

라모스의 추천 도서/자료

안녕하세요. 개발자 라모스입니다. 🙋

제가 코딩과 개발 공부를 처음 시작했던 대학교 2학년 2학기 시절부터 현재까지 어떤 책과 자료를 주로 봤는지 추천하고자 본 멘토링 특강과 별개로 공유해드립니다.

주로 학생 신분부터 주니어 백엔드 개발자가 보면 좋을 기본 자료라고 생각하시면 될 것 같고 굵은 표기는 필독입니다. (특강 대상자를 위해 Docker, Kubernetes, Kafka, DDD, 헥사고널 아키텍처, MongoDB 등과 같이 과도하고 불필요한 주제에 대한 자료는 추천하지 않았습니다.)

저도 추천 도서/자료들을 전부 다 봤다고는 할 순 없고 지금도 시간날 때 공부중이라 ‘개발자가 되려면 저거 전부 다 봐야해?’ 라는 오해는 없었으면 좋겠네요.

개인의 의견일 뿐 무조건 맞다는건 아니고 어디까지나 취사 선택입니다^^;

코딩테스트 준비

- encrypted-def/basic-algo-lecture: 바킹독의 실전 알고리즘 강의 자료
- 프로그래머스 코딩테스트 문제 (Lv 1~3, Java)

CS 기본

- 운영체제 10판
- 그림으로 배우는 HTTP & Network Basic
- HTTP 완벽 가이드
- 자바로 배우는 핵심 자료구조와 알고리즘

사실 전공서적이 가장 좋긴한데, 딱 추천하는 걸로 꼽아달라하면 위 책들은 default로 보셨으면 해요. 특히 운영체제 10판과 HTTP 완벽 가이드는 바이블입니다.

Database

- SQL 첫걸음
- Real MySQL 8.0 (1권)
- Real MySQL 8.0 (2권)
- 업무에 바로 쓰는 SQL 튜닝

순서대로 보시는 걸 추천드리지만, 바이블로 여겨지는 ‘Real MySQL’의 경우 읽기가 굉장히 어렵기도 하지만 백

엔드 개발자가 우선적으로 주로 봐야할 챕터들 위주로 보길 권장합니다.

ex) MySQL 아키텍처(InnoDB storage engine), 트랜잭션과 lock, MySQL 격리 수준, 인덱스, 옵티마이저, 실행 계획, 데이터 타입

Network / Infra

- IT 엔지니어를 위한 네트워크 입문
- 그림으로 공부하는 IT 인프라 구조
- 대규모 서비스를 지탱하는 기술

‘대규모 서비스를 지탱하는 기술’이라는 책은 실제 CS 지식이 개발에서 어떻게 복합적으로 응용되는지 볼 수 있는 책이라 추천했어요.

나머지 두 책은 심도있게 보진 않더라도 큰 흐름을 이해하고 있으면 좋을 것 같아요.

Java

- **Java의 정석**
- **모던 자바 인 액션**
- **이펙티브 자바 3/E**
- 자바 성능 튜닝 이야기
- 더 자바, 코드를 조작하는 다양한 방법 백기선 강의 - 인프런

위 책들은 순서대로 보시되 반복해서 학습하시길 권장합니다. 그룹 스터디로도 강추합니다. 입사 초기엔 Java의 정석으로 억님(CTO님)과 스터디도 했었습니다.

JDK Docs를 1.8, 11, 17 Version을 토대로 어떻게 주요하게 변화했는지 큰 가닥을 잡고 내부 구현 코드에 달린 Javadoc을 꼭 확인하세요. 제가 학생 시절에는 하지 못했지만 현업에 오고나선 자주 해야 했네요.

[JDK 17 Documentation - Home](#)

Spring

- **스프링5 프로그래밍 입문**
- **스프링 입문을 위한 자바 객체 지향의 원리와 이해**
- **스프링 부트로 개발하는 MSA 컴포넌트 (NHN Dooray 부릉 수석님👍)**
- **자바 ORM 표준 JPA 프로그래밍**
- **우아한형제들 최연소 기술이사 출신 김영한의 스프링 완전 정복 로드맵 - 인프런**
- **토비의 스프링 3.1 Vol. 1 스프링의 이해와 원리**

- 토비의 스프링 3.1 Vol. 2 스프링의 기술과 선택
- 스프링 시큐리티 인 액션

개인적으로 NHN 아카데미 합류 전 김영한님 인프런 강의를 다 본 상태로 들어갔어서 백엔드 과정에서 배우고 과제하는 내용을 우선적으로 했었고 그 외 필요 시 위 책들을 보려고 노력했습니다.

부릉님의 저서는 아카데미 시절 저희 기수 내에선 진짜 바이블로 여겨졌었고 제 선배 기수분들도 현업에 가신 뒤로 저 책으로 스터디를 하신걸로 알고 있어요.

토비의 스프링은 쉽게 읽히는 책이 아니라서, 지금도 여유가 있을 때 가끔 조금씩 보곤해요...

개발 시에는 버전에 맞는 **Spring 공식 문서**를 1순위로 확인합니다.

스프링에 대해 어느정도 학습이 됐다면 공식문서를 제일 추천드리는 이유는 당연하게도 스프링 하위의 다양한 프로젝트에 대해 이만큼 제대로 나온 자료가 없고 특히나 Spring Security는 버전 업데이트가 자주 일어나고 그에 따라 코드가 굉장히 달라지기 때문이에요.

하지만 가장 전제는, Java와 OOP, 디자인패턴입니다. 병행해서 학습해야 하고 Spring은 곧 이 세 가지의 집합체라고 비유할게요. Spring 내부 코드가 어떻게 구현되어 있는지 쭉 따라가다보시면 이게 무슨 말인지 이해하실 거라 생각합니다. Java와 Spring은 결국 내부 구현이 어떻게 되어있는지 Javadoc까지 확인하는게 제일 좋은 공부라고 생각하네요.

Backend 개발자가 기본으로 알아야 할 FE

- 모던 자바스크립트 Deep Dive
- 코어 자바스크립트

React나 Vue, TypeScript를 반드시 해야한다라는건 절대 아니고, 백엔드 개발자도 상황에 따라 백오피스(관리자 페이지)나 사내 플랫폼 서비스 개발 시 jsp, thymeleaf를 자주 사용하곤 해서 간단한 JavaScript는 꼭 해보시길 권장드립니다.

Java 외의 한 가지 언어를 반드시 경험해야한다면, JavaScript를 추천드리고 동작 원리를 보시면서 클래스 기반 OOP 언어와는 정말 다른 프로그래밍 패러다임을 느낄 수 있다면 더더욱 좋을 것 같네요.

(ex. 함수형 프로그래밍, 비동기 처리 등)

그 외 취준 ~ 주니어 때 꼭 봤으면 하는 책들

- 객체지향의 사실과 오해
- 오브젝트

- 가상 면접 사례로 배우는 대규모 시스템 설계 기초
- GoF의 디자인 패턴
- 헤드 퍼스트 디자인 패턴
- Clean Code 클린 코드
- 리팩터링 2판
- 읽기 좋은 코드가 좋은 코드다
- 필독! 개발자 온보딩 가이드

GoF를 그냥 보면 진짜 어려워서 ‘헤드 퍼스트 디자인 패턴’을 먼저 보시길 추천합니다.

‘객체지향의 사실과 오해’, ‘오브젝트’는 조영호님의 시리즈인데 순서대로 필독하시길 바랍니다.

‘가상 면접 사례로 배우는 대규모 시스템 설계 기초’는 개발자 스터디에도 주로 등장하는 책이기도 하고 개인적으로 입사 초기 때 주로 보기도 했어요. 제가 리딩하는 스터디 모임에서 다음 스터디 주제를 골라달라고 한다면 저 책이 우선순위 중에 있어요.

‘필독! 개발자 온보딩 가이드’는 제 지인들이 제게 신입으로 첫 출근을 기다릴 때 뭘 공부하며 준비하면 좋겠냐는 질문에 대한 저의 단호한 대답입니다.

‘클린 코드’, ‘리팩터링’, ‘읽기 좋은 코드가 좋은 코드다’ 같은 책들은 아카데미 시절 수업에 나오는 내용이기도 하고 코딩할 때 유의해서 해주면 좋을 것 같아요. 예를 들면, 약간 이상적일수도 있는데 클래스/메소드 네이밍만 보고도 애네가 어떤 역할을 하고 어떻게 동작하는지 한 눈에 알 수 있을 정도로 가독성에 신경쓰면 협업할 때 정말 좋습니다. 개발자 포지션에 관계 없이 습관을 들이면 정말 좋을 것 같네요.

기술서적 외 도움이 많이 됐던 책

- 함께 자라기
- 완벽한 공부법

개발자 이전과 이후 생활 모두 개인적으로 오랫동안 영향을 많이 받았던 책이라 추천드립니다.

검증된 기술 블로그 / Github 링크

국내 테크 기업 공식 블로그

- NAVER D2
- Blog - LINE ENGINEERING
- 우아한형제들 기술블로그
- NHN Cloud Meetup - NHN 기술 블로그

- [당근 테크 블로그](#)
- [kakao Tech](#)

제가 한번씩 궁금할 때 주로 봅니다. NHN Forward, NAVER DEVIEW, 우아한테크세미나, 인프콘 같은 온/오프라인 컨퍼런스는 온라인에 영상이 올라올 때 보시면 좋을 것 같아요.

당장은 무슨 소린지 당연히 이해 안가는데... 공부할때나 현업생활 하다보면 와닿는 순간이 진짜 있어요.

해외 빅테크 기업 공식 블로그

- [Google for Developers Blog - News about Web, Mobile, AI and Cloud](#)
- [LinkedIn Engineering Blog](#)
- [Netflix TechBlog](#)
- [Engineering at Meta](#)
- [The Airbnb Tech Blog](#)

영어 실력이 된다면(?) 시야를 넓히기 위해 어쩌다 한번쯤은 보시길 추천합니다.

우리가 사용하는 대부분의 오픈소스 라이브러리/프레임워크는 이 회사들이 만들었어서 넣어뒀습니다.

개발자 개인/팀 블로그

- [기억보단 기록을](#) (인프런 CTO 향로님 기술 블로그)
- [자바뚱](#) (카카오 개발자님들의 기술서적 관련 스터디 블로그)
- [Yun Blog | 기술 블로그](#) (한국스프링사용자모임에 활동 중이신 현 카카오페이 김남윤님 기술 블로그)
- [스쿠버 다이빙 클럽 - MANTY SCUBA](#) (NHN Dooray 만티 수석님의 개인 블로그)
- [woowacourse-teams · GitHub](#)

우아한테크코스 프로젝트 그룹은 코드, 아키텍처 설계 뿐만 아니라 Github commit, branch 전략 보고 배우면 좋을 것 같네요.