# 교육
# 과정
# 소개서.

현실 세상의 컴퓨터공학 지식 with 30가지 실무 시나리오
초격차 패키지 Online.

Fast campus

## 강의정보

| | |
|---|---|
| 강의장 | 온라인 강의 | 데스크탑, 노트북, 모바일 등 |
| 수강 기간 | 평생 소장 |
| 상세페이지 | https://fastcampus.co.kr/dev_online_newcomputer |
| 강의시간 | 94시간 |
| 문의 | 고객센터 |

## 강의특징

| | |
|---|---|
| 나만의<br>속도로 | 낮이나 새벽이나<br>**내가 원하는 시간대**에 나의 스케줄대로 수강 |
| 원하는 곳<br>어디서나 | 시간을 쪼개 먼 거리를 오가며<br>오프라인 강의장을 찾을 필요 없이 **어디서든 수강** |
| 무제한<br>복습 | 무엇이든 반복적으로 학습해야<br>내것이 되기에 이해가 안가는 구간 **몇번이고 재생** |



Copyright ⓒ FAST CAMPUS Corp. All Rights Reserved.

## 강의목표

- 실무 연관성이 높은 CS지식을 몽~땅 모아 이론과 실습을 통해 가장 이해하기 쉽게 알려드립니다.
- Google 출신 개발자가 알려주는 CS지식을 활용한 실무 트러블슈팅 사례를 확인해 봅니다.
- IT대기업 기술면접 기출변형 50선부터 개발자라면 꼭 알아야 할 개발자 협업 툴 & AI 도구 사용법을 배워봅니다.

## 강의요약

- 다양한 문제 상황을 통해 CS지식이 어떻게 활용 되는지, 실무에서 발생하는 다양한 문제 상황에 대처할 수 있는 방법을 배워봅니다.
- 실무 연관성을 고려한 15개 CS지식뿐 아니라, 실무 능력 향상을 위한 도구 활용법까지 개발자의 '필수 기본기'를 한 번에 배워봅니다.


Copyright ⓒ FAST CAMPUS Corp. All Rights Reserved.

## 강사

**강민철**

**과목**
- Course.1 컴퓨터 공학

**약력**
- 현) 고려대학교 석박사 통합 과정
- 전) 펜타시큐리티시스템 개발자
- 전) 오픈소스컨설팅 솔루션 아키텍트

- [집필 저서]
- 혼자 공부하는 컴퓨터구조 + 운영체제 (교보문고 2022 올해의 IT 책 선정)
- 모두의 Git & GitHub
- 혼자 공부하는 네트워크(출판 예정)

- [강의 이력]
- 패스트캠퍼스 외 다수의 대학 및 교육업체 강의 (컴퓨터 공학, 자료구조/알고리즘, Git&GitHub)

- [기타 이력]
- 한국 코드페어 멘토
- 오픈소스 컨튜리뷰션 아카데미 리드 멘토
- 각종 컨퍼런스 기술 발표
- 각종 오픈소스 프로젝트 참여

**박매일**

**과목**
- Course.2 소프트웨어공학

**약력**
- [강의 이력]
- 한양대학교, ERICA, 조선대학교 외 다수 교육기관 강의
- 한국 스마트 미디어 학회, 한국 전력공사 외 다수 기업 강의
- 인프런 : Java TPC (생각하고, 표현하고, 코딩하고) 등
- 패스트캠퍼스 : 내 인생 처음이자 마지막 자바, 백엔드 시그니처


Copyright ⓒ FAST CAMPUS Corp. All Rights Reserved.

## 강사

**신경식**

| | |
|---|---|
| **과목** | - Course.3 컴퓨터과학 |
| **약력** | - [강의 이력]<br>- 패스트캠퍼스 수학적으로 접근하는 딥 러닝<br>- 딥러닝을 위한 파이썬 레벨1, 레벨2<br>- 삼성전기 신입사원 교육 16주<br>- 원광대학교 AI 장단기 과정<br>- 기상과학원 AI 부스트캠프<br>- 국가 과학 기술 인력 개발원 R&D 수행 역량 강화 장기 멘토링 외 다수<br><br>- [기타 이력]<br>- 유튜브 공대형아 운영 |

**박은종**

| | |
|---|---|
| **과목** | - Course.3 컴퓨터과학 |
| **약력** | - 전) 현대 정보기술 사내 시스템 개발<br>- 전) 스펙트라 개발 팀장<br>- 전) NHN NEXT 겸직 교수<br>- 전) 이노베이션 아카데미 멘토<br><br>- [집필 저서]<br>- Do it 자바 프로그래밍 입문<br><br>- [강의 이력]<br>- 패스트캠퍼스 자바 강의 외 다수<br>- 카카오 디자인 패턴 사내 강의 |


Copyright ⓒ FAST CAMPUS Corp. All Rights Reserved.

## 강사

| | | | |
|---|---|---|---|
| **이광운** | **과목** | - | Course.4 소프트웨어 공학 심화 |
| | **약력** | - | 현) 야놀자 F&B CTO |
| | | - | 전) 트렌비 Tech Leader |
| | | - | 전) Tech Leader |
| | | - | [강의 이력] |
| | | - | 패스트캠퍼스 백엔드 개발자를 위한 한 번에 끝내는 대용량 데이터 & 트래픽 처리 |
| | | - | 신서계 I&C Go Programming 강의 |
| | | - | 신한 DS GO Programming 강의 |
| | | - | LG CNS Python 프로그래밍 강의 |
| **CYJ** | **과목** | - | Course.5 실무 문제 해결 |
| | **약력** | - | 현) 카카오계열사/ SRE |
| | | - | 전) 클라우드 공급업체/ Solution Architect |
| | | - | 전) LG 계열사 Cloud Architect, System admin |
| | | - | [강의 이력] |
| | | - | 패스트캠퍼스 한 번에 끝내는 AWS 기반 아키텍처 설계와 Devops |
| **정채상** | **과목** | - | Course.5 실무 문제 해결 |
| | **약력** | - | 전) 뱅크샐러드 / 개발 이사 |
| | | - | 전) Google / Tech Lead Manager |
| | | - | 전) Google / Software Lead Manager |
| | | - | [강의 이력] |
| | | - | Google IO 2015 - Get your app into the google index |
| | | - | Playtime- KR @ 2017 - keynote “Google Play Updates - Play Store” |
| | | - | 실리콘밸리 부트캠프 멘토 |



Copyright ⓒ FAST CAMPUS Corp. All Rights Reserved.

CURRICULUM

# [Course 1].

## Part 1.
## 컴퓨터구조

파트별 수강 시간 06:41:58

| |
|---|
| **CH01. 오리엔테이션** |
| 01. 컴퓨터구조 거시적으로 보기 |
| **CH02. 명령어** |
| 01. 소스코드에서 명령어로 |
| 02. [실습] 컴파일 - 명령어 관찰하기 |
| 03. 명령어 구조 |
| 04. 주소 지정 |
| **CH03. 데이터** |
| 01. 이진수와 2의 보수 |
| 02. 부동 소수점 |
| 03. 문자 인코딩과 디코딩 |
| **CH04. CPU** |
| 01. CPU 구성 요소 |
| 02. 레지스터 살펴보기 |
| 03. 명령어 사이클과 인터럽트 |
| 04. 멀티 코어와 멀티 프로세서 |
| 05. 명령어 병렬 처리 - 파이프라이닝 |
| 06. 비순차적명령어처리 |
| **CH05. 메인 메모리와 캐시 메모리** |
| 01. RAM & ROM |
| 02. 리틀 엔디안과 빅 엔디안 |
| 03. 주소 공간 - 논리 주소와 물리 주소 |
| 04. 저장 장치 계층 구조와 캐시 메모리 |
| 05. [실습] 캐시 친화적 코드 |
| **CH06. 보조기억장치와 입출력장치** |
| 01. 하드 디스크와 플래시 메모리 |
| 02. RAID |
| 03. 입출력 기법 |

CURRICULUM

# [Course 1].

# Part 2. 운영체제

파트별 수강 시간 06:12:39

| |
|---|
| **CH01. 오리엔테이션** |
| 01. 운영체제 거시적으로 보기 |
| 02. [실습] 가상 머신에 리눅스 설치하기 |
| 03. strace 기반 시스템 콜 관찰하기 |
| 04. 대표적인 리눅스 시스템 콜 |
| **CH02. 프로세스와 스레드** |
| 01.커널 영역과 사용자 영역의 프로세스 |
| 02.프로세스 생성과 상태 |
| 03.스레드 |
| 04.[실습]멀티프로세스와 멀티스레드 |
| **CH03. CPU 스케줄링** |
| 01.프로세스 우선순위와 스케줄링 큐 |
| 02.CPU 스케줄링 알고리즘 |
| 03.리눅스의 스케줄링 |
| **CH04. 동기화와 교착상태** |
| 01.프로세스 동기화 |
| 02.[실습] 생산자와 소비자 문제 |
| 03.뮤텍스와 세마포어 |
| 04.조건변수와 모니터 |
| 05.[실습] 동기화 실습 |
| 06.교착상태와 해결 방법 |
| **CH05. 가상 메모리 관리** |
| 01.페이징과 페이지 테이블 |
| 02.요구 페이징, 스래싱 |
| 03.페이지 폴트 관찰 |
| 04.페이지 교체 알고리즘 |
| 05.copy on write |
| **CH06. 파일 시스템** |
| 01.파일과 디렉터리 |
| 02.파일 시스템이 만들어지기까지 |
| 03.파일 시스템 만들고 연결하기 실습 |
| 04.파일 시스템 종류와 특성 |

CURRICULUM

# [Course 1].

# Part.3 네트워크

파트별 수강 시간 08:08:54

| |
|---|
| **CH01. 오리엔테이션** |
| 01.네트워크 거시적으로 보기 |
| 02.프로토콜과 캡슐화 |
| 03.[실습] 와이어샤크 |
| 04.네트워크의 성능 |
| **CH02. 네트워크 엑세스 계층** |
| 01.이더넷 |
| 02.허브와 CSMA/CD |
| 03.스위치와 VLAN |
| **CH03. 네트워크 계층** |
| 01.IP |
| 02.ARP |
| 03.ICMP |
| 04.[실습] IPv4, IPv6, ARP, ICMP 패킷 분석 |
| 05.IP 주소 |
| 06.IP 주소의 분류 |
| 07.라우팅 |
| **CH04. 전송 계층** |
| 01.포트 |
| 02.[실습] 포트 확인하기 |
| 03.TCP와 UDP |
| 04.[실습] TCP, UDP 패킷 분석 |
| 05.TCP 연결 |
| 06.TCP 상태 |
| 07.TCP 재전송 기능 |
| 08.TCP의 혼잡 제어와 흐름 제어 |
| **CH05. 응용 계층** |
| 01.DNS |
| 02.자원과 자원의 식별 |
| 03.웹 서버와 웹 어플리케이션 서버 |
| 04.HTTP의 특성 |
| 05.HTTP 메세지 개관 |
| 06.HTTP 헤더 |
| 07.[실습] HTTP 요청 - 응답 메세지 직접 확인하기 |
| 08.캐시 |
| 09.쿠키 |
| 10.컨텐츠 협상 |

CURRICULUM

# [Course 1].

## Part.4 시스템 프로그래밍

파트별 수강 시간 05:21:05

| |
|---|
| **CH01. 오리엔테이션** |
| 01.시스템 프로그래밍의 개념 |
| 02.vi/vim 편집기 |
| **CH02. 파일 다루기** |
| 01.파일 디스크립터와 파일 포인터 |
| 02.파일 입력 |
| 03.파일 출력 |
| 04.디렉터리 다루기 |
| 05.하드 링크와 심볼릭 링크 |
| 06.파일 속성 다루기 |
| 07.파일 메모리 매핑 |
| **CH03. 프로세스와 스레드 다루기** |
| 01.프로세스 다루기 |
| 02.스레드 다루기 |
| 03.뮤텍스와 세마포 |
| 04.공유 메모리 기반 IPC |
| 05.파이프 기반 IPC |
| 06.시그널 다루기 |
| **CH04. 소켓 프로그래밍** |
| 01.소켓이란 |
| 02.TCP 소켓 다루기 |
| 03.TCP 소켓 프로그래밍 |
| 04.UDP 소켓 다루기 |
| 05.UDP 소켓 프로그래밍 |

CURRICULUM

# [Course 1].

## Part.5
## 데이터베이스

파트별 수강 시간 04:15:14

| |
|---|
| **CH01. 오리엔테이션** |
| 01.데이터베이스 거시적으로 보기 |
| 02.[실습] MySQL 설치 및 설정 |
| **CH02. 데이터베이스 생성 및 테이블 설계** |
| 01.기본 키와 외래 키 |
| 02.데이터베이스와 테이블 생성하기 |
| 03.데이터 입력 _ INSERT |
| 04.데이터 조회 _ SELECT |
| 05.데이터 조회 _ 세부적으로 검색하기 |
| 06.데이터 수정 _ UPDATE |
| 07.데이터 삭제 _ DELETE |
| **CH03. 효율적 쿼리** |
| 01.인덱스 |
| 02.뷰 |
| 03.JOIN |
| **CH05. NoSQL** |
| 01.mongoDB |
| 02.mongoDB 실습 |

CURRICULUM

# [Course 2].

## Part 1. 프로그래밍 패러다임

파트별 수강 시간 23:56:01

| |
|---|
| **CH01. 오리엔테이션** |
| 01.오리엔테이션 |
| 02.개발도구설치 |
| **CH02. 메모리 기반 프로그램의 동작 방식** |
| 01.자바의 구동 원리 |
| 02.자바 컴파일과 실행 |
| 03.Bytecode란 무엇인가 |
| 04.JVM을 통한 메모리 이해하기 |
| 05.코드로 알아보는 JVM의 메모리 |
| 06.코드로 알아보는 JVM의 메모리 2 |
| 07.코드로 알아보는 JVM의 메모리 3 |
| 08.프로세스와 스레드 |
| 09.스레드란 무엇인가 |
| 10.실습을 통한 스레드의 이해 |
| **CH03. 데이터를 바라보는 관점 지향 패러다임** |
| 01.기본데이터와 객체데이터 |
| 02.객체 데이터의 표현 방법 |
| 03.객체 데이터의 모델링 실습 |
| 04.한 개의 데이터 여러 개의 데이터 |
| 05.다양한 자료형의 이해 |
| 06.배열을 이용한 데이터 저장 |
| 07.메서드란 무엇인가 |
| 08.변수와 메서드의 관계 |
| 09.배열과 클래스의 관계 |
| 10.모델(Model)이란 무엇인가 |
| 11.다양한 모델(Model)을 만들어보자 |
| 12.DAO,Utillty 모델 만들기 |
| 13.자료구조 알고리즘의 이해 |
| 14.논리적인 사고력 키우기 기초 |
| 15.논리적인 사고력 키우기 기초 (2) |
| 16.논리적인 사고력 키우기 기초 (3) |
| 17.논리적인 사고력 키우기 중급 |

CURRICULUM

# [Course 2].

## Part 1. 프로그래밍 패러다임

파트별 수강 시간 23:56:01

| |
|---|
| **CH04. 설계관점 객체지향 패러다임** |
| 01.객체지향 프로그래밍 이란 |
| 02.커피 자판기 프로그래밍 |
| 03.라이브러리, 모듈(패키지)의 이해 |
| 04.라이브러리, 모듈(패키지)의 실습 |
| 05.라이브러리(API) 활용하기 |
| 06.라이브러리(API) 활용하기 (2) |
| 07.객체지향 프로그래밍의 설계 |
| 08.객체지향 프로그래밍의 설계 실습 |
| 09.객체지향 프로그래밍의 활용 |
| 10.다향성이론을 활용하자 |
| 11.추상클래스의 설계 |
| 12.인터페이스의 설계 |
| 13.Object클래스의 활용 |
| **CH05. 데이터베이스 패러다임** |
| 01.모델링 중심의 데이터베이스 |
| 02.MySQL을 이용한 데이터 모델링 |
| 03.자바데이터베이스 프로그래밍 변천사 |
| 04.프로그램에 임베디드된 SQL |
| 05.자바 데이터베이스 프로그래밍 |
| 06.SQL과 프로그램의 분리 설계 |
| 07.ORM기반 비 SQL프로그래밍 |

CURRICULUM

# [Course 3].

## Part.1 이산수학

파트별 수강 시간 06:06:07

| |
|---|
| **CH01. 오리엔테이션** |
| 01.오리엔테이션 |
| **CH02. 이산수학과 컴퓨터 공학** |
| 01.100명의 죄수 |
| 02.그래프와 100명의 죄수 |
| 03.늑대, 염소, 양배추 문제 |
| 04.이산수학과 컴퓨터 공학 |
| **CH03. 연속변수와 이산변수** |
| 01.팩토리얼 |
| 02.순열 |
| 03.조합과 도수확률 |
| **CH04. 부울대수와 논리회로** |
| 01.논리게이트 |
| 02.XOR 논리게이트 만들기 |
| 03.베릴로그 |
| 04.덧셈기 |
| 05.4비트 덧셈기 |
| 06.메모리 |
| 07.메모리주소 |
| 08.간단한 컴퓨터 프로그램 |
| **CH05. 그래프** |
| 01.그래프의 정의 |
| 02.그래프의 종류와 정보 |
| 03.그래프의 활용 |
| 04.인접행렬 |
| 05.Walk, Path와 최단경로 |
| **CH06. 트리** |
| 01.데이터 전송과 압축 |
| 02.허프만 코딩 |
| 03.의사결정나무 |
| **CH07. 마르코프 체인** |
| 01.마르코프 체인 |

CURRICULUM

# [Course 3].

## Part.2 자료구조 & 알고리즘

파트별 수강 시간 09:41:59

| |
|---|
| **CH01. 오리엔테이션** |
| 01.오리엔테이션 |
| **CH02. 자료구조와 알고리즘** |
| 01.자료구조와 알고리즘에 대해 |
| **CH03. 리스트와 알고리즘** |
| 01.가장 기본적인 자료구조 리스트 |
| 02.n배열 리스트 |
| 03.연결 리스트 |
| 04.스택 구현하고 활용하기 |
| 05.큐 구현하고 활용하기 |
| **CH04. 재귀 호출** |
| 01.재귀 호출과 문제들 |
| **CH05. 트리와 알고리즘** |
| 01.이진트리란 |
| 02.이진트리 구현하기 |
| 03.이진 검색트리 구현하기 |
| 04.이진 검색트리 구현하기 |
| 05.AVL Tree |
| 06.Red Black Tree |
| 07.B-Tree |
| 08.힙 (Heap) |
| **CH06. 정렬** |
| 01.n *n 으로 풀이되는 정렬 알고리즘 |
| 02.QuickSort |
| 03.MergeSort HeapSort |
| 04.추가적인 자료구조를 활용하는 정렬 알고리즘 |

CURRICULUM

# [Course 3].

## Part.2 자료구조 & 알고리즘

파트별 수강 시간 09:41:59

| |
|---|
| **CH07. 그래프** |
| 01.그래프 구현하기 |
| 02.DFS |
| 03.BFS |
| 04.최소 신장 트리 (Kruskal's Algorithms) |
| 05.최소 신장 트리 (Prim's Algorithms) |
| 06.최단 거리 문제 (Dijkstra's Algorithms) |
| 07.위상 정렬 문제 (Topological Sorting Algorithms) |
| **CH08. 검색** |
| 01.검색이란 |
| 02.해싱과 해시테이블 |
| 03.충돌 |
| **CH09. 동적 프로그래밍** |
| 01.동적 프로그래밍과 재귀 호출 |
| 02.막대 자르기 문제 |
| 03.행렬 경로 문제 |
| **CH10. 그리디 알고리즘** |
| 01.그리디 알고리즘은 무엇인가? |
| 02.그리디 알고리즘 문제들 |

CURRICULUM

# [Course 4].

## Part.1
## 소프트웨어 개발 원칙 및 설계

파트별 수강 시간 03:03:29

| |
|---|
| **CH01. 오리엔테이션** |
| 01. 오리엔테이션 |
| **CH02. 클린 코드** |
| 01. 코드 |
| 02. 함수 |
| 03. 클래스 |
| 04. 모듈 |
| 05. 주석 |
| 06. 테스트 |
| **CH03. 디자인 패턴** |
| 01. Gof 디자인 패턴 |
| 02. CQRS 패턴 |
| 03. Strangler Fig 패턴 |
| **CH04. 아키텍처** |
| 01. 설계원칙 |
| 02. 모노로틱 아키텍처 |
| 03. 레이어드 아키텍처 |
| 04. 클린 아키텍처 |
| 05. 이벤트 기반 아키텍처 |
| 06. 람다 아키텍처 |

CURRICULUM

# [Course 4].

## Part 2. 클라우드 컴퓨팅 & 실습 사례

파트별 수강 시간 05:17:33

| |
|---|
| **CH01. 오리엔테이션** |
| 01.강의 목표 및 강사 소개 |
| **CH02. 클라우드 컴퓨팅이란** |
| 01.클라우드 컴퓨팅 이해 |
| 02.온프레미스와 클라우드 |
| 03.클라우드 컴퓨팅 도입 효과 |
| 04.클라우드 컴퓨팅의 범위 |
| **CH03. 클라우드 컴퓨팅 기술** |
| 01.클라우드 컴퓨팅 기술의 구성 |
| 02.서버 가상화 및 컨테이너 |
| 03.네트워크 가상화 |
| 04.스토리지 |
| 05.데이터베이스 |
| 06.그외 클라우드 컴퓨팅 기술 |
| **CH04. 클라우드 아키텍처** |
| 01.클라우드 아키텍처의 이해 |
| 02.성능 |
| 03.운영 우수성 |
| 04.클라우드 보안과 공동책임모델 |
| 05.가용성 |
| 06.비용 효율화 |
| **CH05. 클라우드 실습** |
| 01.클라우드 실습 개요 |
| 02.클라우드 실습 사례 |

CURRICULUM

# [Course 5].

## Part 1. AWS 클라우드 운영 인프라 이슈 & 개선

파트별 수강 시간 04:05:40

| |
|---|
| **CH01. 오리엔테이션** |
| 01.강의 목표 및 강사 소개 |
| **CH02. 실습 환경 소개** |
| 01.실습 환경 소개 |
| 02.실습 환경 배포 |
| **CH03. 클라우드 사용 및 구성 관련 에러** |
| 01.도메인 에러 |
| 02.Load Balancer 설정 에러 -1 |
| 03.healthcheck 설정 에러 |
| 04.방화벽 설정 에러 |
| 05.Load Balancer 설정 에러 -2 |
| 06.TLS 설정 에러 |
| 07.리스너룰 설정 에러 |
| 08.리소스 권한 에러 |
| 09.클라우드 API 기록 확인 & 자원 1차 정리 |
| **CH04. 클라우드 사용 및 구성 개선** |
| 01.운영 우수성 (operational excellence) |
| 02.보안 (security) |
| 03.가용성 (reliability) |
| 04.성능 효율성 (performance efficiency) |
| 05.비용 최적화 (cost optimization) |
| 06.실습 마무리 |

CURRICULUM

# [Course 5].

## Part 2. 데이터기반 의사결정

파트별 수강 시간 01:37:11

| |
|---|
| **CH01. 오리엔테이션** |
| 01.강의 목표 및 강사 소개 |
| **CH02. 기본 확률 & 통계** |
| 01.기본 통계 |
| 02.기본 확률 |
| 03.데이터 수집하기 |
| **CH03. 알아 두면 좋은 개념들** |
| 01.Trade-off |
| 02.Orthogonal |
| 03.Pareto principle |
| **CH04. 의사 결정 내리기** |
| 01.AB 테스트 |
| 02.실험 |

CURRICULUM

# [Course 5].

## Part 3. 운영 & 모니터링

파트별 수강 시간 01:17:35

| |
|---|
| **CH01. 소개** |
| 01.운영계와 분석계 |
| **CH02. 운영계 사용하기** |
| 01.Debug log |
| 02.Trace |
| 03.monitoring |
| 04.Database 접근하기 |
| **CH03. 분석계 사용하기** |
| 01.필요한 데이터 읽기 |
| 02.대시보드 만들기 |

CURRICULUM

# [Course 5].

## Part 4. 서비스 설계 & 개발 관점 이슈 사례

파트별 수강 시간 02:23:07

| |
|---|
| **CH01. 서비스 운영 관련** |
| 01.조회 요청 시 - Scan 안 하기 |
| 02.조회 요청 시 - index 설정 잘 하기 |
| 03.조회 요청 시 - 불필요한 join 안 하기 |
| 04.쓰기 요청 시 - insert , update, delete |
| 05.쓰기 요청 시 - 일관성 체크하기 |
| 06.기타 이슈들 |
| **CH02. 서비스 확장 관련** |
| 01.동접 처리 - web |
| 02.동접 처리 - CPU & memory |
| 03.동접 처리 - Database |
| 04.많아진 데이터로 발생하는 이슈들 |
| 05.데이터베이스 유지 보수 |
| 06.유지 관리 비용 |
| 07.기타 조언들 |

CURRICULUM

# [Special].

# Part 1. 기술면접 TOP 50

파트별 수강 시간 01:36:10

| |
|---|
| **CH01. 컴퓨터구조 & 운영체제** |
| 01.문제 1~10 |
| 02.문제 11~20 |
| **CH02. 네트워크** |
| 01.문제 21~30 |
| 02.문제 31~40 |
| **CH03. 데이터베이스** |
| 01.문제 41~50 |

CURRICULUM

# [Special].

## Part.2 학교에서 알려주지 않는 개발 부가 지식

파트별 수강 시간 03:24:44

| |
|---|
| **CH01. 버전 관리 - Git** |
| 01. 버전 관리란 무엇인가? |
| 02. Git 기본 명령어 |
| 03. Git 고급 명령어 |
| 04. 실무에서의 브랜치 전략 |
| **CH02. 버전 관리 - Github** |
| 01. Github란 무엇인가? |
| 02. Github 저장소 생성 |
| 03. Github 팀 단위 활용하기 |
| 04. Github Flow |
| 05. Github의 실무 활용 사례 |
| **CH03. 이슈 관리 - Jira** |
| 01. 이슈 관리란 무엇인가? |
| 02. 이슈 관리를 실무에서 사용하는 목적 |
| 03. 이슈 관리의 다양한 제품군 |
| 04. 실무에서 이슈 관리 사례 |
| **CH04. 문서 관리 - Notion** |
| 01. 노션이란 무엇인가? |
| 02. 실무에서의 노션 문서 관리 사례 |
| **CH05문서 관리 - Wiki** |
| 01. 위키란 무엇인가? |
| 02. 위키를 실무에서 사용하는 목적 |
| 03. 문서 관리의 다양한 제품군 |
| 04. 실무에서의 위키 문서 관리 사례 |

CURRICULUM

# [Special].

# Part.3 개발자 AI 업무 도구 활용

파트별 수강 시간 00:53:58

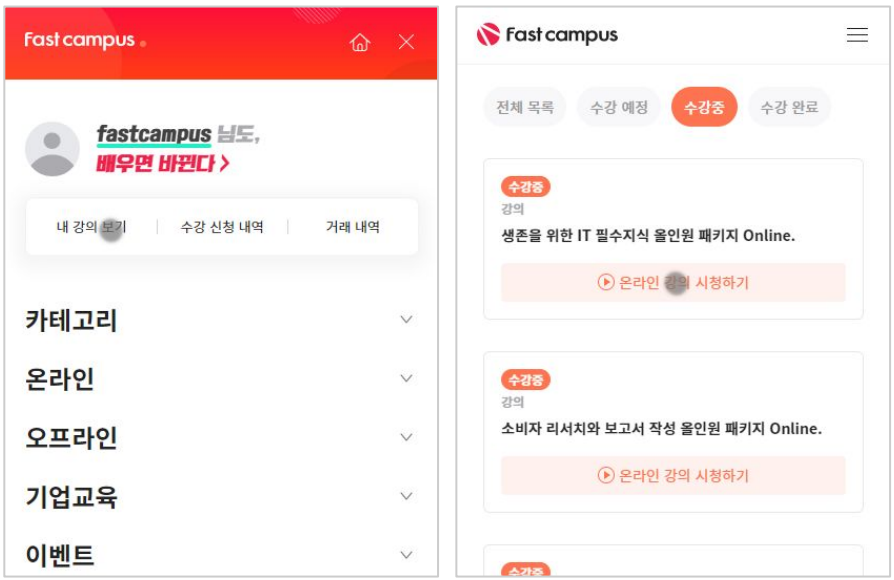
| |
|---|
| **CH01. Chat GPT** |
| 01. ChatGPT란 무엇인가? |
| 02. ChatGPT를 활용한 개발 방법 |
| **CH02. 코파일럿** |
| 01. Github Copilot이란 무엇인가? |
| 02. Github Copilot 기능 살펴보기 |
| 03. 실무에서 Github Copilot 활용하기 |

## 주의 사항

- 상황에 따라 사전 공지 없이 할인이 조기 마감되거나 연장될 수 있습니다.
- 패스트캠퍼스의 모든 온라인 강의는 **아이디 공유를 금지**하고 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<br>환불규정에 따라 환불 가능 |

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


Copyright ⓒ FAST CAMPUS Corp. All Rights Reserved.