

# 교육 과정 소개서.

---

100가지 시나리오로 학습하는 프론트엔드 : 5년 이상 경험을  
초압축한 실전 문제 해결 패키지



## 강의정보

|       |   |
|-------|---|
| 강의장   | 온라인 강의   데스크탑, 노트북, 모바일 등   |
| 수강 기간 | 평생 소장   |
| 상세페이지 | <a href="https://fastcampus.co.kr/dev_online_fe100">https://fastcampus.co.kr/dev_online_fe100</a> |
| 강의시간  | 59시간  |
| 문의    | <a href="#">고객센터</a>  |

## 강의특징

|            |   |
|------------|---|
| 나만의<br>속도로 | 낮이나 새벽이나<br><b>내가 원하는 시간대</b> 에 나의 스케줄대로 수강 |
|------------|---|

|               |   |
|---------------|---|
| 원하는 곳<br>어디서나 | 시간을 쪼개 먼 거리를 오가며<br>오프라인 강의장을 찾을 필요 없이 <b>어디서든 수강</b> |
|---------------|---|

|           |  |
|-----------|--|
| 무제한<br>복습 | 무엇이든 반복적으로 학습해야<br>내것이 되기에 이해가 안가는 구간 <b>몇번이고 재생</b> |
|-----------|--|



## 강의목표

- 성공하는 프로젝트&조직의 개발자가 주니어부터 시니어까지 성장하며 겪는 5년 이상 경험을 90시간으로 초압축 학습하실 수 있습니다.
- 리드&시니어 개발자와 함께 자주 겪고, 까다로운 문제를 해결하며 실전에 바로 적용하는 테크 스킬을 학습해봅니다.
- PM과 개발자가 작성한 고퀄 자료를 활용해 실제 팀 단위 프로젝트의 문제를 해결하듯이 과제를 진행하며 문제해결력을 상승시켜봅니다.

## 강의요약

- 연차별로 마주하는 100가지 문제상황 해결로 배우는 100가지 프론트엔드 기술을 배워봅니다.
- 채용 과제 대비 '조건에 맞는 구현 과제'부터, 실무자로 바탕 '응용 과제'까지 실제적인 경험과 노하우를 쌓아봅니다.
- 상용 환경과 성장하는 프로젝트에서 마주하는 문제들을 상황 및 문제 해결에 필요한 기술들을 상황에 맞춰 이해하기 쉽게 학습할 수 있습니다.



## 강사

|     |    |  |
|-----|----|--|
| 안재원 | 과목 | - (주니어)Part 1 ~ Part 4   |
|     |    | <b>약력</b> <ul style="list-style-type: none"> <li>- 카카오 신입 공채 FE 교수자</li> <li>- 프론트엔드 온오프라인 강사</li> <li>- '따라하며 배우는 도커와 CI환경' 저자</li> <li>- 각종 금융기관 및 공공기관용 SNS 개발</li> </ul> |
| 박세문 | 과목 | - (미드레벨)Part 5 ~ Part 13   |
|     |    | <b>약력</b> <ul style="list-style-type: none"> <li>- 현) 우아한형제들 Senior 개발자</li> <li>- 전) Agoda   EA Korea 등 프론트엔드 개발 15년차</li> </ul>  |
| 마광휘 | 과목 | - (시니어)Part 14 ~ Part 19 & Special   |
|     |    | <b>약력</b> <ul style="list-style-type: none"> <li>- 현) 우아한형제들 배민커머스 웹프론트 개발 팀장</li> <li>- 전) 와인포인트 등 프론트엔드 개발 11년차</li> </ul>   |



CURRICULUM

01.

대규모 이커머스  
서비스 프로젝트  
개발하기 -  
프론트유저단

파트별 수강시간 14:42:57

|  |
|--|
| <b>CH01. Nextjs 살펴보기 및 리액트 앱 생성하기</b>  |
| 00. 강의 및 강사 소개   |
| 01. Nextjs를 이용한 리액트 앱 생성하기   |
| 02. 전체 폴더 및 파일 구조 생성하기   |
| 03. 컴포넌트 폴더 및 파일 생성하기  |
| 04. 남은 폴더 및 파일 생성하기  |
| 05. 애플리케이션 만드는데 필요한 패키지들 설치하기  |
| <b>CH02. Shop App 기본 구조 및 UI 생성하기</b>  |
| 01. 전역 스타일 적용하기  |
| 02. 쿠팡 로그인 UI 생성하기   |
| 03. Loader 컴포넌트 생성하기   |
| 04. React Portal 로 모달 생성하기[참조]   |
| 05. Input 컴포넌트 생성하기  |
| 06. Input 컴포넌트 스타일링하기  |
| 07. Icon 컴포넌트 생성하기   |
| 08. 자동 로그인 컴포넌트 생성하기   |
| 09. Tooltip 컴포넌트 생성하기  |
| 10. Divider 컴포넌트 생성하기  |
| 11. Button 컴포넌트 생성하기   |
| 12. 쿠팡 회원가입 UI 생성하기  |
| 13. 비밀번호 수정 페이지 UI 생성하기  |
| 14. Heading 컴포넌트 생성하기  |
| <b>CH03. Firebase 연결하기 ( Firebase Auth   Firebase Storage   Firebase Firestore )</b> |
| 01. [모든 페이지] 애플리케이션에 파이어베이스 연결하기   |
| 02. [모든 페이지] 알림을 위한 React Toastify 적용하기  |
| 03. [회원가입/로그인 페이지] 파이어베이스 회원가입 기능 추가하기   |
| 04. [회원가입/로그인 페이지] 파이어베이스 로그인 기능 추가하기 & 비밀번호 Reset 기능                                |
| 05. 쿠팡 Footer UI 생성하기  |
| 06. 쿠팡 Header UI 생성하기  |
| 07. Header 컴포넌트 스타일링하기   |
| 08. InnerHeader 컴포넌트 생성하기  |



CURRICULUM

01.

**대규모 이커머스  
서비스 프로젝트  
개발하기 -  
프론트유저단**

파트별 수강시간 14:42:57

|  |
|--|
| <b>CH04. 메인 페이지 생성하기</b>                           |
| 01.리덕스 Slice, Store 생성하기                           |
| 02.로그인 시 유저 데이터 리덕스 스토어에 담기                        |
| 03.[메인 페이지] Slide Carousel 생성하기                    |
| 04.[메인 페이지] Slide Carousel 스타일 생성하기                |
| 05.상품 생성 페이지 UI 생성하기                               |
| 06.상품 생성 페이지 스타일링하기                                |
| 07.상품 데이터 저장하기                                     |
| 08.useFetchCollection Hook를 이용한 데이터 가져오기           |
| 09.Product 데이터 Redux Store에 저장하기                   |
| 10.[메인페이지] 상품나열 필터 기능: Product Filter 컴포넌트 생성하기    |
| 11.[메인페이지] 상품나열 필터 기능: Filter Slice 생성하기           |
| 12.[메인페이지] 상품나열 필터 기능: Product Filter 컴포넌트 UI 생성하기 |
| 13.[메인페이지] 상품나열 필터 기능: Product Filter 컴포넌트 스타일링하기  |
| 14.[메인페이지] 메뉴: Product List 컴포넌트 생성하기              |
| 15.[메인페이지] 메뉴: Product List 컴포넌트 UI 생성하기           |
| 16.[메인페이지] 메뉴: Product List 컴포넌트 스타일링하기            |
| 17.[메인페이지] 개별 상품: Product Item 컴포넌트 생성하기           |
| 18.[메인페이지] 개별 상품: Product Item 컴포넌트 스타일링하기         |
| 19.[메인페이지] 개별 상품: 각 상품의 평점 가져오기                    |
| 20.[메인페이지] 상품 검색 기능 생성하기                           |
| 21.[메인페이지] 페이지 처리를 위한 Pagination 컴포넌트 생성하기         |
| 22.[메인페이지] Pagination UI 생성하기                      |



CURRICULUM

01.

**대규모 이커머스  
서비스 프로젝트  
개발하기 -  
프론트유저단**

파트별 수강시간 14:42:57

|   |
|---|
| <b>CH05. 상세 페이지, 장바구니 페이지, 주소 입력 페이지 등 생성하기</b>                         |
| 01. [상세페이지] 상품 상세 페이지 생성하기  |
| 02. [상세페이지] 상품 상세 페이지 UI 생성하기: 상품 데이터 나열(제품 정보 표시)                      |
| 03. [상세페이지] 상품평 UI 생성하기: 리뷰 보여주기  |
| 04. [상세페이지] 장바구니에 담기 기능 생성하기: 장바구니 데이터 관리(리덕스를 통해 장바구니에 있는 상품 데이터 업데이트) |
| 05. [장바구니 페이지] 장바구니 페이지 생성하기: 상품 삭제, 상품 개수 업데이트                         |
| 06. [장바구니 페이지] Cart Slice 생성하기  |
| 07. [장바구니 페이지] UI 생성하기  |
| 08. [장바구니 페이지] 스타일링하기   |
| 09. [주소 입력 페이지] 생성하기  |
| 10. [주소 입력 페이지] UI 생성하기   |
| 11. [주소 입력 페이지] 스타일링하기  |
| 12. [주문 페이지] 생성하기   |
| 13. [주문 페이지] 토스 결제 서비스 이용을 위한 Key 생성하기                                  |
| 14. [결제 페이지] 결제 기능 생성하기 & 결제가 완료된 후, 사용자의 장바구니를 비우기                     |
| 15. [결제 페이지] 결제 성공 요약 페이지 생성하기 (결제 정보)                                  |
| 16. [주문 목록 페이지] 생성하기  |
| 17. [주문 목록 페이지] UI 생성하기 (주문 번호, 주문 일자)                                  |
| 18. [주문 상세 정보 페이지] 생성하기: 상품 주문 상황에 대한 정보 표시(주문 상품명, 수량, 가격, 주문 상태 등)    |
| 19. [상품평 작성 페이지] 생성하기   |
| <b>CH06. 실력 UP 과제 01</b>  |
| 01. 과제 소개 - 구현 조건 및 와이어프레임을 바탕으로 다른 기능 추가 구현하며 채팅 앱 구현하기                |
| 02. 리엑트 앱 생성 및 전체 구조 생성하기   |
| 03. 파이어베이스 사용을 위한 설정하기  |
| 04. 로그인 페이지 생성하기  |
| 05. 로그인 컴포넌트 생성하기   |
| 06. 유저 데이터 데이터베이스에 저장   |
| 07. 사이드바 유저 나열하기  |
| 08. 유저 리스트 아이템 컴포넌트 생성하기  |
| 09. 채팅 입력 Form 생성하기   |
| 10. 채팅 메시지 데이터 가져오기   |
| 11. Topbar 생성하기   |
| 12. 메시지 보여주기  |

## CURRICULUM

## 02.

## Shop App “Admin” 기능 생성하기

파트별 수강시간 01:53:12

### CH01. Shop App “Admin” 기능 생성하기

01. Admin Navbar 생성하기

02. [관리자 모든 상품 페이지] 생성하기

03. [관리자 모든 상품 페이지] UI 생성하기(페이지네이션 포함)

04. [관리자 모든 상품 페이지] 상품 삭제 기능 생성하기

05. [관리자 모든 상품 페이지] 상품 수정 기능 생성하기

06. [관리자 주문 내역 페이지] 생성하기

07. [관리자 주문 내역 상세 정보 페이지] 생성하기

08. [관리자 주문 내역 상세 정보 페이지] ChangeOrderStatus 컴포넌트 생성하기: 주문 상태 업데이트 기능 포함

09. [관리자 대시보드 페이지] 생성하기: 전체 주문 내역 및 수익 확인을 위한 페이지

10. [관리자 대시보드 페이지] UI 생성하기

11. [관리자 대시보드 홈페이지] Chart 생성하기: 주문 상품 데이터 차트로 보여주기



## CURRICULUM

## 03.

## 개발 중인 프로덕트의 안정성 향상을 위한 TS 적용하기

파트별 수강시간 01:45:15

### CH01. 에러 방지를 위하여 모든 JS를 TS로 변경하기

01. 타입스크립트를 이용해서 리팩토링 하는 이유

02. 타입스크립트를 사용하기 위한 준비하기

03. Input Component 를 TS로 변경하기

04. AutoSignInCheckbox Component를 TS로 변경하기

05. Button Component를 TS로 변경하기

06. 로그인, 회원가입 페이지 타입스크립트로 변경하기

07. 리덕스 타입스크립트로 변경하기

08. 리덕스 Slice 타입스크립트로 변경하기

09. ProductFilter 컴포넌트 TS로 변경하기

10. 여러 컴포넌트들 타입스크립트로 변경하기(1)

11. 여러 컴포넌트들 타입스크립트로 변경하기(2)

12. Hooks를 TS로 변경하기

13. Util 파일들을 TS로 변경하기

14. Checkout 부분을 TS로 변경하기

15. 여러 페이지들 타입스크립트로 변경하기

16. Admin 부분을 TS로 변경하기

17. Vercel을 이용해서 애플리케이션 배포하기

## CURRICULUM

## 04.

## 개발 중인 프로덕트의 편의성 향상을 위한 접근성 알아보기

파트별 수강시간 02:05:03

### CH01. 접근성(Accessibility) 설명 및 사용하기

01. React에서 접근성이란
02. aria-role, onclick
03. tabIndex
04. img alt
05. onmouseover, onmouseout
06. image hash ref
07. 상호작용 가능하지 않은 태그에서 role 사용하기
08. aria-labelledby
09. 더 많은 aria 속성 살펴보기
10. aria-control

### CH02. 실력 UP 과제 02

01. 과제 소개 | K사 온보딩 과제 유사 | 제한 조건 내에서 JS 활용 심화 구현 하기
02. 과제 해설 영상 01\_앱의 기본 구조 생성하기
03. 과제 해설 영상 02\_SpreadSheet를 위한 데이터 생성하기
04. 과제 해설 영상 03\_10개의 셀을 하나의 Row Div로 감싸기
05. 과제 해설 영상 04\_Header 생성하기
06. 과제 해설 영상 05\_엑셀파일 다운로드 하기



CURRICULUM

05.

**개발중인  
프로덕트의  
생산성 향상을  
위한 테스트**

파트별 수강시간 03:45:17

|  |
|--|
| <b>CH01. 파트에서 다룰 내용 안내</b>   |
| 01.전체 담당 파트 개요 소개  |
| 02.part _ overview   |
| <b>CH02. 상황으로 이해하기</b>   |
| 01. 상황 1.<br>빈번한 요구사항 변경, 예외 상황 발생으로 인한 지속적 수정의 발생, 생산성이 하락한다                                |
| 02. 상황 2.<br>복잡한 비즈니스 요구사항에 따른 개발 난이도의 급격한 상승으로 인해 생산성이 하락한다                                 |
| <b>CH03. 문제 상황 해결방법</b>  |
| 01.요구사항 분석을 통한 생산성 향상  |
| 02.유닛테스트 도입으로 인한 생산성 향상  |
| <b>CH04. 리액트 테스트 패턴</b>  |
| 01. 상황1. 설정과 해지를 이용한 환경의 격리가 필요한 때   |
| 02. 상황2. UI 인터랙션 단위의 검증이 필요한 때   |
| 03. 상황3. 렌더링 검증이 필요한 때   |
| 04. 상황4. 모의 데이터를 이용한 검증이 필요한 때   |
| 05. 상황5. 외부 모듈의 Mocking을 이용한 검증이 필요한 때   |
| 06. 상황6. DOM 이벤트 발생을 통한 검증이 필요한 때  |
| 07. 상황7. 타이머 기반 비동기 동작 검증이 필요한 때   |
| 08. 상황8. 스냅샷 테스트가 필요한 때  |
| <b>CH05. Reat Testing Library 실사용 예시</b>   |
| 01. 상황9. 복잡해진 테스트 코드를 간결하게 할 때   |
| <b>CH06. 일반적으로 실수하는 테스트 패턴과 이에 대한 올바른 사용법</b>  |
| 01. 상황들에 맞는 테스트를 구현했지만, 빈번하게 실수하는 테스트 코드들이 존재한다  |
| <b>CH07. 실력 UP 과제</b>  |
| 01.과제 소개: 생산성 향상을 위한 요구사항 구체화 및 이를 기반으로 한 유닛테스트 시나리오 작성(요구사항 문서, Jest React Testing Library) |
| 02.과제 해설 영상  |



CURRICULUM

06.

서로 다른 목적  
조직간의 동기를  
맞추는 배포/형상  
관리

파트별 수강시간 03:18:31

|   |
|---|
| CH01. 파트에서 다룰 내용 안내   |
| 01. part overview   |
| CH02. 상황으로 이해하는 형상관리를 위한 GitHub / GitLab의 사용 목적과 방향성        |
| 01. 다수의 개발자가 동시에 작업하고 다른 브랜치에서 작업하는 상황                      |
| CH03. 다수와의 협업을 위한 형상관리                                      |
| 01. 상황1: rebase conflict 와 해결                               |
| 02. 상황1: rebase conflict 와 해결 실습                            |
| 03. 상황2: 개발 중에 임시 저장 및 revert 방법 이해                         |
| 04. 상황3: 여러가지 형상 관리 방법론을 바탕으로 한 형상 관리 전략                    |
| CH04. 상황으로 이해하는 배포 개념/목적                                    |
| 01. 다수의 개발자가 동시에 작업할 때 호환성을 검증이 필요한 상황                      |
| 02. 상황1. 여러가지 배포 방법론을 알고 적용해야 할 때                           |
| 03. 상황2. Vercel 활용 CICD 구축이 필요할 때                           |
| 04. 상황3. 여러가지 사용 가능한 배포 환경 이해                               |
| 05. 상황4. 구축한 CICD 활용 테스트 - 배포 프로세스 파이프라인 구축이 필요한 때          |
| 06. 상황4. 구축한 CICD 활용 테스트 - GitHub Action 이해                 |
| 07. 상황4. 구축한 CICD 활용 테스트 - GitHub Action을 이용한 CICD 파이프라인 구축 |
| 08. 상황4. 구축한 CICD 활용 테스트 - GitHub Action을 이용한 CICD 파이프라인 구축 |



CURRICULUM

07.

UX 개선을 위한  
효율적인 비동기  
데이터의 관리

파트별 수강시간 03:03:32

|  |
|--|
| <b>CH01. 파트에서 다룰 내용 안내</b>   |
| 01. part overview  |
| <b>CH02. 상황으로 이해하는 비동기 데이터 관리의 필요성</b>                               |
| 02. 비동기 로직의 응답 지연으로 인한 UX 저하가 우려될 때                                  |
| <b>CH03. 속도 향상을 위한 효율적인 비동기 처리</b>                                   |
| 01. 상황1. Data loading시에 사용될 수 있는 loading상태 관리                        |
| 02. 상황2. UX를 저하하는 'API 호출 동안 공백으로 보여질 때                              |
| 03. 상황3. 한 번에 많은 데이터를 불러와 API 호출시간이 길어질 때                            |
| 04. 상황4. 한 번에 많은 데이터를 불러와 API 호출시간이 길어질 때_실습                         |
| 05. 상황5. 초기 상품 목록을 매번 클라이언트에서 불러와 발생하는 API 비용 증가 및 웹사이트 성능 문제가 발생할 때 |
| <b>CH04. 실력 UP 과제</b>  |
| 01. 과제 소개: 다양한 방법의 비동기 로직 처리방법을 이용한 UX 최적화 구현                        |
| 02. 과제 해설 영상   |



CURRICULUM

08.

**프로덕트 성공을  
위한 노출을  
늘리는 검색엔진  
최적화**

파트별 수강시간 01:40:04

|   |
|---|
| <b>CH01. 파트에서 다룰 내용 안내</b>  |
| 01. part overview   |
| <b>CH02. 상황으로 이해하는 SEO 필요성(SEO의 개념)</b>                             |
| 01.유저 유입률이 낮아 프로젝트 성공이 걱정될 때<br>(SEO 개념, 각 구성요소의 역할에 대한 이해)         |
| <b>CH03. 사용자 유입을 위한 SEO 전략1. SNS 공유를 위해 구성하기</b>                    |
| 01. 상황1: 웹페이지를 SNS를 통해 공유하여 노출률 높이기(open graph element)             |
| <b>CH04. 사용자 유입을 위한 SEO 전략2. 효율적인 SEO 구축방법</b>                      |
| 01. 상황2: 효율적인 SEO를 구축하기 위해 고려해야 하는 상황들(Case를 통한 잘 구축된 웹페이지 정보 파악하기) |
| 02. 상황2: 효율적인 SEO를 구축하기 위해 고려해야 하는 상황들(Case를 통한 잘 구축된 웹페이지 정보 파악하기) |
| <b>CH05. 실력 UP 과제</b>   |
| 01.과제 소개: 실전 SEO 적용하기   |
| 02.과제 해설 영상   |



CURRICULUM

09.

개발 생산성을  
확보하는  
의존성을 배제한  
FE개발

파트별 수강시간 01:27:16

|   |
|---|
| CH01. 파트에서 다룰 내용 안내   |
| 01.part overview  |
| CH02. 상황으로 이해하는 의존성을 배제한 FE 개발                                    |
| 01.FE-BE 간의 커뮤니케이션, 이로 인한 개발 지연이 우려될 때                            |
| CH03. API Mocking을 이용한 백엔드 의존성 줄이기                                |
| 01.상황1: 개발을 나누어 하며 발생하는 Context switching에 따른 시간 소모와 낭비, 이를 방지하기  |
| 02.상황2: BE 아키텍처 변경에 따른 FE 수정이 많아지는 때, 이 문제를 효율적으로 다루기 위해 사전에 준비하기 |
| CH04. 실력 UP 과제  |
| 01.과제 소개: 의존성 주입을 활용한 백엔드 API 목킹 구현                               |
| 02.과제 해설 영상   |



CURRICULUM

10.

**운영 비용 관리를  
위한 배포 환경  
내재화**

파트별 수강시간 04:03:11

|   |
|---|
| <b>CH01. 파트에서 다룰 내용 안내</b>                            |
| 01. part overview                                     |
| <b>CH02. 상황으로 이해하는 개발/운영 비용과 배포의 연관성</b>              |
| 01. 트래픽 증가에 따라 발생 가능한 이슈가 예상될 때(배포 환경 내재화)            |
| <b>CH03. 프론트엔드 엔지니어 관점의 인프라스트럭처</b>                   |
| 01. 상황1: 개발 외적인 부분에서 발생하는 환경 구성 방법에 대한 안내 / 이해        |
| 02. 방법1: AWS 서비스 안내 - Amplify                         |
| 03. 방법2: AWS 서비스 안내 - EC2                             |
| 04. 방법3: AWS 서비스 안내 - S3                              |
| 05. 방법4: AWS 서비스 안내 - CloudFront                      |
| 06. 방법4: AWS 서비스 안내 - CodeBuild & Code Deploy         |
| 07. 방법5: AWS 서비스 안내 - AWS CloudWatch                  |
| 08. 방법6: CI/CD 구성을 위한 GitHub Actions 작성 가이드           |
| 09. 방법7: Docker 구성과 Registry                          |
| <b>CH04. 실력 UP 과제</b>                                 |
| 01. 과제 소개: AWS 리소스 생성하기(AWS services, GitHub Actions) |
| 02. 과제 해설 영상  |





CURRICULUM

11.

**개발 생산성  
향상을 위한  
고도화된 테스트  
전략 수립**

파트별 수강시간 01:35:18

|  |
|--|
| CH01. 파트에서 다룰 내용 안내  |
| 01.part overview   |
| CH02. 상황으로 이해하는 테스트 전략 필요성   |
| 01.프로덕트가 성장하며 전체를 보기 힘들어질 때  |
| CH03. 프로젝트 신뢰성을 높이는 테스트  |
| 01.상황1:<br>복잡한 요구사항에 대한 효율적인 개발시간 산정 및 반복되는 실수의 회피를 위한 TDD                     |
| 02.상황2:<br>프로덕트 신뢰성 향상을 위하여 Playwright을 활용한 E2E 테스트 자동화 구현하기                   |
| CH04. 실력 UP 과제   |
| 01.과제 소개: 프로젝트 신뢰성향상을 위한 요구사항 기반 E2E 테스트 환경 구성 및 테스트 스크립트 구현 (playwright,Jest) |
| 02.과제 해설 영상  |



CURRICULUM

12.

**개발생산성 및  
성능 개선을 위한  
퍼포먼스 최적화**

파트별 수강시간 02:35:56

|   |
|---|
| <b>CH01. 파트에서 다룰 내용 안내</b>                                  |
| 01. part overview   |
| <b>CH02. 상황으로 이해하는 퍼포먼스 최적화 필요성</b>                         |
| 01. 개발 중 퍼포먼스 저하가 느껴질 때                                     |
| <b>CH03. 최적화를 위한 리렌더링 개선</b>                                |
| 01. 리렌더링 발생 원인에 대한 근원적 이해 & 발생시 문제 상황                       |
| 02. 상황1: 성능 저하가 발생시에 해당 원인을 파악하는 방법 습득                      |
| 03. 상황2:<br>컴포넌트의 내 사용되는 값들의 레퍼런스 변경에 따른 불필요한 리렌더링이 발생하는 경우 |
| 04. 상황3: form을 사용시에 사용자 입력에 따른 렌더링 증가로 성능 감소가 보여질 때         |
| 05. 상황4: 컴포넌트의 복잡성 증가로 인한 리팩토링이 요구될 때                       |
| <b>CH04. 최적화를 위한 로딩 시간 개선</b>                               |
| 01. 상황5: 페이지 로드애 많은 시간이 걸려서 유저들이 자주 접근하지 않게 되었을 때           |
| 02. 해결1: 페이지 성능 측정  |
| 03. 해결2: bundle size가 커서 최적화 진행                             |
| <b>CH05. 최적화를 위한 순환참조 개선</b>                                |
| 01. 상황1: 순환 참조로 인한 무한로딩의 발생했을 때                             |
| <b>CH06. 실력 UP 과제</b>                                       |
| 01. 과제 소개: 문제 상황이 발생한 컴포넌트 최적화 진행하기                         |
| 02. 과제 해설 영상  |



CURRICULUM

13.

실제 제품을 운영  
중 프로덕션  
환경에서  
발생하는 여러  
문제 해결

파트별 수강시간 03:47:13

|  |
|--|
| <b>CH01. 파트에서 다룰 내용 안내</b>   |
| 01. part overview  |
| <b>CH02. Next.js 서버의 CPU 사용량 증가로 인한 성능 저하 개선</b>                     |
| 01. Next.js 서버 사이드 렌더링 기능이 필요한 상황                                    |
| 02. 상황1: 성능 저하의 원인 판단 및 해결(서버 인스턴스 추가, 코드 최적화)                       |
| 03. 상황2:<br>Next.js 렌더링 작업 최적화를 위한 서버 설정 Optimization(Standalone 배포) |
| <b>CH03. Build 시간의 증가로 인한 CI 파이프라이닝 문제 개선</b>                        |
| 01. 상황1: Next.js의 Build Cache를 사용하여 빌드 시간 단축이 필요한 때                  |
| 02. 상황2: AWS의 CodeBuild에서 빌드 캐시를 사용하여 빌드 시간 단축이 필요한 때                |
| <b>CH04. Error Handling</b>  |
| 01. 상황1: Runtime 에서의 Error 발생 상황 케이스                                 |
| 02. 상황2: Error Boundary 를 활용한 fallback UI 구성 안내                      |
| 03. 상황3: 원활한 유지보수를 위한 Runtime Error 수집을 위한 Sentry 구축                 |
| 04. 상황4: 효과적 유지보수를 위한 Sentry 부가 사용법 안내                               |
| <b>CH05. 의존성 그래프를 이용한 코드 로직의 최적화</b>                                 |
| 01. 상황1: 의존성 파악을 통한 불필요하거나 비효율적 코드 파악이 필요한 때                         |
| 02. 상황2: 프로젝트 내 불필요한 모듈을 찾아, 해당 모듈을 제거 혹은 사용함으로써 프로젝트의 복잡도를 낮춰야 할 때  |
| 03. 상황3: 순환 참조의 발생으로 인한 운영시의 문제를 회피해야 할 때                            |
| 04. 상황4: project wide한 큰 그림을 그려야 하는 때(Summary)                       |
| <b>CH06. 서로다른 개발자들 간의 commit 시에 발생하는 conflict 해결</b>                 |
| 01. 상황 소개: Git conflict 발생 상황에 대한 케이스별 안내                            |
| 02. 문제 방지: Conflict발생 상황을 최소화 시킬 수 있는 실천적인 방법                        |
| 03. 잘못된 rebase로 인한 문제 복구 - reflog 이용                                 |
| <b>CH07. 개발 Feature에 대한 검증 및 rollback 전략 수립</b>                      |
| 01. A/B테스트 도입을 통한 개발 feature의 배포 여부 결정 수립 과정                         |
| 02. Google Analytics를 이용한 A/B 테스트 결과 수집 및 분석                         |

## CURRICULUM

# 14.

## (오프닝) 프로덕트의 생명주기

파트별 수강시간 01:13:06

|                                      |
|--------------------------------------|
| <b>CH01. 커리큘럼 진행 방식</b>              |
| 01. 소개                               |
| 02. 전체 강의 개요                         |
| 03. 커리큘럼 진행 방식 알아보기                  |
| 04. 프로젝트 생명주기 개요                     |
| <b>CH02. 프로젝트가 성장하면 어떤 일이 발생하나요?</b> |
| 01. 개요                               |
| 02. 여러 관점의 성장 살펴보기                   |
| 03. 결론                               |
| <b>CH03. 아키텍처를 왜 만들어야 하나요?</b>       |
| 01. 개요                               |
| 02. 결론                               |
| <b>CH04. 프로젝트 성장을 위한 로드맵</b>         |
| 01. 개요                               |
| 02. 결론                               |



CURRICULUM

15.

**서비스가  
성장하며 단일  
프로덕트에서  
멀티 도메인  
프로덕트로  
가기위해**

파트별 수강시간 03:46:01

|  |
|--|
| <b>CH01. 이 파트에서 다룰 내용</b>                                |
| 01.이 파트에서 다룰 내용  |
| <b>CH02. 1상황: 상황을 정확히 파악하는 메타인지가 필요할 때</b>               |
| 01.강의 개요   |
| 02.비즈니스와 생산성   |
| 03.단일 프로덕트에서 발생하는 문제를 진단하기                               |
| 04.문제 파악을 통한 상황 진단하기                                     |
| 05.메타인지에 도움을 주는 여러 요소                                    |
| <b>CH03. 2상황: 적절한 비즈니스 대응을 위한 아키텍처 그려보기</b>              |
| 01.강의 개요   |
| 02.아키텍처를 그리기 위한 도구 선정                                    |
| 03. 아키텍처를 그려보기   |
| 04. 프론트 기술리더의 역할_솔루션 아키텍트                                |
| <b>CH04. 3상황: 프로덕트 코드의 병목 상황 개선하기</b>                    |
| 01.강의 개요   |
| 02.여러 영역의 병목 상황 진단하기                                     |
| 03.코드 병목 지점 파악하기   |
| 04.다양한 컴포넌트를 나누는 패턴 알아보기                                 |
| 05.비즈니스 로직 단위의 코드 병목 개선하기                                |
| 06.상태 단위의 코드 병목 개선하기                                     |
| <b>CH05. 4상황: API 영역의 문제 상황 개선하기</b>                     |
| 01.강의 개요   |
| 02.서비스 성장에 따른 여러 마이크로 서비스 API를 대응하기 위한 상황 진단 및 해결하기      |
| 03.클라이언트와 서버간 1.1인 상황 진단 및 해결하기                          |
| 04.클라이언트와 서버간 1.n인 상황 진단 및 해결하기                          |
| 05.빠르게 개발되는 과정에서 서버 API 스펙에 빠진 부분을 방어하기<br>(런타임 벨리데이션구축) |
| 06.클라이언트와 서버간 밀접한 업무 상황에서 효율적 개발을 위한 방법 탐색하기             |
| <b>CH06. 5상황: 중복된 기술이 산재하는 문제 상황 해결하기</b>                |
| 01.강의 개요   |
| 02.레거시 코드 내 중복 기술 파악하기                                   |
| 03.비즈니스 로직을 최대한으로 보존하며 분리하기                              |
| 04.점진적으로 분리하기  |
| 05.하나의 기술로정리하기   |
| 06.서버의 상태와 클라이언트 상태를 분리하기                                |
| <b>CH07. 6상황. 그 외 부족한 여러상황을 둘러보고 보완하기</b>                |
| 01.강의 개요   |
| 02.로그와 산재되어 있는 상황에서 중앙화를 통해 로직 개선하기                      |
| 03.중앙화된 웹앱 인터페이스 중앙화를 통한 로직 개선하기                         |



CURRICULUM

16.

늘어난  
개발자와의  
협업을 위하여  
멀티 도메인을  
포함하기 위한  
모노레포로의  
전환

파트별 수강시간 03:37:42

|   |
|---|
| <b>CH01. 이 파트에서 다룰 내용</b>               |
| 01. 이 파트에서 다룰 내용                        |
| <b>CH02. 1상황: 필요한 요구사항의 분석</b>          |
| 01. 멀티레포에서 모노레포로 전환할 상황인지 판단하기          |
| 02. 모노레포 전환의 이점과 단점 살펴보기                |
| 03. 모노레포 전환 후 아키텍처 그리기                  |
| 04. 모노레포 전환 후 아키텍처 변경점 확인하기             |
| 05. 모노레포의 전환에 따른 코드 변경점 확인하기            |
| 06. 모노레포 전환에 따른 인프라 스트럭처 구성 변경점 확인하기    |
| 07. 모노레포 전환에 따른 팀 문화 변경점 확인하기           |
| <b>CH03. 2상황: 모듈 기반의 개발 방식에 대한 이해</b>   |
| 01. 모듈 기반으로 생각해야 하는 이유                  |
| 02. 현재 비즈니스에 적합한 모듈 단위의 레이어 구상하기        |
| 03. 분리되는 코드를 운영하기 위한 프로세스 구축            |
| <b>CH04. 3상황: 모노레포를 적용하기 위한 여정</b>      |
| 01. 구축 상황에 맞는 모노레포의 기술 선택하기             |
| 02. 보일러플레이트 기반 모노레포 구축 및 적합한 보일러플레이트 선택 |
| 03. 모노레포 구축 후 핵심 코드 및 구조 돌아보기           |
| <b>CH05. 4상황 신규 프로젝트를 구축해야 하는 상황</b>    |
| 01. 페이지 요구사항 구현을 위한 상황 진단하기             |
| 02. 공통으로 사용할 수 있는 모듈을 이용한 개발환경 구축하기     |
| 03. 페이지 요구사항을 구현 및 인프라 구축 알아보기          |
| <b>CH06. 5상황: 공통의 라이브러리가 많아지는 상황</b>    |
| 01. 공통 코드의 선형적 증가에 따른 계층의 필요성 대두        |
| 02. DX 개선의 필요성 대두                       |
| 03. Module Federation                   |



CURRICULUM

17.

**디자이너와의  
원활한 협업 및  
개발 생산성을  
위해 비대해진  
컴포넌트,  
디자인시스템으  
로 전환하기**

파트별 수강시간 01:38:27

|   |
|---|
| <b>CH01. 이 파트에서 다룰 내용</b>               |
| 01. 이 파트에서 다룰 내용                        |
| <b>CH02. 1상황: 컴포넌트 라이브러리, 디자인시스템으로</b>  |
| 01. 컴포넌트 라이브러리와 디자인 시스템                 |
| 02. 디자인 시스템으로 가기위해                      |
| <b>CH03. 2상황: 디자인시스템 구축</b>             |
| 01. 최소 수준의 아키텍처 설정                      |
| 02. 파운데이션                               |
| 03. 컴포넌트                                |
| 04. 디자인 언어                              |
| 05. 자동화된 디자인시스템 개발 및 운영 프로세스            |
| 06. 디자인시스템의 안정성을 위한 테스트                 |
| <b>CH04. 3상황: 더 효율적인 디자인 시스템 사용 만들기</b> |
| 01. 필드 테스트로 실사용 적합성 판단하기                |
| 02. 피그마 플러그인으로 디자이너향 디자인시스템 편의성 확장하기    |
| 03. 디자인시스템을 더 체계화 하는 디자인 가이드라인 제작하기     |
| 04. 복잡한 형태의 컴포넌트 지원하기                   |



CURRICULUM

18.

조직의 규모  
성장에 따른  
복잡한  
R&R에서의  
프로덕트  
운영하기

파트별 수강시간 00:00:00

|  |
|--|
| CH01. 이 파트에서 다룰 내용                     |
| 01. 이 파트에서 다룰 내용                       |
| CH02. 상황1: 한 페이지에 복잡한 R&R이 있는 상황       |
| 01. 강의 개요                              |
| 02. 여러 케이스의 R&R 분리                     |
| 03. R&R을 나눌 수 있는 여러 기술 알아보기            |
| 04. 정리                                 |
| CH03. 상황2: 멀티 도메인의 프로덕트를 합쳐야 하는 상황     |
| 01. 강의 개요                              |
| 02. 예시로 상황 파악하기                        |
| 03. 합치기 위한 설계                          |
| 04. 정리                                 |
| CH04. 상황3: 모노레포에서 여러개의 모노레포로 분리해야하는 상황 |
| 01. 강의 개요                              |
| 02. 분리 상황 살펴보기                         |
| 03. 기술적으로 고민하기                         |
| CH05. 끝으로                              |
| 01. 정리                                 |





CURRICULUM

19.

**효율적인 업무를  
위한 개발환경  
성장**

파트별 수강시간 00:42:52

|  |
|--|
| <b>CH01. 이 파트에서 다룰 내용</b>                |
| 01.이 파트에서 다룰 내용                          |
| <b>CH02. 상황1: 조직 관점에서 문제 정의하기</b>        |
| 01.강의 개요                                 |
| 02.문제 파악과 정의                             |
| 03.문화 만들기                                |
| <b>CH03. 상황2: 업무 프로세스의 병목 파악하기</b>       |
| 01.강의 개요                                 |
| 02.협업과 프로세스 차이 이해                        |
| 03.병목 개선                                 |
| <b>CH04. 상황3: 부서 내외의 커뮤니케이션에서의 병목 해결</b> |
| 01.강의 개요                                 |
| 02.여러 조직과 커뮤니케이션                         |
| 03.조직 리더의 커뮤니케이션 역할                      |
| <b>CH05. 끝으로</b>                         |
| 정리                                       |



---

## CURRICULUM

### [Special].

우아한형제들  
개발팀장 X  
원티드 PO :  
PO가 작성한  
PRD 기반으로  
협업하기

파트별 수강시간 02:03:56

---

- |                        |
|------------------------|
| 01. 스페셜 파트 소개          |
| 02. PRD 작성하기           |
| 03. 피그잼을 활용한 플로우차트 실습  |
| 04. 피그마를 활용한 와이어프레임 실습 |
| 05. 기획자와 개발자의 트러블 슈팅   |

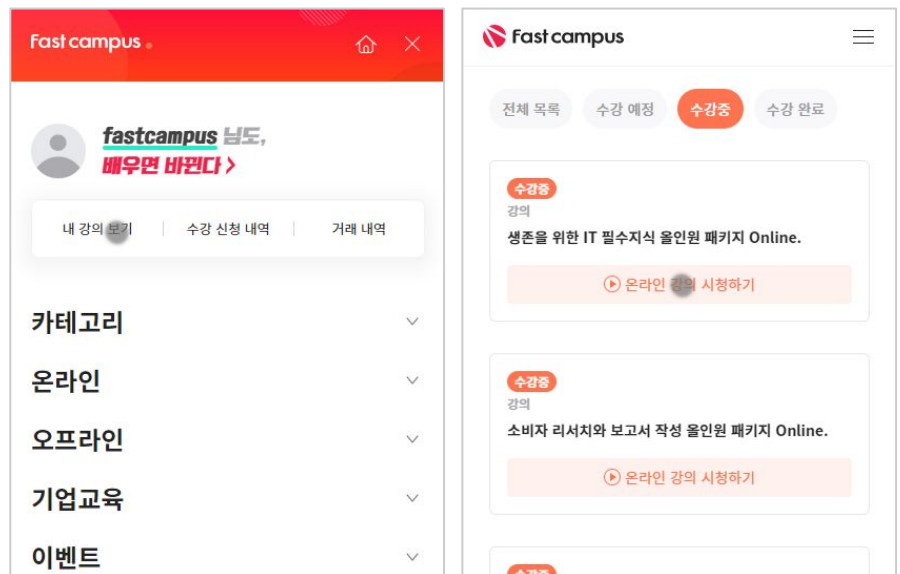


## 주의 사항

- 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.