## 초격차 패키지 :

# 시그니처 백엔드 Path

Last update : 2023. 08. 30

Signatur



Copyright **FAST CAMPUS Corp.**
All Rights Reserved.

Fast campus

# 기본 정보

Signature

- **강의 정보**
  a. 상세 페이지 : https://fastcampus.co.kr/dev_online_besignature
  b. 강의장 : 온라인 강의 | 데스크탑, 노트북, 모바일 등
  c. 수강기간 : 평생 소장, 무제한 반복 수강
  d. 강의 시간 : 194시간 53분
  e. 문의 : 고객센터

- **패스트캠퍼스 온라인 강의 특징**
  a. **나만의 속도로** : 낮이나 새벽이나 내가 원하는 시간대에 나의 스케쥴대로 수강
  b. **원하는 곳 어디서나** : 시간을 쪼개 먼 거리를 오가며 오프라인 강의장을 찾을 필요 없이 어디서나 수강
  c. **무제한 반복 학습** : 무엇이든 반복적으로 학습해야 내것이 되기에 이해가 안가는 구간 몇 번이고 재생

- **주의 사항**
  a. 상황에 따라 사전 공지 없이 할인이 조기 마감 되거나 연장될 수 있습니다.
  b. 패스트캠퍼스의 모든 온라인 강의는 아이디 공유를 금지하고 있으며
  1개의 아이디로 여러 명이 수강하실 수 없습니다.
  c. 별도의 주의사항은 각 강의 상세페이지에서 확인하실 수 있습니다.

- **수강 방법**
  a. 패스트캠퍼스는 크롬 브라우저에 최적화 되어 있습니다.
  b. 로그인 후, [ 내 강의 보기 클릭 ] 수강중 클릭 | 온라인 강의 시청하기
  c. 사전 예약 판매 중인 강의의 경우 1차 공개일정에 맞춰 '온라인 강의 시청하기'가 활성화 됩니다.

Fast campus

Signature

# 강의 정보

## 강의목표

- 이론과 실습으로 끝내는 강의가 아닌, 백엔드 개발자가 실무에서 거치는 프로세스대로 프로젝트 단계별로 경험해봅니다.
- IT 대기업이 채용 과정에서 우대하는 33가지 기술 스택을 다뤄봅니다.
- Java 코딩테스트 부터, 자소서 / 기술면접 / 인성면접까지 취업 과정을 한 번에 다뤄봅니다.

## 강의요약

- 요구사항 분석, 테이블 설계, DB 구축, 기능 구현, 배포 및 성능 개선까지 백엔드 실무 프로세스를 경험해 봅니다.
- 코딩 테스트의 핵심 알고리즘 유형 9가지 + 최종 모의고사 3회를 진행합니다.
- 자기소개서 작성부터 채용 프로세스별 준비 내용, 면접 전략, 백엔드 개발자로서 성장하는 전략까지 다뤄봅니다.
- 온라인 강의를 더욱 몰입하여 수강할 수 있도록
  질의응답/지식나눔/코드리뷰/스터디 등 다채롭게 활용해보세요.
  (디스코드 커뮤니티 운영 기간은 2023년 2월 27일 ~ 2026년 2월 27일 입니다.)

Fast campus

Signature

# 전체 강의 구성

| 주요 학습 툴 | 코스 | 강의 목표 |
|---|---|---|
| Java | 01. 객체지향 프로그래밍 with. Java | 생각 - 표현 - 코딩 학습 방식으로 입문자가 스스로 코딩할 수 있는 능력을 쌓고 객체지향 기반 Java를 익힙니다. |
| Java | 02. 실무 자바 프로그래밍 with. API | 다양한 외부 API를 활용한 5가지 웹 개발 프로젝트로 Java 프로그래밍 역량을 기릅니다. |
| Spring Boot, MySQL | 03. 웹 개발 입문과 데이터베이스 | 웹 개론과 데이터베이스를 학습하여 게시판 프로젝트를 통해 CRUD 와 REST API를 개발합니다. |
| Spring Boot, Spring JPA, Spring Security | 04. 프로젝트로 시작하는 웹 서비스 개발 | 배달 플랫폼 프로젝트를 통해 백엔드 개발 실무 프로세스를 경험합니다. 사용자용 배달 플랫폼과 파트너용 어드민을 개발합니다. |
| Kotlin, Spring Cloud | 05. 최신/심화 백엔드 웹 서비스 | 개발한 배달 플랫폼에 Kotlin 도입하며 Java 와 혼용하거나 Kotlin으로 리팩토링 합니다. 또한 MSA 와 모니터링까지 경험합니다. |
| Spring Batch | 06. 대용량 데이터를 다루는 배치 처리 | 백엔드에서의 배치 작업을 학습하고, Spring Batch의 내부 원리부터 프로젝트 적용까지 진행합니다. |
| Spring Webflux, Redis | 07. 대용량 트래픽 처리를 위한 백엔드 심화 | Redis 와 Spring Webflux를 학습하고 대용량 트래픽 처리를 위한 대기열 시스템을 구축합니다. |
| Docker, Jenkins, AWS | 08. 백엔드 개발자를 위한 데브옵스 입문 | 백엔드 아키텍처와 컨테이너를 학습하여 DevOps 기반의 인프라 및 CI/CD 파이프라인을 구축하고 배포하여 운영합니다. |
| IntelliJ, Git, Github | 부록 1. 신입 개발자를 위한 Start Kit | 백엔드 개발자라면 꼭 알아야 하는 툴, IntelliJ 와 Git/Github 사용법을 익히고, To-Do List 프로젝트로 실습합니다. |
| - | 부록 2. Java 알고리즘 코딩테스트 | 9개 유형의 알고리즘을 탄탄하게 학습하고, IT 대기업 출제 경향을 분석한 최종 모의고사로 코딩테스트를 준비합니다. |
| - | 부록 3. 백엔드 개발자 해외 취업 A-Z | 국내에도 적용될 수 있는 개발자 해외 취업 프로세스를 이해하고, 자소서, 기술 면접, 인성 면접 등 단계별로 준비합니다. |
| - | 부록 4. IT 대기업 이직을 위한 면접 팁 | IT 대기업이 공통적으로 선호하는 인재상을 알고 면접 전략, 경력 코딩테스트 대비 노하우 등을 전수합니다. |

Fast campus

Signature

# 강사 정보

| | | |
|---|---|---|
| **박매일** | **과목** | - Course 1. 입문자를 위한 객체지향 프로그래밍 With. Java<br>- Course 2. 실무 자바 프로그래밍 With. API |
| | **약력** | - 한양대학교 ERICA, 조선대학교, 광주인공지능사관학교, 소프트웨어마이스터고 외 다수 교육기관 강의<br>- 한국스마트미디어학회, 한국전력공사,스마트인재개발원 외 다수 기업 강의<br>- 인프런: 'Java TPC (생각하고, 표현하고, 코딩하고)' 등 |
| **예상국** | **과목** | - Course 3. 웹 개발 인문과 데이터베이스<br>- Course 4. 웹 서비스 개발 실전<br>- Course 5. 최신/심화 웹 개발 실전 |
| | **약력** | - 현) 레몬트리 Server Lead<br>- 전) 뱅크샐러드 Back End Developer<br>- 전) 카카오페이 Back End Developer<br>- 전) 카카오 Back End Developer |
| **김민수** | **과목** | - Course 6. 대용량 데이터를 다루는 배치 처리<br>- 부록 5. 백엔드 개발자의 N가지 성장 전략 |
| | **약력** | - 현) 카카오뱅크 Back End Developer<br>- 전) 스타트업 기업들<br><br>- [발표]<br>- if kakao 2022 '알림 서비스로 시작하는 서버 개발'<br>- ahea 2019 'Spring WebFlux는 어떻게 적은 리소스로 많은 트래픽을 감당할까?' |
| **이형구** | **과목** | - Course 7. 대용량 트래픽 처리를 위한 백엔드 심화 |
| | **약력** | - 현) 카카오 모빌리티 Software Engineer<br>- 전) 숨고 등<br><br>- [발표]<br>- if kakao 2022 'JVM Warm up' |

Fast campus

Signature

# 강사 정보

| | | |
|---|---|---|
| **이성미** | **과목** | - Course 8. 백엔드 개발자를 위한 DevOps 입문 - 이론 담당 |
| | **약력** | - 현) 베스핀 글로벌<br>- 전) HPE 등<br><br>- [강의]<br>- 사내교육: Docker, Kubernetes,EKS<br>- 기업강의: 삼성 SDS PaaS 과정, SK 하이닉스 등 |
| **양재현** | **과목** | - Course 8. 백엔드 개발자를 위한 DevOps 입문 - 실습 담당 |
| | **약력** | - 현) 베스핀 글로벌<br>- 전) 신세계아이앤씨, 서울신용평가정보 |
| **김하은** | **과목** | - 부록 1. 신입 개발자를 위한 Start Kit |
| | **약력** | - 현) 카카오뱅크 Software Development Engineer<br>- 전) EST software Development Engineer<br>- 전) Google Code in Mentor |
| **류호석** | **과목** | - 부록 2. Java 코딩 테스트 |
| | **약력** | - [출제 및 수상 경력]<br>- 삼성 SDS/삼성 SW/tkatjd SCPC/LG사내 대회, 코딩테스트 문제 출제 및 출제 조교<br>- 2014-16 ACM-ICPC 대전 지역 수상<br>- 외 다수의 수상 경력 |
| **문혜림** | **과목** | - 부록 3. 백엔드 개발자 해외 취업 A-Z |
| | **약력** | - 현) 카카오<br>- 전) Amazon<br>- 전) SAP |
| **개발자 K 강사님** | **과목** | - 부록 4. IT 대기업 이직을 위한 면접 팁 |
| | **약력** | - 현) 네카라 경력 공채 합격<br>- 전) 네카라 신입 공채 합격 |

Fast campus

# 커리큘럼

CURRICULUM

## [Course 1].

## Part 1. 생각, 표현,코딩하기란

파트별 수강시간 00:37:04

| Ch 01. 생각,표현,코딩하기란 |
| --- |
| 01. [생각하기] 사고개념정리 |
| 02. [표현하기] 메모리 이해하기 |
| 03. [코딩하기] 구현하기 |

Fast campus

# 커리큘럼

CURRICULUM

## [Course 1].

## Part 2. 자바 기초 다지기

파트별 수강시간 27:34:08

| |
|---|
| **Ch 01. 자바 개발환경 구축하기** |
| 01. 자바 개발환경(플랫폼)의 이해 |
| 02. 자바 개발도구 설치(JDK) |
| 03. IntelliJ 개발환경 구축 |
| **Ch 02. 첫 자바 프로그래밍 도전하기** |
| 01. JavaSE 프로젝트 만들기 |
| 02. JavaSE 프로젝트 구조 살펴보기(1) |
| 03. JavaSE 프로젝트 구조 살펴보기(2) |
| 04. JavaSE 프로젝트 구조 살펴보기(2)(실습) |
| 05. JVM과 자바의 구동방식 |
| **Ch 03. 이것만 알자! 프로그래밍의 3대요소** |
| 01. 변수(V),자료형(D),할당(A)이란 |
| 02. 변수(V),자료형(D),할당(A)이란(실습) |
| 03. 변수선언과 할당 및 초기화 |
| 04. 변수선언과 할당 및 초기화(실습) |
| 05. 변수와 심볼테이블(symbol table) |
| **Ch 04. 자료형(DataType)을 이해하라** |
| 01. 자료형의 종류 알아보기 |
| 02. 자료형의 종류 알아보기(실습) |
| 03. 기본자료형과 문자자료형 |
| 04. 기본자료형과 수의 표현(실습) |
| 05. 문자자료형(char)의 이해 |
| 06. 문자자료형(char)의 이해(실습) |
| 07. 자료형과 형 변환(Casting) |
| **Ch 05. 객체와 클래스의 등장** |
| 01. 새로운자료형이 필요한 이유 |
| 02. 새로운 자료형 만들기 |
| 03. 새로운 자료형 만들기(실습) |
| 04. 새로운 자료형 사용하기 |
| 05. 객체를 생성하는 방법(실습) |

Fast campus

# 커리큘럼

CURRICULUM

## [Course 1].

## Part 2. 자바 기초 다지기

파트별 수강시간 27:34:08

| |
|---|
| **Ch 06. 데이터의 이동 배열을 이해하라** |
| 01. 데이터를 이동하는 배열 소개 |
| 02. 데이터를 이동하는 배열(array) |
| 03. 배열을 사용하는 방법 |
| 04. 배열을 사용하는 방법(실습) |
| 05. 배열을 초기화하는 방법(실습) |
| 06. 기본배열과 객체배열(실습) |
| 07. 기본배열과 객체배열(실습2) |
| 08. 이미지를 표현하는 2차원 배열 |
| 09. 다차원 배열의 활용(실습) |
| **Ch 07. 논리적인 사고력 키우기 훈련** |
| 01. 연산자와 명령문을 이해하라 |
| 02. 증가,감소,삼항연산자를 이해하라 |
| 03. 산술,대입연산자를 이해하라 |
| 04. 관계, 논리연산자를 이해하라 |
| 05. 논리적인 사고력 if |
| 06. 논리적인 사고력 switch- case |
| 07. 논리적인 사고력 for |
| 08. 논리적인 사고력 while |
| 09. 논리적인 사고력 이중 for |
| 10. 논리적인 사고력 break,continue |
| **Ch 08. 메서드와 변수의 관계를 이해하라** |
| 01. 메서드란(Method)무엇인가-injeekim의 iMac |
| 02. 메서드는 변수다 |
| 03. 매개변수 전달기법(parameter passing) |
| 04. 메서드 오버로딩(Overloading) |
| 05. 우리가 사용하게 될 메서드의 형태 |

Fast campus

# 커리큘럼

CURRICULUM

## [Course 1].

## Part 3. 객체지향 프로그래밍 시작하기

파트별 수강시간 11:18:03

| |
|---|
| **Ch 01. 클래스로 객체 모델링하기** |
| 01. 사용자 정의 자료형의 이해 |
| 02. 클래스란 무엇인가 |
| 03. 객체생과정의 이해 |
| 04. 생성자메서드와 this |
| **Ch 02. 우리가 만들어보는 클래스의 종류** |
| 01. 모델(Model)이란 무엇인가 |
| 02. DTO, VO 클래스가 왜 필요할까 |
| 03. DAO 클래스가 왜 필요할까 |
| 04. Utility 클래스가 왜 필요할까 |
| **Ch 03. 객체를 접근하는 권한 이해하기** |
| 01. 접근권한(Access Modifier)이란 |
| 02. 패키지(package)란 무엇인가 |
| 03. 클래스를 접근하는 이름 이해하기 |
| 04. 클래스를 접근하는 방법 이해하기 |
| **Ch 04. 잘 설계된 VO 클래스(객체)** |
| 01. 정보은닉이란? |
| 02. setter, getter 메서드 만들기 |
| 03. 생성자를 이용한 초기화 |
| 04. toString() 메서드로 객체값 출력하기 |
| 05. 잘 설계된 VO 클래스 설계해보기 |
| **Ch 05. 배열과 클래스의 관계** |
| 01. 동일한 구조와 이질적인 구조 |
| 02. 기본배열과 객체배열의 관계 |
| Ch 06. static과 JVM 메모리 모델 이해하기 |
| 01. static과 메모리의 관계 |
| 02. static과 none static 멤버들의 접근 방법 |
| 03. static과 none static 멤버들의 접근 방법(2) |
| 04. JVM이 사용하는 메모리 영역 |
| 05. 객체생성과 static과의 관계 |
| 06. class,object, instance의 상호관계 |

CURRICULUM

# [Course 1].

## Part 4. 객체지향 프로그래밍 설계하기

파트별 수강시간 09:23:33

| |
|---|
| **Ch 01. 자바에서 상속이란** |
| 01. 상속의 컨셉 |
| 02. 수직적 구조와 수평적 구조의 설계 방법 |
| 03. 클래스를 계층화 하여 사용할 때 장점 |
| 04. 메모리를 통한 상속(extends)의 이해 |
| 05. 메모리를 통한 상속(extends)의 이해(실습) |
| 06. 상속관계에서 객체생성 |
| 07. 상속관계에서 객체 초기화 |
| **Ch 02. 상속관계에서 객체생성 및 Override** |
| 01. 동작측면에서 클래스 설계하기 |
| 02. 상속관계에서 객체생성 방법 |
| 03. 상속 체이닝과 super |
| 04. 메서드의 재정의(Override)란 |
| **Ch 03. 객체 형변환(Object Casting)** |
| 01. 부모와 자식 간에 형 변환이 된다 |
| 02. Upcasting 과 Downcasting |
| **Ch 04. 다형성(Polymorphism)과 활용** |
| 01. 다형성이란 무엇인가 |
| 02. 다형성 이론을 설명하기 위한 전제조건 |
| 03. 다형성 활용(다형성 인수) |
| 04. 다형성 활용(다형성 배열) |
| **Ch 05. 추상클래스와 인터페이스의 등장** |
| 01. 다형성을 보장한다는 의미는 무엇인가 |
| 02. 추상클래스와 다형성 |
| 03. 인터페이스의 등장 |
| 04. 인터페이스와 다형성 |
| 05. 인터페이스와 다중상속 |
| 06. 추상클래스와 인터페이스 비교 |
| **Ch 06. 자바 최상위 클래스 Object** |
| 01. Object클래스를 이용하여 객체 생성하기 |
| 02. Object클래스를 활용한 다형성 적용 |
| 03. Object클래스의 toString() 메서드 |

Fast campus

커리큘럼

CURRICULUM

# [Course 2].

# Part 1. 자바 API 활용 및 API 만들기

파트별 수강시간 07:01:16

| |
|---|
| **Ch 01. 우리가 사용하게 될 클래스들(APIs)** |
| 01. 직접 만들어서 사용하는 class들 |
| 02. Java에서 제공해 주는 class들 |
| 03. 다운 받아서 사용하는 class들 |
| **Ch 02. 내가 만들어본 최초의 API** |
| 01. 배열처럼 동작하는 API만들어 보기 |
| 02. IntArray 클래스를 만들어 사용하기 |
| 03. BookArray 클래스를 만들어 사용하기 |
| 04. ObjectArray 클래스를 만들어 사용하기 |
| 05. ArrayList 클래스 사용하기 |
| **Ch 03. 자바 String 클래스 사용하기** |
| 01. 문자열 생성방법 2가지 |
| 02. String클래스에서 제공하는 Method 활용하기 |
| 03. 자바에서 문자열을 비교하는 방법 |
| 04. 자바에서 문자열을 분리하는 방법 |
| **Ch 04. 인터페이스 기반의 프로그래밍** |
| 01. 인터페이스 기반의 프로그래밍이란 |
| 02. 데이터베이스 연결 Driver란 무엇인가 |
| 03. 인터페이스를 이용하여 데이터베이스 접속 |

CURRICULUM

# [Course 2].

# Part 2. Collection Framework API 이해 및 활용

파트별 수강시간 05:06:11

| Ch 01. 모아, 모아 컬랙션(Collection) API |
|---|
| 01. Wrapper 클래스란 |
| 02. 숫자와 문자열의 상호 변환 |
| 03. Collection Framewok API란 |
| 04. 순서가 있고 중복 가능한 List API |
| 05. 순서가 없고 중복 불가능한 Set API |
| 06. Key-Value로 관리하는 Map API |
| **Ch 02. 자바 제네릭(Generic) 이란** |
| 01. 왜 제네릭을 사용해야 하는가? |
| 02. 자바 제네릭 타입이란 |
| 03. 제네릭 멀티 타입 파라미터 |
| 04. 제네릭 제한된 타입 파라미터 |
| **Ch 03. 람다와 스트림 API 활용하기** |
| 01. 함수형 인터페이스(Functional Interface) |
| 02. 함수형 인터페이스 메서드 참조 |
| 03. 람다식 이란 무엇인가 |
| 04. 람다식의 사용방법 |
| 05. Stream API의 이해 |
| 06. Stream API의 활용 |

CURRICULUM

# [Course 2].

# Part 3. API를 활용하기

파트별 수강시간 01:47:25

| Ch 01. Others API를 활용하기 |
|---|
| 01. JSON, XML Data Format 이해하기 |
| 02. Gson API 활용하기 |
| 03. Jsoup API 활용하기 |
| 04. URLConnection 네트워킹 하기 |
| 05. Open API활용 날씨 정보를 가져오기 |

CURRICULUM

## [Course 2].

## Part 4. 자바 리펙토링 및 미니프로젝트

파트별 수강시간 03:01:53

| Ch 01. 실전 API 활용하기 사전지식 |
|---|
| 01. Excel API(POI) 활용하기 |
| 02. Excel에 회원정보 저장하기 |
| 03. Pdf API(iText) 활용하기 |
| **Ch 02. 실전API활용하기 (1)** |
| 01. 애완동물 진료관리 애플리케이션(MVC) |
| **Ch 03. 실전API활용하기 (2)** |
| 01. 크롤링하여 PDF에 책 목록 만들기 |
| **Ch 04. 실전API활용하기 (3)** |
| 01. 카카오 REST Open API 활용하기 |

CURRICULUM

## [Course 2].

## Part 5. 실전 프로젝트 과제

파트별 수강시간 00:29:27

| Ch 01. 실전프로젝트 과제 |
|---|
| 01. 이력서 자동생성 프로그램 요구사항 |
| 02. 이력서 자동생성 프로그램 기능정의 및 설계 |
| 03. 나만의 웹 크롤러 만들기 요구사항 |
| 04. 나만의 웹 크롤러 만들기 기능정의 및 설계 |

CURRICULUM

## [Course 2].

## Part 6. 실전 프로젝트 리뷰

파트별 수강시간 00:55:33

| Ch 01. 실전프로젝트 해설 |
|---|
| 01. 이력서 자동생성 프로그램 해설 |
| 02. 나만의 웹 크롤러 만들기 해설 |

Fast campus

# 커리큘럼

CURRICULUM

## [Course 3].

## 웹 개발 입문과 데이터베이스

파트별 수강시간 15:15:14

| |
|---|
| **Ch 00. 들어가며** |
| 01. 강의소개 및 강사소개 |
| 02. 개발 환경 설치 |
| **Ch 01. Web과 HTTP 통신에 대해서 알아보기** |
| 01. WEB 개론 |
| 02. REST API 개론 |
| 03. Spring Boot 와 Spring Web 소개 |
| **Ch 02. REST API** |
| 01. REST API - GET - 01 |
| 02. REST API - GET - 02 |
| 03. REST API - GET - 03 |
| 04. REST API - POST - 01 |
| 05. REST API - POST - 02 |
| 06. REST API - PUT |
| 07. REST API - DELETE |
| **Ch 03. Spring Boot Web** |
| 01. Spring Boot Web에서 응답 만드는 방법 - Resposne Entity |
| 02. Spring Boot의 다양한 기능 살펴보기 |
| 03. Spring Boot 에서의 예외처리 소개 - 1 |
| 03. Spring Boot 에서의 예외처리 소개 - 2 |
| 04. Spring Boot 예외처리 실전 적용 - 1 |
| 05. Spring Boot 예외처리 실전 적용 - 2 |
| 06. Spring Boot Validation 소개 |
| 07. Spring Boot Validation 실전 적용 - 1 |
| 08. Spring Boot Validation 실전 적용 - 2 |
| 09. Spring Boot Validation 실전 적용 - 3 |
| **Ch 04. Memory DataBase** |
| 01. Memory DataBase CRUD 적용해보기 - 1 |
| 02. Memory DataBase CRUD 적용해보기 - 2 |
| 03. Memory DataBase CRUD 적용해보기 - 3 |
| 04. Memory DataBase CRUD 적용해보기 - 4 |
| 05. Memory DataBase CRUD 적용해보기 - 5 |

CURRICULUM

# [Course 3].

## 웹 개발 입문과 데이터베이스

파트별 수강시간 15:15:14

| |
|---|
| **Ch 05. MySQL** |
| 01. 데이터 베이스 설치 및 설정 |
| 02. 간단한 MySQL 쿼리 배우기 - 1 SQL 쿼리 알아보기 |
| 03. 간단한 MySQL 쿼리 배우기 - 2 SQL 쿼리 실습 |
| 04. 간단한 MySQL 쿼리 배우기 - 3 SQL 쿼리 실습 |
| 05. 간단한 MySQL 쿼리 배우기 - 4 SQL 데이터타입 |
| **Ch 06. Spring Data JPA** |
| 01. Spring Data 소개 |
| 02. Spring Data 적용 및 개발 |
| **Ch 07. JPA로의 변환** |
| 01. Memory DB를 MySQL로 변환 - 01 |
| 02. Memory DB를 MySQL로 변환 - 02 |
| **Ch 08. 간단한 게시판 프로젝트** |
| 01. 간단한 게시판 시스템 설계 |
| 02. 간단한 게시판 테이블 설계 |
| 03. 간단한 게시판 개발하기 Entity 개발 |
| 04. 간단한 게시판 개발하기 API End Point 개발 - 1 |
| 05. 간단한 게시판 개발하기 API End Point 개발 - 2 |
| 06. 간단한 게시판 개발하기 API End Point 개발 - 3 |
| 07. 간단한 게시판 개발하기 JPA 연관관계 설정하기 -1 |
| 08. 간단한 게시판 개발하기 JPA 연관관계 설정하기 - 2 |
| **Ch 09. Spring Boot Web 활용** |
| 01. Filter 와 Interceptor |
| 02. Interceptor 적용 |
| 03. Spring AOP |
| 04. Spring AOP Pointcut 문법 |
| **Ch 10. 간단한 게시판 프로젝트 개선** |
| 01. 추상화를 통한 게시판 프로젝트 개선 - 1 |
| 02. 추상화를 통한 게시판 프로젝트 개선 - 2 |

Fast campus

# 커리큘럼

CURRICULUM

## [Course 4].

## 프로젝트로 시작하는 웹 서비스 개발

파트별 수강시간 18:25:55

| |
|---|
| **Ch 00. 들어가며** |
| 01. 강의 소개 |
| **Ch 01. Web Service의 인증** |
| 01. 인증이란? |
| 02. HTTP Session 인증 - 1 |
| 02. HTTP Session 인증 - 2 |
| 03. HTTP Cookie 인증 - 1 |
| 03. HTTP Cookie 인증 - 2 |
| 04. HTTP Header 인증 |
| 05. JWT Token 인증 - 1 |
| 05. JWT Token 인증 - 2 |
| **Ch 02. 실전 프로젝트 1: 배달 플랫폼 백엔드 구축** |
| 01. 배달 플랫폼 프로젝트 생성 |
| 02. 배달 플랫폼 프로젝트 멀티모듈 설정하기 |
| 03. 배달 플랫폼 프로젝트 멀티모듈 설정하기 - 2 |
| 04. 배달 플랫폼 프로젝트 DB 모듈 설정하기 - 1 |
| 05. 배달 플랫폼 프로젝트 DB 모듈 설정하기 - 2 |
| 06. 배달 플랫폼 프로젝트 API 기본 설정 추가_1 |
| 07. 배달 플랫폼 개발하기 - Swagger UI 설정 |
| **Ch 03. 실전 프로젝트 2: 프로젝트 베이스 개발** |
| 01. Filter를 통한 Request, Response Log 설정하기 |
| 02. Api 공통 Spec 적용하기 |
| 03. Api Error Code 적용하기 |
| 04. Exception Handler 적용하기 - 1 |
| 04. Exception Handler 적용하기 - 2 |
| 05. Interceptor 를 통한 인증 기반 적용하기 |
| **Ch 04. 실전 프로젝트 3: 사용자 도메인 개발** |
| 01. 사용자 데이터 베이스 개발 |
| 02. UserEntity 개발 |
| 03. 사용자 서비스 로직 - 1 |
| 03. 사용자 서비스 로직 - 2 |
| 04. JWT 토큰 서비스 로직 적용하기 |
| 05. 사용자 로그인 토큰 발행 적용하기 |
| 06. 사용자 인증 로직 적용하기 |

Fast campus

커리큘럼

CURRICULUM

# [Course 4].

## 프로젝트로 시작하는 웹 서비스 개발

파트별 수강시간 18:25:55

| |
|---|
| **Ch 05. 실전 프로젝트 4: 스토어 도메인 개발** |
| 01. Store 데이터 베이스 개발 |
| 02. Store 서비스 로직 개발 |
| 03. Store Menu 데이터 베이스 설계 |
| 04. Store Menu 서비스 로직 개발 |
| **Ch 06. 실전 프로젝트 5 : 상품 주문 개발** |
| 01. User Order 데이터 베이스 설계 |
| 02. User Order 서비스 로직 개발 - 1 |
| 03. User Order 서비스 로직 개발 - 2 |
| 04. User Order 서비스 로직 개발 - 3 |
| 05. User Order 서비스 로직 개발 - 4 |
| **Ch 07. 실전 프로젝트 6 : 가맹점 서버 개발** |
| 01. Spring Security 소개 |
| 02. Spring Security를 통한 가맹점 서버 개발 - 1 |
| 03. Spring Security를 통한 가맹점 서버 개발 - 2 |
| 04. 가맹점 유저 가입 개발 |
| 05. Spring Security 에서의 가맹점 유저 로그인 처리 |
| 06. Spring Security 에서의 사용자 정보 확장하기 |
| **Ch 08. 실전 프로젝트 7 : 비동기 주문 개발** |
| 01. 비동기 처리란? |
| 02. 비동기를 위한 Message Queue |
| 03. Docker에 Rabbit MQ 설정하기 |
| 04. Producer 개발하기 -1 |
| 05. Producer 개발하기 - 2 |
| 06. Consumer 개발하기 |
| 07. SSE Server Sent Events 란? |
| 08. SSE를 통한 사용자 주문 Push 알림 개발하기 - 1 |
| 09. SSE를 통한 사용자 주문 Push 알림 개발하기 - 2 |
| 10. SSE를 통한 사용자 주문 Push 알림 개발하기 - 3 |
| 11. SSE를 통한 사용자 주문 Push 알림 개발하기 - 4 |

Fast campus

# 커리큘럼

CURRICULUM

## [Course 5].

## 백엔드 웹 개발 실전/최신

파트별 수강시간 16:25:51

| |
|---|
| **Ch 00. 들어가며** |
| 01. 강의소개 |
| **Ch 01. Kotlin 소개** |
| 01. Kotlin 소개 |
| 02. Kotlin과 Java 코드 비교하며 배워보기 - 1 변수선언 |
| 03. Kotlin과 Java 코드 비교하며 배워보기 - 2 null 안정성과 엘비스 연산자 |
| 04. Kotlin과 Java 코드 비교하며 배워보기 - 3 가변, 불변 컬렉션 |
| 05. Kotlin과 Java 코드 비교하며 배워보기 - 4 고차함수 |
| 06. Kotlin과 Java 코드 비교하며 배워보기 - 5 클래스 |
| 07. Kotlin과 Java 코드 비교하며 배워보기 - 6 data class |
| 08. Kotlin과 Java 코드 비교하며 배워보기 - 7 default value |
| 09. Kotlin과 Java 코드 비교하며 배워보기 - 8 when |
| 10. Kotlin과 Java 코드 비교하며 배워보기 - 9 확장함수 |
| 11. Kotlin 표준함수 let |
| 12. Kotlin 표준함수 also |
| 13. Kotlin 표준함수 apply |
| 14. Kotlin 표준함수 run |
| **Ch 02. Spring 과 Kotlin** |
| 01. Java에서 Kotlin 함께 사용하기 |
| 02. Spring 에서 Kotlin 적용하기 |
| 03. 기존 프로젝트를 Kotlin 으로 변경하기 - 1 kotlin 설정 추가하기_1 |
| 04. 기존 프로젝트를 Kotlin 으로 변경하기 - 2 common 모듈 옮기기 |
| 05. 기존 프로젝트를 Kotlin 으로 변경하기 - 3 common 모듈 옮기기 |
| 06. 기존 프로젝트를 Kotlin 으로 변경하기 - 4 JPA 다루기 |
| 07. 기존 프로젝트를 Kotlin 으로 변경하기 - 5 JPA 연관관계 설정 |
| 08. 기존 프로젝트를 Kotlin 으로 변경하기 - 6 Entity |
| 09. 기존 프로젝트를 Kotlin 으로 변경하기 - 7 Kotlin Entity |
| 10. 기존 프로젝트를 Kotlin 으로 변경하기 - 8 비지니스 로직 변경 |
| **Ch 03. 마이크로서비스 맛보기** |
| 01. 마이크로서비스란- |
| 02. API Gateway |
| 03. Spring Cloud API Gateway -1 |
| 04. Spring Cloud API Gateway - 2 |
| 05. API Gateway 인증 - 1 |
| 06. API Gateway 인증 - 2 |
| 07. API Gateway 인증 - 3 |
| 08. API Gateway 인증 - 4 |

CURRICULUM

# [Course 5].

## 백엔드 웹 개발 실전/최신

파트별 수강시간 16:25:51

| |
|---|
| **Ch 04. 모니터링** |
| 01. 모니터링 |
| 02. Spring Application Log 모니터링 하기 |
| 03. ELK Stack 을 통한 Log 모니터링 |
| 04. Spring Boot Admin 을 통한 모니터링 |
| 05. Prometheus Grafana 를 통한 모니터링 |
| 06. TICK Stack을 통한 모니터링 |
| **Ch 05. 마치며** |
| 01. 마치며 - 1 redis의 도입 |
| 02. 마치며 - 2 redis의 도입 |
| 03. 마치며 - 3 redis의 도입 |
| 04. 마치며 |

Fast campus

# 커리큘럼

CURRICULUM

## [Course 6].

## 대용량 데이터를 다루는 배치 처리

파트별 수강시간 09:48:17

| |
|---|
| **Ch 00. OT** |
| 01. OT |
| Ch 01. Spring batch 원리와 철학 이해 |
| 01. 배치 프로그램이란 무엇인가 |
| 02-1. 프로젝트 세팅 및 (원시적인) 배치 프로그램 구현하기 - 기반 구현 |
| 02-2. 프로젝트 세팅 및 (원시적인) 배치 프로그램 구현하기 - 본 구현 |
| 03. 스프링 배치처럼 개선하기 |
| 04. 단계가 있는 배치 만들기 |
| **Ch 02. Spring batch 적용** |
| 01-1. Spring Batch 도메인 용어 익히기 |
| 01-2. Spring Batch 도메인 용어 익히기 - 적용 |
| 02. Job |
| 03-1. Step |
| 03-2. Step - 기본 적용 |
| 03-3. Step - Flow 적용 |
| 04-1. ItemReader |
| 04-2. ItemReader - File |
| 04-3. ItemReader - Database |
| 05-1. ItemWriter |
| 05-2. ItemWriter - 적용 |
| 06. ItemProcessor |
| 07-1. 확장을 통한 성능 개선 |
| 07-2. 확장을 통한 성능 개선 - Multi-threaded, Parallel |
| 07-3. 확장을 통한 성능 개선 - Partitioning |
| **Ch 03. 정산 시스템 만들기** |
| 01. 요구사항 분석 |
| 02. API 호출 이력 배치 만들기 |
| 03. API 호출 이력 배치 확장하기 |
| 04. 일일 정산 배치 만들기 |
| 05. 주간 정산 배치 만들기 |
| **Ch 04. 백엔드 개발자의 N가지 성장전략** |
| 01. 백엔드 개발자의 N가지 성장전략 |

Fast campus

커리큘럼

CURRICULUM

# [Course 7].

## 대용량 트래픽 처리를 위한 백엔드 심화

파트별 수강시간 16:34:43

| |
|---|
| **Ch 01. 들어가며** |
| 01. 강의 소개 |
| 02. 개발 환경 준비 |
| **Ch 02. Redis** |
| 01. in-memory database |
| 02. Reids 소개 |
| 03. Redis 설치 |
| 04. Redis CLI을 통한 접속 |
| 05. Data types에 대한 이해 |
| 06. Data types String 실습 |
| 07. Data types List, Set 실습 |
| 08. Data types Hash 실습 |
| 09. Data types Sorted Set 실습 |
| 10. Data types Geospatial 실습 |
| 11. Data types Bitmap 실습 |
| 12. Transactions |
| 13. Keys, Scan 명령어 |
| 14. Cache 이론 |
| 15. Cache 실습 |
| 16. Spring Boot Cache |
| 17. Spring Boot Session Store |
| 18. Spring Boot Pub Sub |
| 19. Monitoring |
| 20. Replication |
| **Ch 03. Webflux** |
| 01. Spring Webflux 소개 |
| 02. CPU bound vs IO Bound |
| 03. synca sync와 block non-block |
| 04. Spring MVC vs Webflux |
| 05. Reactor 이론 |
| 06. Reactor 실습 |
| 07. Spring Webflux 실습(1) |
| 08. Spring Webflux 실습(2) |
| 09. R2DBC 이론 |
| 10. R2DBC 실습(1) |
| 11. R2DBC 실습(2) |
| 12. Reactive Redis 이론 |
| 13. Reactive Redis 실습 |
| 14. Spring MVC vs Webflux 성능 |
| 15. Blockhound |

Fast campus

# 커리큘럼

CURRICULUM

## [Course 7].

## 대용량 트래픽 처리를 위한 백엔드 심화

파트별 수강시간 16:34:43

| Ch 04. 프로젝트 - 접속자대기시스템 |
| --- |
| 01. 요구사항 분석 |
| 02. 아키텍처 설계 |
| 03. 개발 환경 준비 |
| 04. 대기열 등록 API 개발 |
| 05. 진입 요청 API 개발 |
| 06. 접속 대기 웹페이지 개발 |
| 07. 대기열 스케쥴러 개발 |
| 08. 대기열 이탈 |
| 09. 테스트 |
| 10. 마무리 |

Fast campus

# 커리큘럼

CURRICULUM

## [Course 8].

## 백엔드 개발자를 위한 DevOps 맛보기

파트별 수강시간 24:57:33

| |
|---|
| **Ch 01. 백엔드 개발자를 위한 DevOps 입문 소개** |
| 01. OT |
| **Ch 02. 백엔드 아키텍처 맛보기 (멀티모듈, MSA, DDD)** |
| 01. DDD(Domain Driven Design) 이해하기 |
| 02. MSA 아키텍쳐 기반의 서비스 이해 |
| 03. 멀티모듈로 구성하는 코드 - Java기반 |
| 04. LAB code로 다루는 DDD - MSA - multi-module 사례보기 |
| **Ch 03. 백엔드 개발자를 위한 클라우드 인프라 운영 - Amazon** |
| 01. Amazon 클라우드 사용하기(1) - 클라우드란 |
| 02. Amazon 클라우드 사용하기(2) - AWS 계정 만들기 |
| 03. Amazon 클라우드 사용하기(3) - IAM 사용자, 정책, 권한 |
| 04. Amazon 클라우드 사용하기(4) - 프리티어 서비스 |
| 05. AWS 클라우드 네트워크 운영(이론) |
| 06. AWS 클라우드 네트워크 운영(실습) |
| 07. Amazon EC2 인스터스 만들기(이론) |
| 08. Amazon EC2 인스터스 만들기(실습) |
| 09. Amazon EC2 원격 로그인 실습 (for Windows) |
| 10. Amazon EC2 원격 로그인 실습(for Mac) |
| 11. Spring 기반 Web 서버 만들기 |
| 12. Amazon RDS에 데이터 저장하기(이론) |
| 13. Amazon RDS에 데이터 저장하기(실습) |
| 14. 간단한 클라우드 기반 Application 운영 |
| 15. Amazon Route53 으로 도메인 서비스 구축하기(이론) |
| 16. Amazon Route53 으로 도메인 서비스 구축하기(실습) |
| 17. Amazon 리소스 삭제 |
| 18. LAB- AWS 개발환경 만들기 - 답안 & 해설 1-2 |
| 19. LAB- AWS 개발환경 만들기 - 답안 & 해설 2-2 |
| **Ch 04. 백엔드 개발자를 위한 컨테이너 서비스 - Docker** |
| 01. 컨테이너 개념과 동작원리 |
| 02. docker 명령어!! 이것만 알면된다. -이론 |
| 03. docker 명령어!! 이것만 알면된다. - 실습 |
| 04. docker 스토리지 관리 |
| 05. docker 네트워크 관리 |
| 06. Best Practice & Security를 고려한 컨테이너 빌드(1) |
| 07. Best Practice & Security를 고려한 컨테이너 빌드(2) |
| 08. 컨테이너 저장소 운영하기(1) |
| 09. 컨테이너 저장소 운영하기(2) |
| 10. Docker Compose 사용하기(1) |
| 11. Docker Compose 사용하기(2) |
| 12. 현업 사례 중심 컨테이너 기반 애플리케이션 운영 |
| 13. LAB 컨테이너 기반 애플리케이션 서비스 운영하기 - 연습문제 풀이 |

Fast campus

# 커리큘럼

CURRICULUM

## [Course 8].

## 백엔드 개발자를 위한 DevOps 맛보기

파트별 수강시간 24:57:33

| |
|---|
| **Ch 05. 클라우드 기반 컨테이너 서비스 운영 - Amazon** |
| 01. Amazon 컨테이너 서비스 이해 (ECR, ECS) |
| 02. Amazon ECR 리포지토리 운영 실습 |
| 03. Amazon ECS 기반 컨테이너 애플리케이션 배포 실습 (1) |
| 04. Amazon Fargate 이해 |
| 05. Amazon Fargate로 컨테이너 애플리케이션 배포 실습 |
| 06. Amazon 리소스 삭제 |
| 07. LAB 현업중심의 Amazon 컨테이너 서비스를 이용한 컨테이너빌드에서 배포까지 - 문제_1 |
| 08. LAB Amazon 컨테이너 서비스를 이용한 컨테이너빌드에서 배포까지 - 답안 & 해설 |
| **Ch 06. 백엔드 개발자를 위한 빌드자동화 - Jenkins** |
| 01. Git을 이용한 소스코드 버전관리 |
| 02. Git Brunch를 통한 협업 개발하기 |
| 03. Jenkins 프로그램 소개 및 설치하기 |
| 04. GIt 연동을 위한 Jenkins 환경설정 |
| 05. Jenkins 파이프라인을 이용한 컨테이너 빌드 |
| 06-1. Jenkins Slack Webhook 연동하기 |
| 06-2. Jenkins Slack Webhook 연동하기 |
| 07. 실전 빌드하기 - source push → container build → ecr에 업로드 ( FLOW arch 설명 ) |
| 08. LAB- git 코드 업로드 부터 컨테이너 빌드까지 - CI 구축하기 |
| **Ch 07. [프로젝트] 웹 개발 프로젝트 6 : 아키텍처 적용한 MVP 프로젝트 배포하기** |
| 01. MVP 프로젝트를 위한 DDD - MSA - 멀티모듈 기반의 백엔드 아키텍처 설계 |
| 02. MVP 프로젝트를 위한 프론트 & 백엔드 소스코드 리뷰 |
| 03. MVP 프로젝트 운영을 위한 클라우드 인프라 구축 |
| 04. DevOps 구성을 위한 GIT JENKINS 서버 구성 |
| 05. DevOps 구성을 위한 Jenkins CI-CD 빌드 배포라인 구성 |
| 06. 컨테이너 기반의 애플리케이션을 ECS Fargate에서 운영 - 1 |
| 07. 컨테이너 기반의 애플리케이션을 ECS Fargate에서 운영 - 2 |

Fast campus

# 커리큘럼

CURRICULUM

## [부록1].

## 신입 개발자 Start Kit

파트별 수강시간 04:48:05

| |
|---|
| **Ch 01. 인텔리제이란** |
| 01. 인텔리제이 없이 코딩해보기 |
| 02. 인텔리제이란? |
| 03. 인텔리제이 둘러보기 |
| **Ch 02. 인텔리제이 가이드** |
| 01. 인텔리제이 활용하기 |
| 02. 인텔리제이 활용하기 - 검색 |
| 03. 인텔리제이 활용하기 - 코딩 |
| 04. 인텔리제이 활용하기 - 디버깅 |
| **Ch 03. Git** |
| 01. Git 이란 |
| 02. Git 설치하기 |
| 03. Git 명령어 |
| 04. Git 브랜치 전략 |
| **Ch 04. 프로젝트 실습** |
| 01. 프로젝트 개요 |
| 02. 환경설정 및 프로젝트 구성 |
| 03. 프로젝트 구현 1 |
| 04. 프로젝트 구현 2 |
| 05. 프로젝트 구현 3 |
| 06. 테스트 코드 작성하기 |
| 07. 디버깅 실습 |

Fast campus

# 커리큘럼

CURRICULUM

## [부록2].

## 단기완성 알고리즘

파트별 수강시간 14:29:22

| |
|---|
| **Ch 01. 코딩 테스트를 위한 준비** |
| 01. 강의 소개 및 최종 목표 |
| 02. 최신 입사 코딩 테스트 분석 |
| 03. 꿀팁 - 좋은 습관 |
| **Ch 02. 알고리즘** |
| 01. 어떻게든 푼다. 완전 탐색 (Brute Force) |
| 01. 어떻게든 푼다. 완전 탐색 (Brute Force) - 응용편 |
| 02. 다양한 정렬 응용법(Sort Application) |
| 02. 다양한 정렬 응용법(Sort Application) - 응용편 |
| 03. 이분 탐색 (Binary Search) |
| 03. 이분 탐색 (Binary Search) - 응용편 |
| 04. 두 포인터(Two Pointers) |
| 04. 두 포인터(Two Pointers) - 응용편 |
| 05. 그래프와 탐색(Graph & Search) |
| 05. 그래프 탐색 (Graph Search, DFS & BFS) - 응용편 1 |
| 05. 그래프 탐색 (Graph Search, DFS & BFS) - 응용편 2 |
| 06. 트리(Tree) |
| 07. 위상정렬 |
| 08. 최단 경로 (Shortest Path) |
| 09. 동적 프로그래밍 (Dynamic Programming) |
| 10. 동적 프로그래밍 (Dynamic Programming) - 2 |
| 11. 동적 프로그래밍 (Dynamic Programming) - 3 |
| **Ch 03. 모의 코딩 테스트 풀이** |
| 01. 류호석배 1회 모의 코딩 테스트 |
| 02. 류호석배 2회 모의 코딩 테스트 |
| 03. 류호석배 3회 모의 코딩 테스트 |

CURRICULUM

# [부록3].

## 백엔드 개발자 해외 취업 A-Z

파트별 수강시간 04:51:18

| |
|---|
| **Ch 01. 오리엔테이션** |
| 01. 강의소개 |
| 02. FAQ |
| **Ch 02. Application Process Tips** |
| 01. 해외 취업 프로세스 |
| 02. 취업 사전 준비 |
| 03. 직무 고르기 |
| 04. 용어와 참고사항 |
| **Ch 03. Resume & Cover Letter** |
| 01. 레쥬메란? |
| 02. 글쓰기 팁 |
| 03. 레쥬메 구성 |
| 04. 레쥬메 예시 |
| 05. 커버레터 |
| **Ch 04. Soft Skill Interview 인적성** |
| 01. 커뮤니케이션 |
| 02. FAQ |
| 03. 정리법 |
| 04. 예시 1 |
| 05. 예시 2 |
| **Ch 05. Hard Skill Interview 기술 / 코딩 인터뷰** |
| 01. scope |
| 02. 답변 구성 |
| 03. 예제 1 |
| 04. 예제 2 |

Fast campus

Signature

## 커리큘럼

CURRICULUM

# [부록4].

# IT 대기업 이직을 위한 면접 팁

파트별 수강시간 04:51:18

| |
|---|
| **Ch 01. 차가운 현실** |
| 01. 회사는 꿈을 이뤄주는 곳이 아니다- |
| **Ch 02. 자소서** |
| 01. 지원 동기의 함정 |
| 02. 읽기 좋은 자소서 작성법 |
| 03. 포트폴리오 작성 |
| **Ch 03. 코딩 테스트** |
| 01. 경력 코딩 테스트 팁 |
| 02. 코드 리뷰 준비 방법 |
| **Ch 04. 면접** |
| 01. 면접 왜 어려울까 ? |
| 02. 면접의 흐름 |
| 03. 선택과 집중 |
| 04. 면접에 대한 고찰 |

Fast campus

# 안내 사항

Signature

• **주의 사항**

a. 상황에 따라 사전 공지 없이 할인이 조기 마감 되거나 연장될 수 있습니다.

b. 패스트캠퍼스의 모든 온라인 강의는 아이디 공유를 금지하고 있으며
1개의 아이디로 여러 명이 수강하실 수 없습니다.

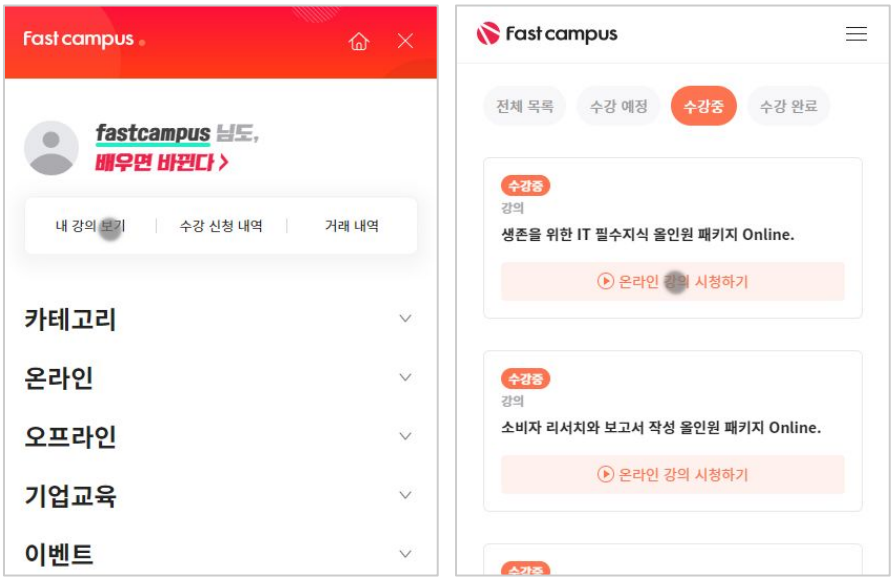
c. 별도의 주의사항은 각 강의 상세페이지에서 확인하실 수 있습니다.

• **수강 방법**

a. 패스트캠퍼스는 크롬 브라우저에 최적화 되어 있습니다.

b. 로그인 후, [ 내 강의 보기 클릭 ] 수강중 클릭 | 온라인 강의 시청하기

c. 사전 예약 판매 중인 강의의 경우 1차 공개일정에 맞춰 '온라인 강의 시청하기'가 활성화 됩니다.




• **환불 규정**

a. 온라인 강의는 각 과정 별 '정상 수강기간(유료수강기간)'과 정상 수강기간 이후의
'복습 수강기간(무료수강기간)'으로 구성됩니다.

b. 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<br>환불규정에 따라 환불 가능 |

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