

# AI를 활용한 JAVA기반 DevOps 개발자 양성과정

[ 2026.02.02 ~ 2026.07.10 ]

[8253jang@daum.net](mailto:8253jang@daum.net)

장희정강사

# Artificial Intelligence

# ✓ AI(Artificial Intelligence)란?

## Artificial intelligence

인간의 지능을 모방한 기술

## Machine Learning

데이터를 학습하여 패턴을 발견하고 예측하는 기술

## Deep Learning

인간의 뇌를 모방한 인공신경망을 사용하여 데이터의 패턴을 학습하는 기술

### 인공지능 (AI : Artificial Intelligence)

- 인간처럼 판단·추론·문제 해결을 수행하는 기술
- 규칙 기반 + 학습 기반 모두 포함

👉 "생각하는 컴퓨터"

### 머신러닝 (Machine Learning)

- 데이터로부터 스스로 학습하는 인공지능
- 명시적인 규칙 없이 패턴을 찾아 예측

👉 "데이터로 배우는 AI"

### 딥러닝 (Deep Learning)

- 신경망(Neural Network) 기반 머신러닝
- 대량 데이터로 고정밀 학습

👉 "뇌 구조를 닮은 AI"

# ✓ LLM(Large Language Model)이란?

대규모 언어 모델(LLM) 은

👉 방대한 텍스트 데이터를 학습해 사람처럼 언어를 이해하고 생성하는 AI

- 문장 이해
- 문장 생성
- 질문 답변
- 요약, 번역, 코드 생성

💬 대표 예시

- ChatGPT
- 코드 자동 생성 AI
- AI 챗봇 / 업무 자동화 비서

---

⚙️ 어떻게 동작하나?

- 인터넷·문서·코드 등 대량의 텍스트 데이터 학습
- 다음에 올 단어(토큰)를 예측하며 문장 생성
- 딥러닝(Transformer 구조) 기반

# ✓ LLM발전 흐름

언어 중심 → 지식 중심 → 맥락 중심 → 행동 중심 → 생태계 중심    협력적 인공지능 노드

오늘날의 MCP 개념은 단번에 등장한 것이 아닙니다. 그것은 언어모델의 발전과 함께 점진적으로 형성된 실천적 지식의 축적 결과물입니다

언어 이해기



Fine-tuning

대규모 사전학습 모델을 특정 도메인에 맞게 추가 학습

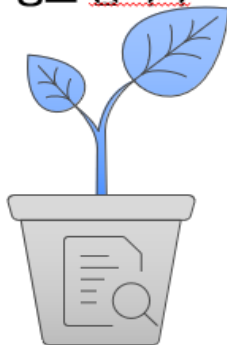
지식 응답자



Prompt Engineering

모델을 재학습하지 않고, 입력 설계로 성능 제어

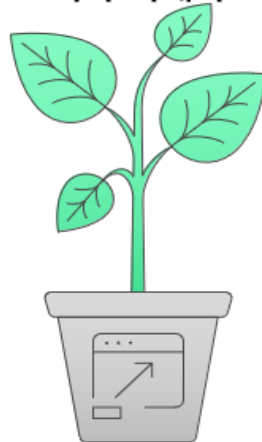
정보 탐색자



RAG

외부 문서나 DB의 정보를 검색해 모델 입력에 포함

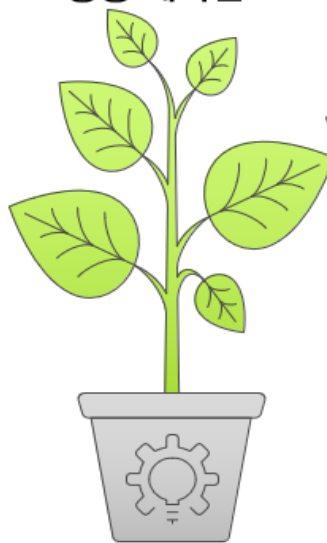
맥락 이해자



Context & Memory

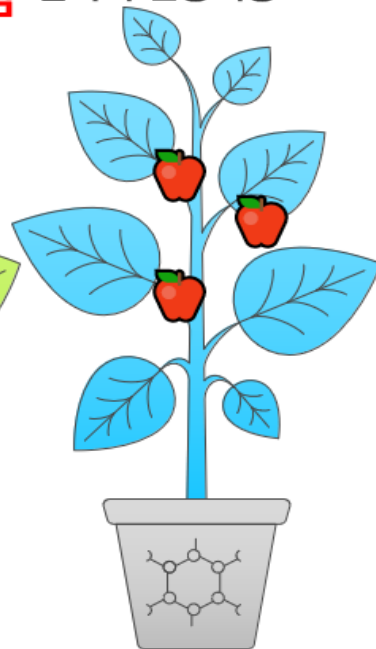
장기 대화 맥락과 사용자 상태를 지속 관리

능동 에이전트



Function Calling

모델이 API/도구를 직접 호출하여 실제 작업 수행



MCP

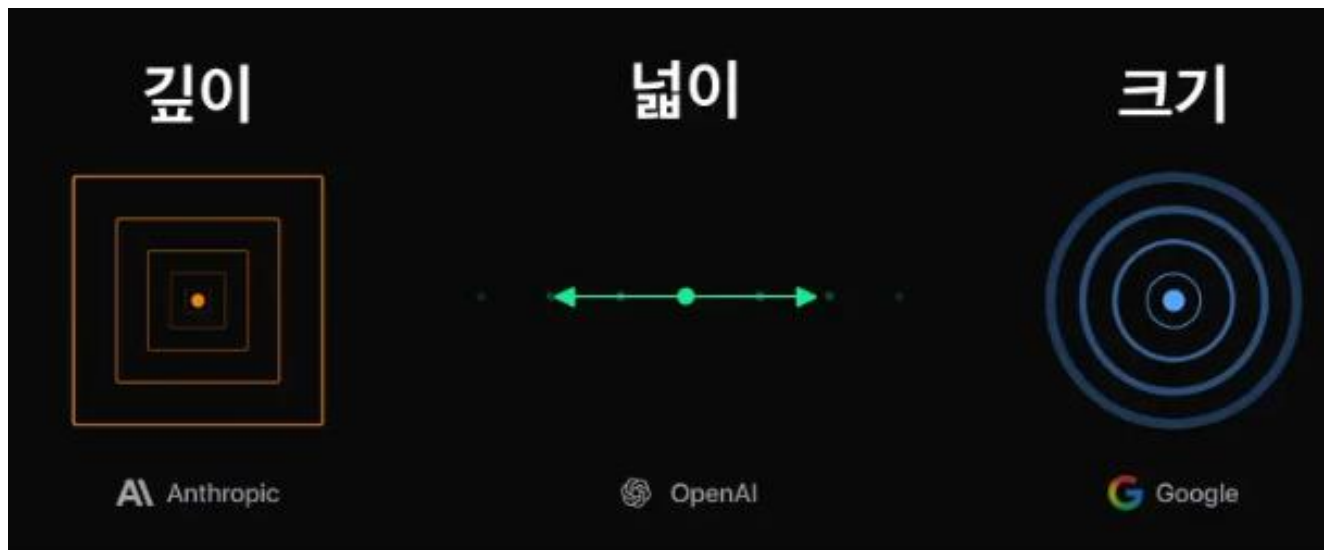
LLM이 다양한 앱·서비스·도구와 표준화된 방식으로 상호작용

## ✓ 주요 AI 기업별 대표 LLM 비교

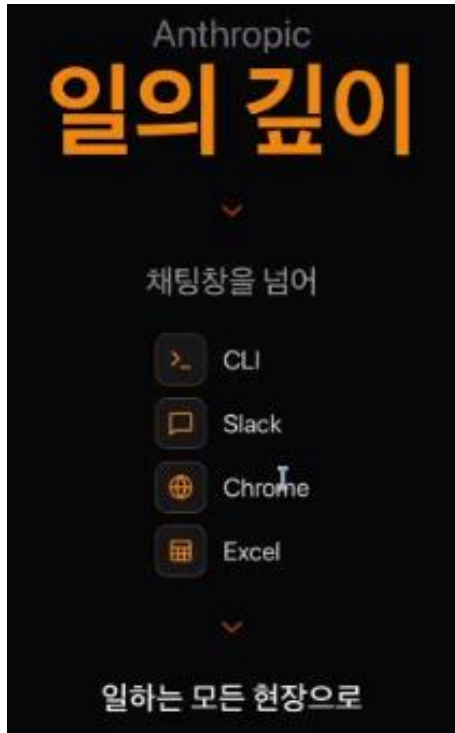
회사	대표 LLM	특징 요약	강점
<u>OpenAI</u>	GPT-4 / GPT-4o	범용 생성형 LLM	코드 생성, 추론, 멀티모달(텍스트·이미지·음성)
<u>Anthropic</u>	Claude 3 (Opus / Sonnet / Haiku)	안전성과 추론 중심 LLM	긴 문맥 처리, 논리적 답변, 안정성
<u>Google</u>	Gemini (Ultra / Pro / Nano)	Google 서비스 통합 LLM	검색·YouTube·GCP 연계, 멀티모달

LLM은 하나가 정답이 아니라, 목적에 따라 선택하는 도구.

## ✓ 주요 AI 기업별 대표 LLM 비교



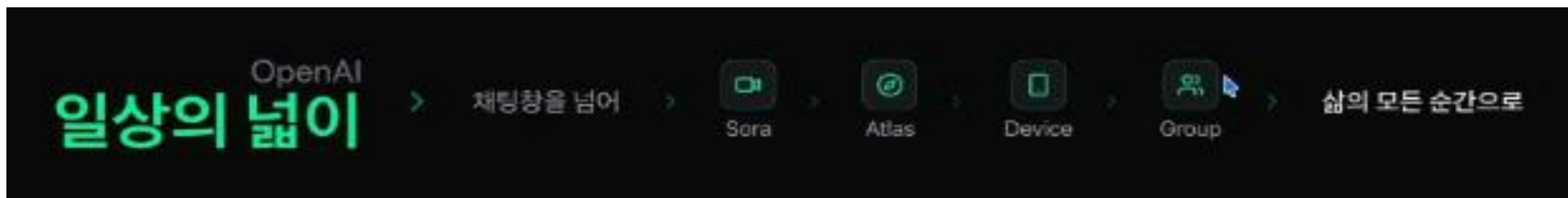
## ✓ 주요 AI 기업 – Anthropic Claude Code / Skills



Anthropic은 일 잘하는 동료  
플랫폼에 종속되지 않고 다양한 툴에서 붙여서  
사용할 수 있도록 발전해왔다.



## ✓ 주요 AI 기업 – OpenAI ChatGPT



OpenAI 는 우리의 일상을 점령!  
영상, 브라우저, 기기, ChatGPT 제품 내 기능 등등,  
AI 시대의 다양한 일상 영역에서 활용될 서비스적인  
시도들이 많았다.

## ✓ 주요 AI 기업 – Google



Google은 다른 회사가 가지고 있지 않는 방대한 데이터를 가지고 있다.

압도적인 모델 발전!

보유하고 있던 Product을 AI로 묶어내면서 판의 크기를 적극적으로 키우고 장악하려 시도했다.

- Gemini(Ultra / Pro / Nano)
- NotebookLM(Google의 문서 기반 AI 노트/리서치 도구)
- Veo(Google의 텍스트 → 영상 생성 AI)
- Nano Banana(AI 이미지 생성·편집 기술)
- Google Antigravity(Gemini 3 기반의 AI 코딩/개발 플랫폼)

## ✓ 주요 AI 기업



서로 다른 경쟁을 가지고 달려온 이들 덕분에  
2025년도에 우리의 일상은 많은 변화가  
있었다.

## ✓ Assistant · Agent · Agentic 비교

구분	Assistant	Agent	Agentic
개념	요청에 응답하는 AI	목표를 받아 행동하는 AI	스스로 목표를 세우고 판단.행동하는 AI
동작 방식	질문 → 답변	목표 → 작업 수행	상황 인식 → 계획 → 실행
주도권	사용자	사용자 + AI	AI 중심
자율성	낮음	중간	높음
기억 / 상태	없음 또는 제한적	상태 유지 가능	지속적 맥락·전략 유지
예시	ChatGPT 답변	배포 스크립트 실행	자동 장애 대응 시스템
역할	도움주는 비서	일하는 직원	스스로 판단하는 시스템

## ✓ Assistant · Agent · Agentic 비교

- Assistant: 코드 설명, 로그 해석
- Agent: CI/CD 자동 실행
- Agentic: 장애 감지 → 원인 분석 → 자동 조치
- **Assistant**: 말해주면 답한다
- **Agent**: 시키면 일한다
- **Agentic**: 알아서 판단한다

“ 우리는 Assistant를 넘어서  
**Agent를 만들고, Agentic 시스템을 설계하는 개발자** “

## ✓ CPU · GPU · TPU 비교

구분	CPU	GPU	TPU
풀네임	Central Processing Unit	Graphics Processing Unit	Tensor Processing Unit
역할	컴퓨터의 두뇌	대량 연산 처리	AI 전용 연산
연산 방식	순차 처리	병렬 처리	행렬·텐서 최적화
강점	범용성, 제어 로직	대규모 병렬 계산	AI 학습·추론 고속
주 사용처	OS, 서버, 앱	그래픽, AI 학습	딥러닝, LLM
AI 활용	제어·서비스 로직	모델 학습	대규모 AI 모델
대표 예	Intel, AMD	NVIDIA	Google

**CPU(범용) : 생각하고 판단 / GPU(병렬) : 많이 계산 / TPU(AI특화) : AI만 빠르게 계산**

# Development + Operations

## ✓ DevOps란?

DevOps는 소프트웨어 개발(Development)과 IT 운영(Operations) 간의 협업과 통합을 촉진하는 **문화이자 방법론**이다.

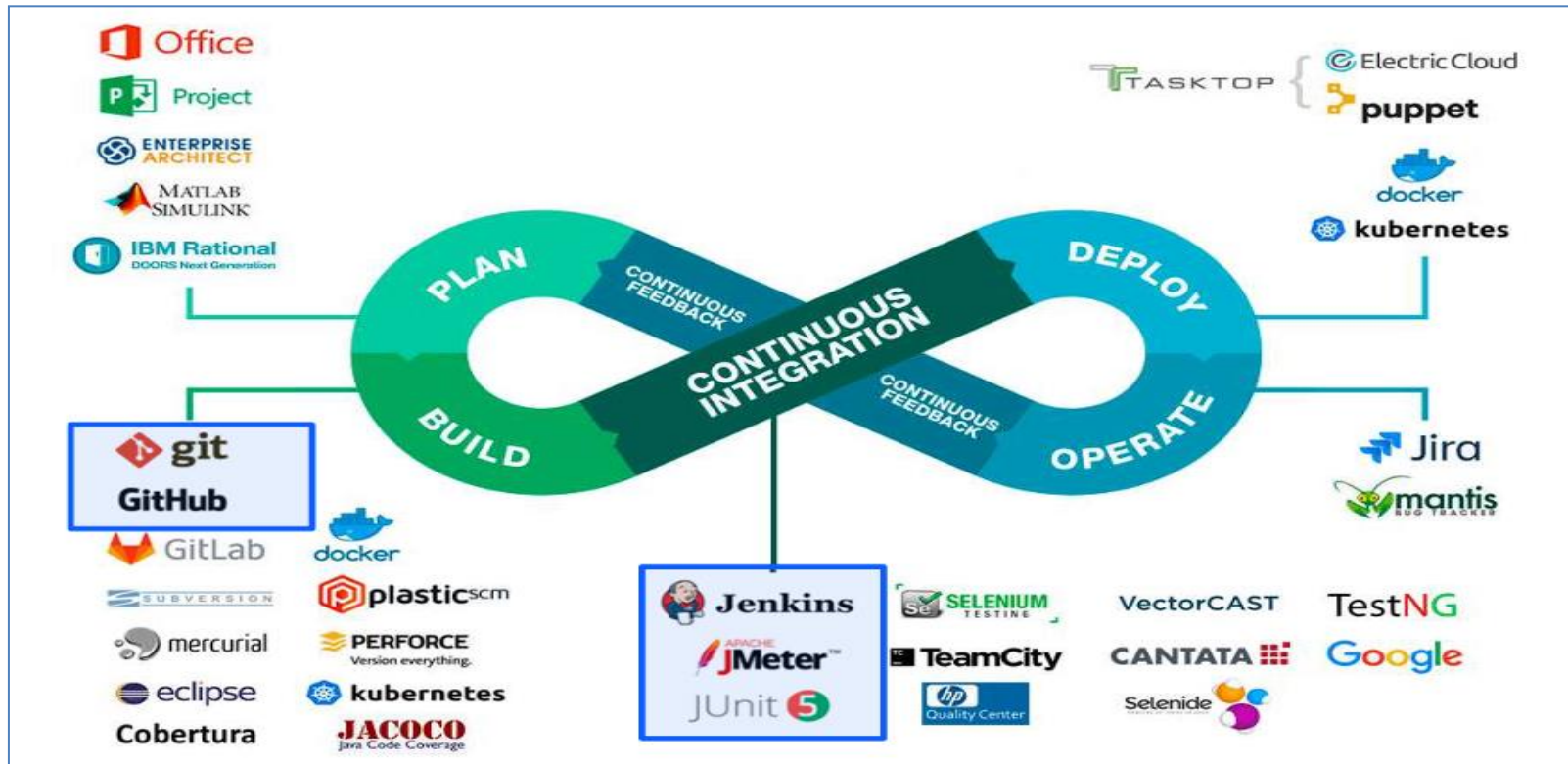
DevOps는 기업의 비즈니스를 지속, 가속 및 성장 시키기 위한 **다양한 디지털 전략의 집합**이다

DevOps를 통해 개발과 운영 간의 협업을 강화하고, 자동화를 도입하여 소프트웨어를 더 빠르게 제공 할 수 있다.

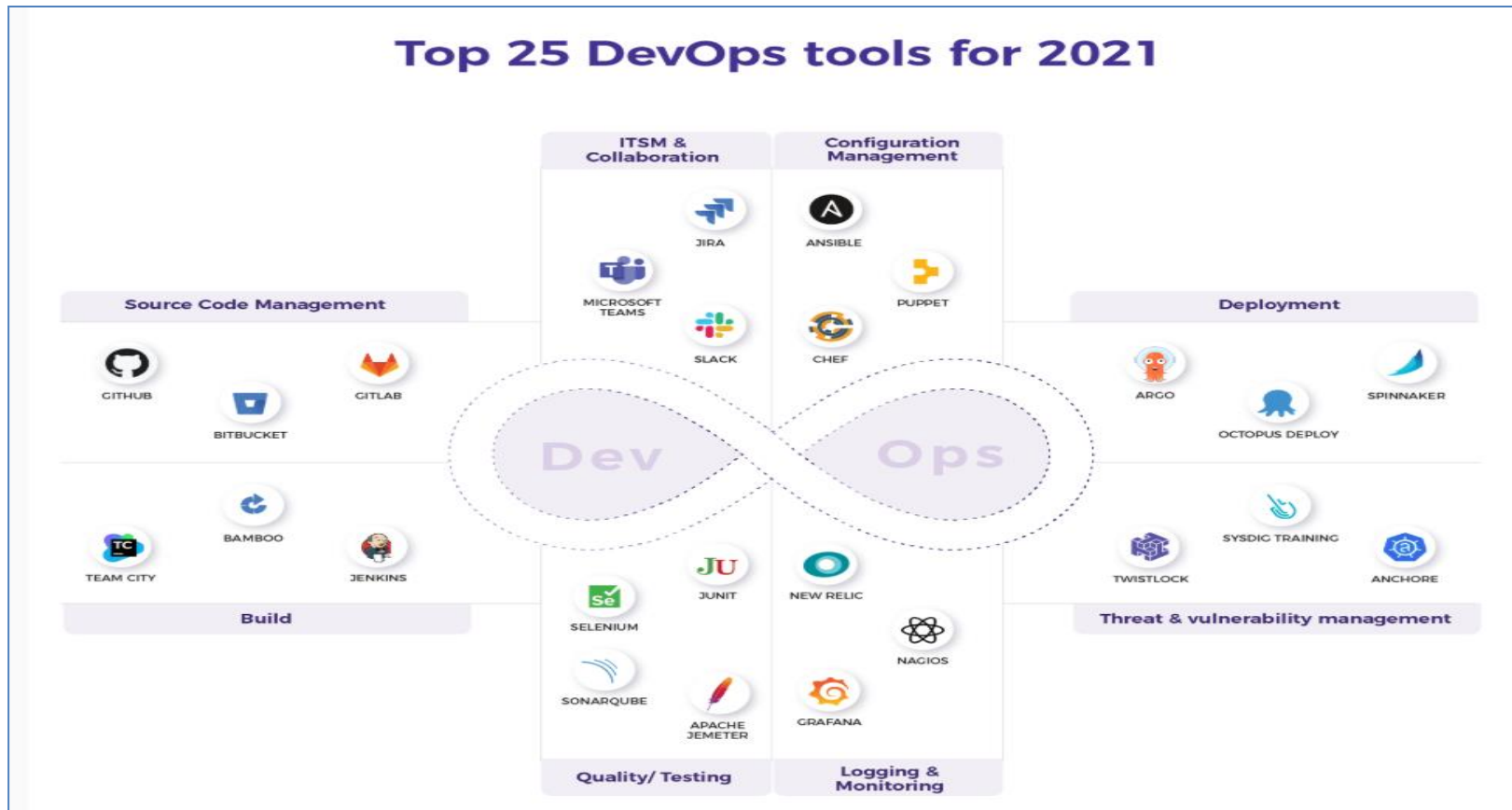
“개발부터 배포, 운영, 모니터링까지를 자동화하고 협업하는 문화와 방법론”



# ✓ DevOps란?



# ✓ DevOps란?



**AI시대 개발자로서?**

## ✓ AI · DevOps 시대, 개발자의 출발점

AI와 DevOps가 기본이 된 시대에서,

**문제를 해결할 수 있는 개발자가 되는 것이 목표.**

이 과정은 기술을 많이 배우는 과정이 아니라,

**어떻게 공부하고, 어떻게 성장할지를 배우는 5개월.**

✓ 프로그래밍의 방향 = 기초 + 활용

AI가 코드를 만들어주는 시대이지만,

**기초가 없는 개발자는 그 코드를 이해도, 수정도 못 한다.**

기본기를 통해 **'판단(검증)할 수 있는 개발자'**가 되는 걸 목표.

기초는 느려 보여도,

가장 빠르게 성장하는 길이다.

.

✓ AI의 방향 = 만드는 사람 ✕, 활용하는 사람 ○

AI 연구원이 아닌,

대신 **AI를 가장 잘 활용하는 개발자**가 되어야 한다.

AI는 코드를 대신 짜주는 도구가 아니라,

**개발자의 생산성을 10배로 늘려주는 도구이다.**

질문을 잘하고, 결과를 검증하고,

AI를 팀원처럼 쓰는 능력이 중요하다.

✓ DevOps의 방향 = 자동화 + 흐름 이해

요즘은 "개발만 잘하면 되는 시대"가 아니다.

Docker, CI/CD를 통해

코드가 어떻게 서비스로 이어지는지

**전체 흐름을 이해하는 개발자가 되어야 한다.**

.

## ✓ AI × DevOps × 개발자의 역할

AI가 모든 걸 대신해 주지는 않는다.

DevOps도 자동으로 모든 걸 해결해 주지는 않는다.

결국 중요한 건

**어떤 문제인지 정의하고, 결과를 판단하는 개발자 자신이다.**

기술은 계속 바뀌지만,

**판단력은 우리의 평생 자산이다.**



## ✓ 신입 개발자의 공부 전략

1 기본기 탄탄

2 AI 활용

3 DevOps 자동화


4 클라우드 경험.

“신입이지만 배포할 수 있는 개발자”,  
“AI를 쓰면서 일하는 개발자”

우리는 완벽한 개발자를 만드는 과정이  
아니라,  
**성장할 수 있는 개발자를 만드는 과정**

# 과정 로드맵

# ✓ AI를 활용한 Java기반 DevOps 개발자 5개월 반 성장 로드맵



	Front-End Developer 사용자와 만나는 화면 구현 API 기반 화면 설계	Back-End Developer 비즈니스 로직 구현 REST API 설계	AI-Powered Developer AI를 “도구”로 활용하는 개발자 생산성과 자동화 중심 사고	DevOps Engineer 배포 · 운영 · 자동화 실무형 인프라 이해
실무통합	<b>Final Project</b> 분석 · 설계 · 구현 · 테스트 · CI/CD · 클라우드 배포 Slack / Notion / Jira / GitHub / Figma			
실무	React Front Semi Project	Spring Framework Spring Boot ORM Back Semi Project	LLM 이해 Spring AI RyanCarson's 3-File System	Linux Docker AWS Terraform GitHub Actions
핵심	HTML · CSS · JavaScript	JDBC · MySQL		
공통	<b>프로그래밍 언어 기본 (공통)</b> Java · Python기초 · SQL · 객체지향 · 로직			

# 과목소개

## ✓ 과목 소개

번호	항목	과목명	일수	시간
1	Basic	Java Programming	10	80
2		MySQL	5	40
3		JDBC API	3	24
4		<b>Java Semi Project</b>	<b>5</b>	<b>40</b>
5	Front End	Html, CSS, JavaScript	5	40
6		React	5	40
7		<b>Front Semi Project</b>	<b>5</b>	<b>40</b>
8	Back End	Spring Framework	5	40
9		Spring Boot	7	56
10		ORM Framework	3	24
11		<b>Back Semi Project</b>	<b>10</b>	<b>80</b>

## ✓ 과목소개

번호	항목	과목명	일수	시간
12	Artificial Intelligence	LLM의 이해	1	8
13		Python기초	1	8
14		Spring AI	5	40
15	DevOps	Docker	3	24
16		AWS	2	16
17		Terraform	1	8
18		Github Actions	3	24
19	협업 및 형상관리	Slack, Notion, Git&Github, Figma , Jira	2	16
20	Final Project	분석, 설계, 구현, 테스트, CI/CD	20	160
21	인성교육	이력서 및 자기소개서	1	8
		포트폴리오	1	8
총일 수(103일)			103	824

# 과목 진행 순서

## ✓ 과목 진행 순서

1 ~ 2개월 (2월 ~ 3월 중순)	Java Basic & Advance	13
	Database 모델링 및 MySQL	5
	AWS RDS	1
	Git&Github	2
	자바콘솔기반의 Semi Project	5
2 ~ 3개월 (3월 중순 ~ 4월 중순)	HTML, CSS , JavaScript	5
	React	5
	Docker	3
	AWS EC2 Docker기반 배포	1
	Front Semi Project	5

4 ~ 5개월 (4월 중순 ~ 5월 말)	Spring Framework	5
	Spring Boot	7
	ORM Framework	3
	Github Actions 활용한 CI/CD	3
	Terraform	1
5 ~ 6개월 (6월 초 ~ 7월 10일)	Back Semi Project	10
	LLM의 이해	1
	Python기초	1
	Spring AI	5
	Final Project	20



## ✓ 알아두기

1. 매일 아침 테스트, 오후 과제 있어요!
2. 오늘의 keyword 준비 및 경청
3. 팀 미션
4. 조 편성 3번 진행 예정(자리배치 변경)
5. 9 to 8

# ✓ 알아 두기

## 6. 수업시간 공유 Drive

[https://drive.google.com/drive/u/0/folders/1wB8X2WjYGZJOwg-jE6z0kbKpnxg7vZ\\_d](https://drive.google.com/drive/u/0/folders/1wB8X2WjYGZJOwg-jE6z0kbKpnxg7vZ_d)

## 7. 수업시간 소통 Slack

<https://app.slack.com/client/T0ABTADV6DV/C0ACMMDA43B>

## 8. 수업교안 Notion

[https://www.notion.so/KOSTA\\_306-\\_AI\\_Java\\_DevOps-2faa7a6c42ce804b8cb9fd70c4f38fae](https://www.notion.so/KOSTA_306-_AI_Java_DevOps-2faa7a6c42ce804b8cb9fd70c4f38fae)

## ✓ 우리에게 필요한 것

- 메타인지(Meta-Cognition)를 잘 하자.

: 프로그래밍 능력 + 발표능력 + AI도구 활용 + 소통 + 팀워크

- 개인역량을 향상 시키고 올바른 생활습관 길들이기
- 지식과 경험 나누기
- 서로 격려하고 지원하는 학습 커뮤니티 형성

## ✓ 학습자세

1. 복습하자.(반복/반복 무한 반복이 최고!)

2. 지각, 조퇴, 결석 하지 말자.

3. 노트정리하자.

4. 서로 도와 가며 공부하자.

5. 목표와 꿈을 확인하자.

6. 자기주도학습을 잘하자.

7. 항상 청결 유지

## ✓ 수업시간에는

1. 인터넷 쇼핑 NO!

2. 채팅 NO!

3. 휴대폰 사용 NO!

4. 게임, 주식 NO!

5. 서로 존중하게 대화법 OK

## ✓ 과정 기간 동안

롤프 메르클레

천재는 노력하는 사람을 이길 수 없고,  
노력하는 사람은 **즐거는** 사람을 이길 수 없다.

성공은 **노력**의 과정이다!

## ✓ Algorithm Site

- 정올

<http://www.jungol.co.kr/>

- 백준

<https://www.acmicpc.net/>

<https://solved.ac/>

- 프로그래머스

<https://programmers.co.kr/>

- SWEA

<https://swexpertacademy.com/>

## ✓ 자격증

- SQLD

<https://www.dataq.or.kr/www/main.do>

- 정보처리

<https://www.q-net.or.kr/crf005.do?id=crf00503&jmCd=1320>

- AICE(AI Certificate for Everyone)

<https://aice.study/main>