 1	10:12:00
13. 조건문 - 2	7:24:00
14. 조건문 - 3	11:57:00
15. 조건문 - 4	8:58:00
16_1. 반복문 - 1	8:10:00
16_2. 반복문 - 2	5:10:00
17. 함수 A	10:58:00
18. 함수 B	9:41:00
19. 함수 C	9:26:00
20_1. 객체 A	8:33:00
20_2. 객체 B	11:43:00
20_3. 객체 C	7:25:00
20_4. 객체 D	5:14:00
21. 클래스 A	6:40:00
22. 클래스 B	9:00:00
23. 클래스 C	11:39:00
24. 클래스 D	12:01:00
25. Promise - 1	10:01:00
27. Promise - 2	10:27:00
26. Promise - 3	8:39:00
28. Promise - 4	14:03:00
29. async function 과 await - 1	12:05:00
30. async function 과 await - 2	11:55:00
31. 프로젝트 개요	4:24:00
32. 리스트 페이지	20:26:00
33. 로그인 페이지	20:09:00
34. 책 추가하기	21:46:00
35. 책 정보 보기	13:51:00
36. 책 정보 수정	16:33:00

00. 강의 소개	3:20:00
01. 리액트는 어쩌다가 만들어졌을까	7:00:00
02. 작업환경 준비	9:28:00
03. 나의 첫번째 리액트 컴포넌트	6:47:00
04. JSX1	13:20:00
05. props 를 통해 컴포넌트에게 값 전달하기	8:18:00
06. 조건부 렌더링	5:02:00
07. useState 를 통해 컴포넌트에서 바뀌는 값 관리하기	10:45:00
08. input 상태 관리하기	5:16:00
09. 여러개의 input 상태 관리하기	9:33:00
10. useRef 로 특정 DOM 선택하기	4:10:00
11. 배열 렌더링하기	12:33:00
12. useRef 로 useRef 로 컴포넌트 안의 변수 만들기	6:05:00
13. 배열에 항목 추가하기	12:21:00
14. 배열에 항목 제거하기	8:12:00
15. 배열에 항목 수정하기	6:04:00
16. useEffect를 사용하여 마운트언마운트업데이트시 할 작업 설정하기	14:22:00
17. useMemo 를 사용하여 연산한 값 재사용하기	4:36:00
18. useCallback 를 사용하여 함수 재사용하기	6:44:00
19. React.memo 를 사용한 컴포넌트 리렌더링 방지	8:32:00
20. useReducer - 기초	7:25:00
21. useReducer - App 에서 useReducer 사용하기	15:19:00
22. 커스텀 Hook 만들어서 사용하기	8:23:00
23. Context API 를 사용한 전역 값 관리	9:19:00
24. UserDispatch Context 만들기	8:23:00
25. Immer 를 사용한 더 쉬운 불변성 지키기	9:52:00
26. 리듀서 Immer 로 구현하기	11:09:00
27. 클래스형 컴포넌트 소개	5:28:00
28. 클래스형 컴포넌트의 state 와 setState	14:09:00
29. Lifecycle 메서드	19:30:00
30. componentDidCatch 메서드	11:11:00
31. componentDidCatch 와 Sentry 연동	8:27:00
32. Prettier	9:20:00

33. ESLint 소개	8:13:00
34. ESLint 설정 커스터마이징	7:53:00
35. Snippet 사용하기	9:41:00
36. 리액트 입문 끝	0:48:00
[벨로퍼트's 컴포넌트 스타일링]	
00. Intro	1:46:00
01. Sass	11:43:00
02. Sass 버튼 A - size props 설정하기	8:24:00
03. Sass 버튼 B- color props 설정하기	6:36:00
04. Sass 버튼 C - outline,fullWidth props 설정	5:42:00
05. Sass 버튼 D - ___rest props 전달하기	7:12:00
06. CSS Module	17:44:00
07. Styled-components	16:05:00
08. Styled-components 버튼만들기	5:01:00
09. Styled-components - Polished 스타일 유틸 함수 사용	9:21:00
10. Styled-components 버튼 사이즈 조정하기	6:53:00
11. Styled-components 버튼 outline, fullWidth 설정	5:33:00
12. Styled-components Dialog 만들기	12:46:00
13. Styled-components 트랜지션 구현하기	11:02:00
[벨로퍼트's 멋진 투두리스트 만들기]	
00. TodoList - 인트로	1:36:00
01. TodoList - 프로젝트 생성 UI 준비 - 1	13:29:00
02. TodoList - 프로젝트 생성 UI 준비 - 2	20:31:00
03. Context API 를 활용한 상태관리	18:02:00
04. 기능 구현하기	11:52:00

API연동 및 Router와 Redux

[벨로퍼트와 함께 하는 모던 리액트 : API 연동]

00. 소개	1:30:00
01. API 연동의 기본	13:46:00
02. useReducer로 요청 상태 관리하기	8:31:00
03. useAsync 커스텀 Hook 만들어서 사용하기	16:25:00
04. react-async로 요청 상태 관리하기	8:25:00
05. Context 에서 비동기작업 상태 관리하기	20:29:00

06. Context에서 비동기 작업 상태 관리하기 - 리팩토링	15:26:00
07. 정리	2:08:00
[벨로퍼트와 함께 하는 모던 리액트 : Router]	
00. 소개	8:51:00
01. 프로젝트 준비 및 기본적인 사용법	8:53:00
02. 파라미터와 쿼리	10:45:00
03. 서브라우트	5:45:00
04. 라우터 부가기능 - history, withRouter	12:53:00
05. 라우터 부가기능 - Switch, NavLink, 기타.	7:48:00
07. 정리	0:58:00
[벨로퍼트와 함께 하는 모던 리액트 : Redux]	
00. 소개	7:00:00
01. 리덕스에서 사용되는 키워드 숙지하기	7:26:00
02. 리덕스의 3가지 규칙	13:44:00
03. 리덕스 사용 할 준비하기	14:26:00
04. 리덕스 모듈 만들기	15:28:00
05. 카운터 구현하기	11:15:00
06. 리덕스 개발자 도구 적용하기	3:01:00
07. 할 일 목록 구현 A - 프리젠테이션 컴포넌트 만들기	10:20:00
08. 할 일 목록 구현 B - 컨테이너 컴포넌트 만들기	5:20:00
09. 할 일 목록 구현 C - 컴포넌트 최적화를 위한 작업	3:15:00
10. useSelector 최적화	8:43:00
11. connect, HOC 함수 소개	4:42:00
12. connect 를 사용해서 컨테이너 구하기	11:12:00
13. connect에 대해 알아두면 좋은 것들	4:06:00
14. 정리	2:08:00
[벨로퍼트와 함께 하는 모던 리액트 : 리덕스 미들웨어]	
00. 소개	4:42:00
01. 리덕스 프로젝트 준비하기	10:11:00
02. 리덕스 미들웨어 만들어보고 이해하기	10:19:00
03. redux-logger 사용 및 미들웨어와 DevTools 함께 사용	5:05:00
04. redux-thunk	9:25:00
05. redux-thunk로 Promise 다루기 A - thunk 작성하기	14:20:00

06. redux-thunk로 Promise 다루기 B - reucerUtils 작성	13:30:00
07. redux-thunk로 Promise 다루기 C - createPromiseThunk 로 리팩토링하기	12:49:00
08. redux-thunk로 Promise 다루기 D - handleAsyncActions	8:15:00
09. redux-thunk로 Promise 다루기 E - 포스트 리스트 구현하기	7:01:00
10. redux-thunk로 Promise 다루기 F - 라우터 연동, 특정 포스트 읽기	12:43:00
11. 사용자 경험 개선 A - 포스트 리스트 데이터 유지 및 포스트 데이터 초기화	8:13:00
12. 사용자 경험 개선 B - 포스트 데이터 상태 구조 바꾸기	12:53:00
13. 사용자 경험 개선 C - createPromiseThunkByld, handleAsyncActionsByld 유틸함수 작성	9:34:00
14. Thunk 함수에서 리액트 라우터 History 사용하기	5:17:00
15. json-server	6:55:00
16. CORS and Proxy	5:13:00
17. redux-saga A - 소개, Generator 문법	12:27:00
18. redux-saga B - 기초 비동기 카운터 구현	13:18:00
19. redux-saga C - 프로미스 다루기	11:35:00
20. redux-saga D - 프로미스 관련 saga 리팩토링	7:20:00
21. redux-saga E - 라우터 연동	3:50:00
22. redux-saga F - select로 현재 상태 조회하기	3:16:00

TypeScript와 React

[TypeScript란 무엇인가]	
Chapter 00. Intro - 01. 코스소개	2:44:00
Chapter 01. TypeScript 시작하기 - 01. TypeScript 소개 및 개발환경 구성	10:03:00
Chapter 01. TypeScript 시작하기 - 02. TypeScript 컴파일러	20:16:00
Chapter 01. TypeScript 시작하기 - 03. TypeScript 컴파일러 설정파일	11:34:00
Chapter 01. TypeScript 시작하기 - 04. 변수선언	16:37:00
Chapter 01. TypeScript 시작하기 - 05. 기본타입	17:51:00
Chapter 01. TypeScript 시작하기 - 06. 인터페이스	18:39:00
Chapter 01. TypeScript 시작하기 - 07. 함수형 타입	14:41:00
Chapter 01. TypeScript 시작하기 - 08. enum 타입	9:34:00
Chapter 02. TypeScript - 01. 클래스 - 1	15:25:00
Chapter 02. TypeScript - 02. 클래스 - 2	12:57:00
Chapter 02. TypeScript - 03. 제네릭 - 1	8:48:00
Chapter 02. TypeScript - 04. 제네릭 - 2	4:49:00
Chapter 02. TypeScript - 05. 제네릭 - 3	11:05:00

Chapter 03. 고급타입 - 01. Intersection & Union Types	18:07:00	
Chapter 03. 고급타입 - 02. 타입 별칭	7:59:00	
Chapter 03. 고급타입 - 03. 인덱스타입	9:28:00	
Chapter 04. 실습 - 01. Parcel과 모듈	11:58:00	
Chapter 04. 실습 - 02. 프로젝트 구성 및 보드판 작성 - 1	14:49:00	
Chapter 04. 실습 - 02. 프로젝트 구성 및 보드판 작성 - 2	11:01:00	
Chapter 04. 실습 - 02. 프로젝트 구성 및 보드판 작성 - 3	11:11:00	
Chapter 04. 실습 - 03. 게임구현 - 1	20:03:00	
Chapter 04. 실습 - 03. 게임구현 - 2	19:55:00	
Chapter 04. 실습 - 03. 게임구현 - 3	16:34:00	
["벨로퍼트와 함께 하는 모던 리액트 : 타입스크립트와 리액트 함께 사용하기"]		
00. 소개	3:20:00	
01. 타입스크립트 연습 A - 작업환경 설정, 기본타입, 함수	15:43:00	
02. 타입스크립트 연습 B - Interface, Type Alias	13:36:00	
03. 타입스크립트 연습 C - Generics	9:57:00	
04. 타입스크립트로 리액트 컴포넌트 만들기	16:33:00	
05. 타입스크립트로 리액트 상태 관리 A - useState, 이벤트	13:04:00	
06. 타입스크립트로 리액트 상태 관리 B - useReducer	16:34:00	
07. 타입스크립트와 Context API 활용하기	12:09:00	
08. 타입스크립트와 리덕스 A - 기초	21:21:00	
09. 타입스크립트와 리덕스 B - 투두리스트 만들기	21:01:00	
10. typesafe-actions 로 리덕스 모듈 리팩토링	16:23:00	
11. 리덕스 모듈 여러 파일로 분리하기	8:00:00	
12. 타입스크립트와 리덕스 미들웨어 사용하기 A - redux-thunk1	19:34:00	
13. 타입스크립트와 리덕스 미들웨어 사용하기 B - 프리젠테이셔널 컴포넌트	11:01:00	
14. 타입스크립트와 리덕스 미들웨어 사용하기 C - 컨테이너 컴포넌트	4:54:00	
15. 타입스크립트와 리덕스 미들웨어 사용하기 D - thunk, 리듀서 리팩토링	24:20:00	
16. 타입스크립트와 리덕스 미들웨어 사용하기 E - redux-saga	7:26:00	
17. 타입스크립트와 리덕스 미들웨어 사용하기 F - saga 리팩토링	11:12:00	
18. 마무리	3:17:00	
<hr/>		
Git & GitHub	Chapter 00. Intro - 01. Introduction	6:24:00
	Chapter 01. 환경 설정하기 - 01. 버전 관리가 뭔가요	10:45:00
	Chapter 01. 환경 설정하기 - 02. 환경 설정하기 - Git 설치	5:13:00

Chapter 01. 환경 설정하기 - 03. 환경 설정하기 - GitHub, VSCode 설치	8:15:00
Chapter 02. 익히기 feat.CLI - 01. Git 초기화와 로컬 저장소	9:41:00
Chapter 02. 익히기 feat.CLI - 02. 첫번째 버전 만들기	12:27:00
Chapter 02. 익히기 feat.CLI - 03. 만든 버전 GitHub에 올리기	12:00:00
Chapter 02. 익히기 feat.CLI - 04. 다른 사람이 만든 저장소 받아오기	15:05:00
Chapter 03. 다지기 feat. GUI - 01. 소스트리 GUI로 Git 다지기	7:45:00
Chapter 03. 다지기 feat. GUI - 02. 그림으로 배우는 Add와 Commit	9:26:00
Chapter 03. 다지기 feat. GUI - 03.I GUI로 add, commit, push, pull 하기	8:47:00
Chapter 03. 다지기 feat. GUI - 04. 평행세계 나누기 - 브랜치(branch)	13:34:00
Chapter 03. 다지기 feat. GUI - 05. 두 버전 합치기 - 머지(merge)	4:14:00
Chapter 03. 다지기 feat. GUI - 06. 합치다가 충돌이 났어요 - 컨플릭트(Conflict)	16:58:00
Chapter 03. 다지기 feat. GUI - 07. 저장소 통째로 복제하기 - 포크(fork)	13:03:00
Chapter 03. 다지기 feat. GUI - 08. 내 코드를 머지해주면 안되겠니 - 풀 리퀘스트(Pull request)	12:28:00
Chapter 03. 다지기 feat. GUI - 09. 리뷰	11:36:00

(끝)