

교육 과정 소개서.

10개 프로젝트로 끝내는 Node.js의 모든 것(Express & Nest.js) 초격차 패키지 Online.



강의정보

강의장	온라인 강의 데스크탑, 노트북, 모바일 등
수강 기간	평생 소장
상세페이지	https://fastcampus.co.kr/dev_online_node23
강의시간	93시간 23분
문의	고객센터

강의특징

나만의 속도로	낮이나 새벽이나 내가 원하는 시간대 에 나의 스케줄대로 수강
------------	---

원하는 곳 어디서나	시간을 쪼개 먼 거리를 오가며 오프라인 강의장을 찾을 필요 없이 어디서든 수강
---------------	---

무제한 복습	무엇이든 반복적으로 학습해야 내것이 되기에 이해가 안가는 구간 몇번이고 재생
-----------	--



강의목표

- 백엔드 기능 개발, 테스트, 배포, 크롤링.. Node.js로 할 수 있는 모든 내용을 프로젝트를 통해 학습합니다.
- 일반적인 JavaScript 뿐만 아니라 유지보수성을 높여주는 TypeScript를 활용한 노트까지 배웁니다.
- 가장 대표적인 프레임워크 Express와 반박불가 대세 Nest.js를 한 번에 마스터할 수 있습니다.

강의요약

- Node.js의 모든 것을 70시간, 26가지 기술스택, 2가지 언어로 학습
- 10개 프로젝트로 필수 기능 구현부터 다양한 프레임워크의 실전 활용까지
- 풀스택 개발자로 성장하기 위한 쿠팡이츠 클론 코딩 프로젝트
- 오류를 줄이고 유지보수성을 높이는 TDD와 TypeScript 까지 학습
- JavaScript & TypeScript 문법과 기초강의 제공



강사

안재원	과목	- Part 1~9
약력		<p>[저서]</p> <ul style="list-style-type: none"> - 따라하며 배우는 도커와 CI환경 저서 저자 <p>[강의]</p> <ul style="list-style-type: none"> - 카카오 기업 신입 공채 FE 교수자 - 패스트 캠퍼스 온라인/오프라인 강의 진행 - 부산교육청, 무역 협회, 동서 발전, 국토연구원 등 금융기관 및 공공기관 사내 SNS 개발 - 코빗 타운, 신한 메타버스 개발 - 유튜브 Johnahn 채널 운영 (리액트 강의, 노드 강의) - 고등학교 코딩 강의 진행 - 멋쟁이 사자처럼 리액트 강의 진행 - 러닝 스폰즈 HTML & CSS 온라인 라이브 강의 진행 - 인프런 코딩 강의 진행 (도커 강의, 리액트 강의, 노드 강의, 파이어베이스 강의 등)
조승목	과목	- Part 11, Part 16
약력		<ul style="list-style-type: none"> - 현) 옴니아트 CTO - 전) 시지온 개발본부장 - 전) 퍼플랩 - 전) 누스랩
하성욱	과목	- Part 15
약력		<ul style="list-style-type: none"> - 현) N사 소프트웨어 엔지니어 - 전) 스타트업 CTO - 전) LendingHome Software Engineer <p>[프로젝트]</p> <ul style="list-style-type: none"> - N사 쇼핑검색 서비스 프론트엔드 개발 - 스마트오피스 자율좌석시스템 개발 - 미국 주택담보대출 자동심사 소프트웨어 개발 <p>[교육]</p> <ul style="list-style-type: none"> - 제로베이스 프론트엔드스쿨 강사 (React.js 핵심요약 강의)



강사

김신록

과목

- Part 12, Part 13

약력

- 현) 쿼타랩 소프트웨어 엔지니어
- 전) 스이드 개발팀 리드
- 전) 직방 CTO실 소프트웨어 엔지니어
- 전) 플리토 개발팀 소프트웨어 엔지니어

[프로젝트]

- 중학생 대상 실시간 석차 제공 서비스 개발
- 직방 거점 오피스 IoT 시스템 도입
- 번역 플랫폼

[교육]

- 스파르타 코딩클럽 강의

정경호

과목

- Part 14

약력

- 현) 그린랩스 백엔드 개발
- 전) NFTBank 백엔드 개발 리드
- 전) 아틀라스랩스 Node.js 백엔드 개발

[프로젝트]

- 농산물 거래 플랫폼 개발
- NFT 가치 평가 서비스 개발
- iOS, AOS 통화녹음, AI STT 서비스 개발

CURRICULUM

01.

웹 개발 기초를 위한 자바스크립트

파트별 수강시간 04:25:25

Ch01. 웹 개발 기초를 위한 자바스크립트

01. 챕터 설명
02. var, let, const
03. 변수의 참조 범위(scope)
04. 호이스팅
05. 자바스크립트 타입
06. Loops
07. Window Object
08. DOM 이란?
09. Document Object 이용해보기
10. DOM Event
11. Event 종류
12. OOP란?
13. 다형성이란?
14. javascript prototype
15. ES6 Classes
16. Sub Class(Inheritance)
17. super()란?
18. 자바스크립트 Closure
19. Javascript Event Loop
20. Callbacks, Promise 그리고 Async, Await
21. ES6 Promise
22. Async, Await



CURRICULUM

02.

Node.js 내부 동작 알아보기

파트별 수강시간 02:31:42

Ch01. Node.js란?
01. 자바 스크립트 엔진
02. Node.js History
03. Node.js Runtime
04. Node.js 설치하기
05. Node.js REPL
06. Browser API & Node.js Api
07. global 객체에 대해서
Ch02. Node.js 의 내부
01. Node.js가 작업을 처리하는 방법
02. Node 오픈 소스 코드를 통한 이해
03. libuv
04. 비동기와 동기란 ? (asynchronous vs synchronous)
05. 코드로 보는 async & sync
06. javascript는 Synchronous 언어
07. blocking, non-blocking in node.js
08. 프로세스 및 스레드
09. 노드 JS가 비동기 작업을 처리하는 방법
10. Event Loop이란?
11. Event Loop 심화
12. setImmediate vs setTimeout vs process.NextTick
13. Node.js Event Emitter

CURRICULUM

03.

Node.js 모듈 시스템 & 패키지 관리

파트별 수강시간 01:44:35

Ch01. Node.js 모듈 시스템

01. Module 이란?
02. HTTP 서버 구축하기
03. 모듈을 사용하는 이유
04. 모듈 생성하기
05. 모듈에서 export 하는 방법
06. CommonJS와 ECMA script 모듈의 차이
07. Node.js에서 ECMAScript 모듈 이용하기
08. 모듈 캐싱에 대해서
09. index.js에 대해서

Ch02. Node.js 패키지 관리

01. npm 이란?
02. npm package 설치하기
03. node_module 살펴보기
04. Semantic Versioning
05. package-lock.json
06. npm audit
07. npm install -g
08. Yarn 이란?



CURRICULUM

[+ Bonus +]

JavaScript 선행 학습

파트별 수강시간 10:13:05

Ch 01. JS 데이터
01. 개요 및 프로젝트 구성
02. 원시형 - String, Number
03. 원시형 - Boolean, null, undefined
04. 참조형 - Array
05. 참조형 - Object
06. 참조형 - Function
07. 형 변환(Type Conversion)
08. 참과 거짓(Truthy & Falsy)
09. 데이터 타입 확인
Ch 02. 연산자와 구문
01. 산술, 할당, 증감 연산자
02. 부정, 비교 연산자
03. 논리 연산자
04. Nullish 병합, 삼항 연산자
05. 전개 연산자
06. 구조 분해 할당
07. 선택적 체이닝
08. If, Switch 조건문
09. For, For of, For in 반복문
10. While, Do while 반복문
Ch 03. 함수
01. 선언과 표현 그리고 호이스팅
02. 반환 및 종료
03. 매개변수 패턴
04. 화살표 함수
05. 즉시실행함수(IIFE)
06. 콜백
07. 재귀
08. 호출 스케줄링
09. this

CURRICULUM

[+ Bonus +]

JavaScript 선행
학습

파트별 수강시간 10:13:05

Ch 04. 표준 내장 객체
01. 문자 1
02. 문자 2
03. 숫자
04. 수학
05. 날짜
06. 배열 1
07. 배열 2
08. 배열 3
09. 배열 4
10. 객체
11. JSON
Ch 05. DOM
01. Node vs Element
02. 검색과 탐색
03. 생성, 조회, 수정 1
04. 생성, 조회, 수정 2
05. 크기와 좌표



CURRICULUM

[+ Bonus +]

TypeScript 선행 학습

파트별 수강시간 04:08:07

Ch 01. 문법
01. 개요 및 빠른 개발 환경 구성
02. 타입 종류 알아보기 - 문자부터 함수까지
03. 타입 종류 알아보기 - Any부터 Intersection까지
04. 타입 추론(Inference)
05. 타입 및 할당 단언(Assertions)
06. 타입 가드(Guards)
07. 인터페이스 - 기본
08. 인터페이스 - 함수 타입
09. 인터페이스 - 인덱싱 가능 타입
10. 인터페이스 - 확장(상속)
11. 타입 별칭(Alias)
12. 함수 - 명시적 this 타입
13. 함수 - 오버로딩(Overloading)
14. 클래스와 접근 제어자
15. 제네릭 - 함수
16. 제네릭 - 클래스
17. 제네릭 - 인터페이스, 제약 조건
18. 패키지의 타입 선언
19. 타입 가져오기와 내보내기
20. tsconfig.json 구성 옵션

CURRICULUM

04.

Node.js로 웹
개발하기

파트별 수강시간 02:51:54

Ch 01. 웹 서버 만들기
01. 웹 서버란?
02. HTTP Requests
03. Stateless Protocol
04. HTTP Request, Response 구조
05. HTTP Status Codes
06. 웹 서버 생성하기
07. HTTP Routing
08. Post 요청으로 데이터 추가하기
Ch 02. Express.js
01. Express를 사용하는 이유
02. Express js 기본 구조 코드 생성
03. req.params
04. res.json() vs res.send()
05. res.send() vs res.end()
06. 포트맨 설치하기
07. nodemon 설치
08. middleware에 대해서
09. middleware 생성하기
10. POST 요청하기 & express.json()
11. Model, View, Controller
12. 앱 소스 코드 MVC 패턴으로 변경해주기
13. Router, Controller
14. RESTful API
15. Express에서 파일 전송하기(res.sendFile)
16. express.static()
17. Template Engine 사용하기
18. Template Layout 생성하기



CURRICULUM

05.

**로그인, 인증
기능 구축
프로젝트[미니
프로젝트 1]**

파트별 수강시간 05:24:27

Ch 01. jwt를 이용한 인증 앱 생성하기
01. 인증 서비스 구현을 위해서 필요한 지식
02. JWT에 대하여
03. 간단한 인증 시스템 구현
04. 토큰을 이용해서 요청 보내기
05. RefreshToken에 대해서
06. RefreshToken 생성하기
07. RefreshToken으로 accessToken 생성하기
Ch 02. passport를 이용한 인증 앱 생성하기
01. Passport 모듈을 이용한 인증 시스템 구현
02. sparse indexes
03. 로그인 회원가입을 위한 화면 구현하기
04. Passport를 이용한 로그인 회원가입 기능 구현하기
05. Passport를 이용한 세션 생성하기
06. 인증을 위한 미들웨어 생성
07. 로그아웃 기능 만들기
08. 비밀번호를 데이터베이스에 저장하는 방법
09. 비밀번호 암호화 코드 작성하기
10. Google OAuth Key 생성
11. Google OAuth 구현(소셜 로그인)
12. 설정 파일 및 환경 변수 생성
13. Router를 이용한 구조 정리
14. Email 기능 구현하기
15. 구글 이메일 앱 비밀번호 생성하기
16. SMTP(Simple Mail Transfer Protocol)
17. 카카오 OAuth Key 생성
18. 카카오 OAuth 구현(소셜 로그인)



CURRICULUM

06.

데이터베이스

파트별 수강시간 02:54:30

Ch 01. Database에 대해서
01. Database 란
02. SQL vs NoSQL
03. MongoDB 사용하기
04. Mongoose에 대해서
05. Model 및 Schema 생성하기
06. CRUD 구현하기 (Create)
07. Read 구현하기
08. Update, Delete 구현하기
09. Express 에러 처리
Ch 02. ORM에 대해서
01. ORM 에 대해서
02. TypeORM 사용을 위한 준비
03. Typeorm을 이용해서 앱과 데이터베이스 연결하기
04. 윈도우에서 도커 설치하기
05. 맥에서 도커 설치하기
06. 도커를 이용한 Postgres 실행하기
07. pgAdmin 사용법
08. Typeorm을 이용한 CRUD 구현
09. Sequelize 사용하기
10. Sequelize를 이용한 CRUD 구현

CURRICULUM

07.

GraphQL 기본

파트별 수강시간 02:06:53

Ch 01. GraphQL 기본
01. GraphQL 이란
02. GraphQL 사용해보기
03. GraphQL의 장단점
04. Express GraphQL Server 생성하기
05. GraphiQL 사용하기
06. Schema 작성하기
07. GraphQL Tools
08. Resolver
09. Resolvers 모듈화 하기
10. 필터링 기능 추가하기
11. ID로 데이터 가져오기
12. Mutation
13. Comment 생성하기
14. Apollo 란 무엇인가요-
15. Apollo Server V3
16. Migrating to Apollo Server 4

CURRICULUM

08.

**실시간 채팅 구축
프로젝트[미니
프로젝트 2]**

파트별 수강시간 05:29:36

Ch 01. Socket IO 기본
01. Rest vs WebSocket
02. Polling 에 대해서
03. Websocket 이용하기
04. Socket IO
05. Socket io를 이용한 간단한 메시지 앱 생성하기
06. Express Socket io 이용한 채팅 앱 구현
07. Socket io 연동하기
08. 유저가 방에 들어왔을 때 유저 추가하기
09. 유저가 방에 들어왔을 때 환영 메시지 추가하기
10. 채팅방 입장 시 화면 구현하기
11. Mustache 사용해서 메시지 보여주기
12. 채팅 페이지 스타일 적용하기
13. 채팅방에서 메시지 보내기
14. 채팅방에서 나가기
15. Namespace vs Rooms
Ch 02. Private Chat 구현하기(DM)
01. 앱 기본 구조 생성
02. 앱 기본 UI 생성하기
03. Socket io 연동하기
04. 몽고 DB 연동하기
05. 유저 세션 생성하기
06. 유저 세션 생성 코드 작성하기
07. 메시지 보낼 상대 선택하기
08. 메시지 보내기
09. 서버에서 클라이언트로 메시지 전달하기
10. 채팅방 나가는 로직 구현하기
11. 데이터베이스에서 메시지 가져오기
12. 채팅앱 스타일링 해주기



CURRICULUM

09.

SNS 서비스 구축
프로젝트[미니
프로젝트 3]

파트별 수강시간 06:18:49

Ch 01. SNS 앱 만들기
01. 만들게 되는 앱 소개
02. 앱 기본 구조 생성하기
03. 로그인, 회원가입 페이지 생성
04. Header, Footer
05. 모델 생성하기
06. 포스트 UI 생성하기
07. 이미지 업로드 및 포스트 생성하기
08. 에러 처리기 생성하기
09. 포스트 리스트 나열하기
10. 포스트 리스트 UI 생성하기
11. Connect-Flash에 대해서
12. Connect-Flash를 앱에서 사용하기
13. 포스트 수정하기
14. 포스트 수정 페이지 UI 생성하기
15. 포스트 삭제하기
16. 댓글 UI 생성하기
17. 댓글 기능 생성하기
18. 댓글 삭제하기
19. 댓글 수정하기
20. 좋아요 기능 생성하기
21. Friends 페이지 생성하기
22. Friends 페이지를 위한 UI 생성하기
23. Friends 페이지 기능 생성하기
24. 친구 요청 수락 기능 생성하기
25. Header UI 생성하기
26. Profile 페이지 데이터 생성하기
27. Profile 페이지 UI 생성하기
28. Profile 수정하기
29. Profile 수정 페이지 UI 생성하기

CURRICULUM

10.

결제 시스템 구축 프로젝트[미니 프로젝트 4]

파트별 수강시간 05:03:08

Ch 01. 쇼핑몰 앱 관리자 페이지 생성하기

01. 만들게 되는 앱 소개
02. 앱 기본 구조 생성하기
03. 로그인, 회원가입 페이지 생성
04. Header, Footer
05. 모델 생성하기
06. Connect-Flash 적용하기
07. 카테고리 생성 페이지 만들기
08. 카테고리 리스트 나열
09. 상품 생성 페이지 만들기
10. FileReader API

11. express fileupload
12. 상품 리스트 UI 페이지 생성하기
13. 상품 삭제하기
14. 상품 수정하기
15. DropZone
16. 원본 및 썸네일 이미지 삭제하기

Ch 02. 상품 페이지 및 장바구니 페이지 생성하기

01. 메인 페이지 생성하기(상품 페이지)
02. 카테고리 페이지 생성하기
03. 상품 상세 페이지 생성하기
04. Header UI 생성하기
05. 장바구니에 상품 추가하기
06. 장바구니 페이지 생성하기
07. 장바구니 로직들 생성하기
08. express-session으로 변경하기
09. 결제시스템 연동하기



CURRICULUM

11.

**프로젝트 5 :
Nest.js 기본 활용
게시판 서비스
구현하기**

파트별 수강시간 08:36:08

Ch 01. Nest.js 소개
01. 강의소개
02. Nest JS 란
03. Nest JS 를 해야하는 이유
04. 왜 Nest.js 는 대세가 되었는가
Ch 02. Nest JS 기본 개념
01. Nest JS 의 기본 구조
02. 프로젝트 구성
03. Module
04. Controller
05. Service
06. DI 이해하기
Ch 03. Nest JS로 API 설계하기
01. 실습 프로젝트 - 게시판 만들기
02. 사전준비 - node.js, yarn 설치
03. 사전준비 - Nest JS CLI 설치 및 프로젝트 생성
04. 사전준비 - IDE 셋팅 (vscode, 플러그인)
05. Controller 생성 및 샘플 API 구현
06. File Naming Conventions
07. Routing - 게시글 생성 가져오기 업데이트 삭제 만들기
08. Module 생성
09. Service 생성
10. 게시글 가져오기 구현
11. 게시글 작성 구현
12. 게시글 업데이트 구현
13. 게시글 삭제 구현
14. API 문서화 (Swagger)
Ch 04. Nest JS를 활용한 좀더 실용적인 API 구현
01. DTO 란-
02. 게시글 생성 DTO 구현
03. Validation 적용
04. 나머지 DTO 구현
05. Nest JS 데코레이터-
06. 나만의 커스텀 데코레이터 만들기
07. Exception filters를 이용한 예외처리
08. LoggerModule 을 이용한 로깅 처리
09. ConfigueModule 적용하여 환경설정 적용



CURRICULUM

11.

**프로젝트 5 :
Nest.js 기본 활용
게시판 서비스
구현하기**

파트별 수강시간 08:36:08

Ch 05. 데이터베이스 연동을 위한 TypeORM 소개
01. TypeORM 이란
02. PostgreSQL 설치 및 셋팅
03. TypeORM Entity 작성
04. TypeORM 관계 설정
05. TypeORM Migrations 를 이용하여 PostgreSQL 에 테이블 생성하기
06. TypeORM Seeding 을 통한 초기 데이터 생성
07. TypeORM 기반으로 게시글 가져오기 수정
08. TypeORM 기반으로 게시글 작성 수정
09. TypeORM 기반으로 게시글 업데이트 수정
10. TypeORM 기반으로 게시글 삭제 수정
Ch 06. Nest JS 인증 기능 구현
01. 인증기능 추가하기위한 작업명세
02. 회원가입 - User Module, Controller, Service 생성
03. 회원가입 - User Entity 구현
04. 회원가입 - 회원가입 기능 구현
05. 회원가입 - 비밀번호 암호화
06. 로그인 - 로그인 기능 구현
07. 로그인 - 인증 토큰 생성하기 (JWT)
08. 로그인 - Passport 적용한 로그인 구현
09. 로그인 - 토큰 인증 API 구현
10. 로그인 - 객체직렬화를 통한 데이터 커스텀
11. 게시판에 회원정보 적용하기
12. Nest JS Guard 를 이용하여 자신의 글 보호하기
Ch 07. Nest JS를 배포해보자
01. 프로젝트 빌드하기
02. pm2 를 이용한 노드 프로세스 관리
03. 유닛테스트 적용
04. e2e테스트 적용



CURRICULUM

12.

**프로젝트 6 :
Nest.js 고급 활용
개발 심화 및
테스트, 배포**

파트별 수강시간 09:23:42

Ch 01. 강의 소개 및 템플릿 프로젝트 분석
01. 강의 소개
02. 실습 환경 구축
03. 템플릿 프로젝트 분석
Ch 02. Decorator 심화
01. Decorator 파트 1
02. Decorator 파트 2
03. Validation, Transform 에서의 활용
04. Swagger 에서의 활용 1
05. Swagger 에서의 활용 2
06. 커스텀 매개변수 Decorator 만들기 1
07. 커스텀 매개변수 Decorator 만들기 2
08. 커스텀 매개변수 Decorator 만들기 3
Ch 03. Provider 심화
01. Provider 종류
02. 테스트에서의 활용
Ch 04. Module 심화
01. 마이크로서비스를 염두해 둔 설계
02. 동적 모듈 활용
Ch 05. Authentication(인증), Authorizatio(인가) 심화
01. Authentication, Authorization 개념
02. 슬라이딩 세션과 리프레쉬 토큰 구현 1
03. 슬라이딩 세션과 리프레쉬 토큰 구현 2
04. 메타데이터 개념
05. Authorization 구현
Ch 06. 미들웨어와 인터셉터
01. Middleware와 Interceptor 개념. 로깅 기능
02. nest-winston을 활용한 로깅
03. Interceptor를 활용한 요청, 응답 매핑
Ch 07. DB, ORM 심화
01. Transaction 개념
02. Transaction 구현
03. Index 개념 및 구현
04. 쿼리 분석 및 개선 1
04. 쿼리 분석 및 개선 2

CURRICULUM

12.

프로젝트 6 : Nest.js 고급 활용 개발 심화 및 테스트, 배포

파트별 수강시간 09:23:42

Ch 08. 보안과 앱 상태 관리
01. Swagger 기본 보안 적용
02. Hashing 으로 암호 처리
03. Brutal force 공격을 예방하기 위한 Rate limit
04. Sentry를 활용하여 에러 발생시 슬랙 알람 구축
05. Health 체크
Ch 09. CQRS
01. CQRS 적용 1
02. CQRS 적용 2
Ch 10. 앱을 완성하기 위한 기능 구현
01. 파일 업로드 구현
02. Task 스케줄링 구현 1
03. Task 스케줄링 구현 2
Ch 11. 테스트 및 배포
01. 테스트 케이스 작성
02. 테스트 커버리지
03. CI CD 개념
04. 운영환경 인프라 생성
05. Github Actions를 활용한 CD 적용 (1)
06. Github Actions를 활용한 CD 적용 (2)
07. Github Actions를 활용한 CD 적용 (3)



CURRICULUM

13.

**Nest.js 기반
MSA 프로젝트**

파트별 수강시간 02:28:48

Ch 01. MSA 이해하기
01. 강의 소개
02. 모놀리스 vs 마이크로서비스
03. 마이크로서비스 아키텍처 프리뷰
Ch 02. 첫 번째 마이크로서비스 띄우기
01. NestJS 마이크로서비스 이해
02. API Gateway 생성
03. docker 개념 이해
04. docker container로 마이크로서비스 띄우기
Ch 03. 도커 컴포즈로 마이크로서비스 띄우기
01. docker compose 개념 이해
02. docker compose로 마이크로서비스 띄우기
03. Postgres 추가
Ch 04. 마이크로서비스간의 동기 통신
01. 동기 vs 비동기 통신 개념 이해
02. 동기 통신을 할 마이크로서비스들 구현 1
03. 동기 통신을 할 마이크로서비스들 구현 2
04. TCP를 사용한 동기 통신 구현
Ch 05. 마이크로서비스간의 비동기 통신
01. Kafka 개념 이해
02. 비동기 통신을 할 마이크로서비스 구현
03. Kafka를 사용한 비동기 통신 구현
Ch 06. 로컬 개발환경에서의 MSA 완성하기
01. 마이크로서비스 리뷰
02. 운영 환경 구축을 위해 필요한 기술들 안내



CURRICULUM

14.

**Node.js
유지보수를 위한
TDD 실전**

파트별 수강시간 05:09:30

Ch 01. 테스트의 기본
01. TDD 테스트 코드 기본 개념
02. 유닛테스트
03. 통합 테스트, E2E 테스트
04. cypress를 이용해서 e2e 테스트 이해하기
05. BDD와 TDD 차이점 알아보기
06. Jest에 대해서
07. Jest 파일 구조 & 사용법
08. jest.fn()
09. jest.spyOn()
10. NestJS 에서 테스트하기
11. 테스트 커버리지
12. Github Action으로 테스트 자동화
Ch 02. TDD를 이용한 Express 앱 생성 (Create)
01. Tdd를 이용한 express crud 앱 생성 준비하기
02. Create Product 함수 생성
03. Create Method로 데이터 저장하기
04. node-mocks-http
05. beforeEach
06. 상태 값 전달
07. 결과 값 전달
08. 임의로 데이터를 저장하기
09. 에러 처리를 위한 단위 테스트 작성



CURRICULUM

14.

**Node.js
유지보수를 위한
TDD 실전**

파트별 수강시간 05:09:30

Ch 03. TDD를 이용한 Express 앱 생성 (Read)
01. 통합 테스트에 대해서
02. 통합 테스트 작성하기
03. 에러 처리를 위한 통합 테스트
04. Express 에러 처리에 대해서
05. Read 시작
06. getProducts 단위 테스트 작성 (1)
07. getProducts 단위 테스트 작성 (2)
08. getProducts 에러 처리 단위 테스트 작성
09. getProducts 통합 테스트 작성
10. getProductById 단위 테스트 작성 (1)
11. getProductById 단위 테스트 작성 (2)
12. getProductById 통합 테스트 작성 (1)
13. getProductById 통합 테스트 작성 (2)
Ch 04. TDD를 이용한 Express 앱 생성 (Update, Delete)
01. Update 시작
02. updateProduct 단위 테스트 작성 (1)
03. updateProduct 단위 테스트 작성 (2)
04. updateProduct 통합 테스트 작성(1)
05. updateProduct 통합 테스트 작성 (2)
06. Delete 시작
07. deleteProduct 단위 테스트 작성 (1)
08. deleteProduct 단위 테스트 작성(2)
09. deleteProduct 통합 테스트 작성



CURRICULUM

15.

**MERN 스택
쿠팡이츠
클론코딩**

파트별 수강시간 09:23:20

Ch 01. React 기본 개념에 대하여 알아보자 (Component / Props / State / Hooks)
01. React란 무엇인가
02. React 개발환경 설정하기
Ch 02. Component와 Hook
01. Component와 Props State 이해하기
02. Class Component에 대하여
03. React Hook이란
Ch 03. CSR 과 SSR
01. Routing 과 Next.js
Ch 04. 상태관리 및 쿼리
01. 상태는 이렇게 관리해요
02. 서버에서 데이터는 이렇게 가져와요
Ch 05. 이쁘게 만들기
01. Styling하기
Ch 06. Docker에 대한 이해
01. Docker란
Ch 07. Docker 설치 및 활용
01. Docker 설치하기
02. Docker 활용하기 (Dockerfile 작성하기)
Ch 08. 기획 및 분석해보기
01. 쿠팡이츠 사용해보기
02. 필요한 기능 분석해보기
Ch 09. 데이터 준비하기
01. Atlas를 이용해 몽고DB 생성하기
02. MongoDB Schema Model 설계하기
03. Mock Data 만들기
Ch 10. 백엔드 구현하기
01. API 구성하기
02. API 기능 구현하기 1
03. API 기능 구현하기 2
04. API 기능 구현하기 3
05. Docker 로 빌드하고 배포해보기



CURRICULUM

15.

MERN 스택
쿠팡이츠
클론코딩

파트별 수강시간 09:23:20

Ch 11. 프론트엔드 프로젝트 설정하기
01. create-next-app을 통해 프론트엔드 프로젝트 구성하기
Ch 12. 로그인 구현하기
01. Next-auth.js 를 통한 로그인 구현
02. Google OAuth를 통한 로그인 구현
03. 로그인페이지 구현하기
Ch 13. 프론트엔드 구현하기
01. 메인페이지 - 매장 탐색페이지 구현하기 1
02. 메인페이지 - 매장 탐색페이지 구현하기 2
03. 메인페이지 - 매장 탐색페이지 구현하기 3
04. 무한스크롤 및 카테고리 페이지 구현하기
05. 검색페이지 구현하기
06. 메뉴 탐색 페이지 구현하기 1
07. 메뉴 탐색 페이지 구현하기 2
08. 주문 페이지 구현하기-
09. 결제 페이지 구현하기
10. 배달내역 페이지 구현하기 1
11. 배달내역 페이지 구현하기 2
12. 리뷰 등록 페이지 구현하기 1
13. 리뷰 등록 페이지 구현하기 2
14. 리뷰 페이지 구현하기 1
15. 리뷰 페이지 구현하기 2
16. 프론트엔드 배포해보기



CURRICULUM

16.

**Node.js 크롤링
프로젝트 :
개발지식 구독
서비스 구축하기**

파트별 수강시간 05:10:18

Ch 01. 프로젝트 소개
01. 프로젝트 소개
Ch 02. 크롤러 사용하기
01. 크롤러 - 프로젝트 셋팅
02. 크롤러 - puppeteer 기본 사용법
03. 크롤러 - medium 에서 아티클 수집하기
04. 크롤러 - 크롤링 스케줄링 동작 설정
05. 크롤러 - Firebase 설정하기
06. 크롤러 - Firebase Store 에 수집데이터 저장하기
07. 크롤러 - Firebase Cloud Function 에 크롤러 배포하기
08. 크롤러 - Cloud Function 으로 스케줄링 동작 설정
Ch 03. 서비스 구현하기
01. 서비스 - Next.js 설정하기
02. 서비스 - 메인 화면 구현
03. 서비스 - Firebase 연동하기
04. 서비스 - Next.js API 를 이용하여 크롤링 데이터 가져오기
05. 서비스 - 키워드 입력 화면 만들기
06. 서비스 - 키워드 입력 API 구현
Ch 04. 마무리 단계
01. 크롤러 - 등록된 키워드의 아티클을 수집하도록 수정
02. 서비스 - Vercel 을 통해 배포하기

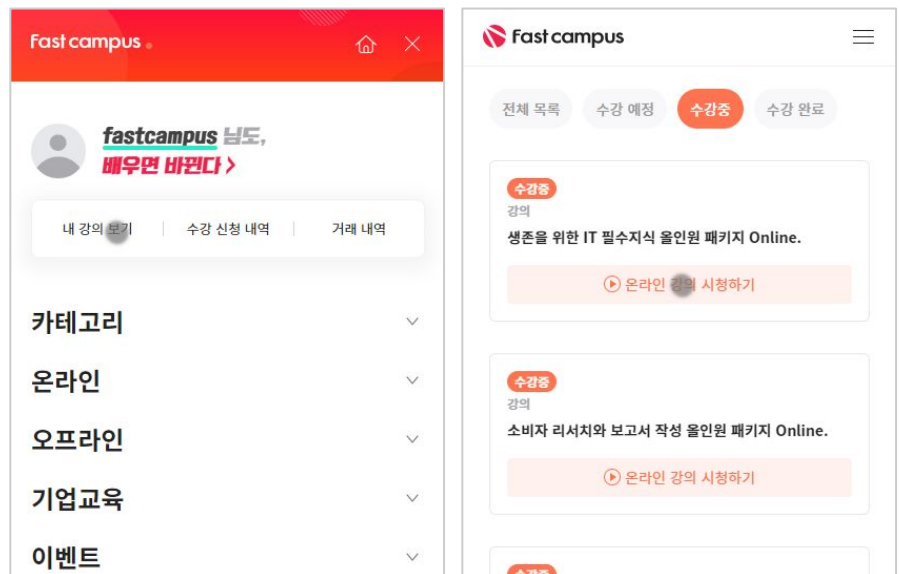


주의 사항

- 상황에 따라 사전 공지 없이 할인이 조기 마감되거나 연장될 수 있습니다.
- 패스트캠퍼스의 모든 온라인 강의는 아이디 공유를 금지하고 있으며 1개의 아이디로 여러 명이 수강하실 수 없습니다.
- 별도의 주의사항은 각 강의 상세페이지에서 확인하실 수 있습니다.

수강 방법

- 패스트캠퍼스는 크롬 브라우저에 최적화 되어있습니다.
- 사전 예약 판매 중인 강의의 경우 1차 공개일정에 맞춰 '온라인 강의 시청하기'가 활성화됩니다.



환불 규정

- 온라인 강의는 각 과정 별 '정상 수강기간(유료수강기간)'과 정상 수강기간 이후의 '복습 수강기간(무료수강기간)'으로 구성됩니다.
- 환불금액은 실제 결제금액을 기준으로 계산됩니다.

수강 시작 후 7일 이내	100% 환불 가능 (단, 수강하셨다면 수강 분량만큼 차감)
수강 시작 후 7일 경과	정상(유료) 수강기간 대비 잔여일에 대해 환불규정에 따라 환불 가능

※ 강의별 환불규정이 상이할 수 있으므로 각 강의 상세페이지를 확인해 주세요.