

# LLM 마스터클래스 #1

AI 서비스 개발부터 운영까지  
- 강의 시작에 앞서



## Part 1-1.

# 강사 소개, 강의 내용 소개



## 강사소개

# Hanna Kroukamp



## 강사소개

# Hanna Kroukamp

## Senior Data Scientist @Microsoft

- Business Applications and Platform group - Business Copilot AI Team
- IT 에서 24년차 - dev, data engineer, project manager, data scientist
- 남아공 - 영국 - 미국
- 현재 Copilot Studio 를 위한 생성형 AI 기반 기능 개발중



무엇을 배우나요

코스 내용 소개



# 거대 언어 모델

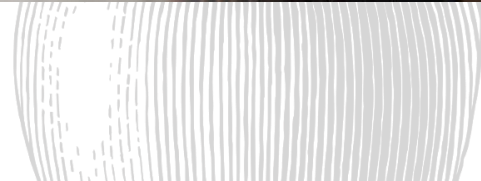
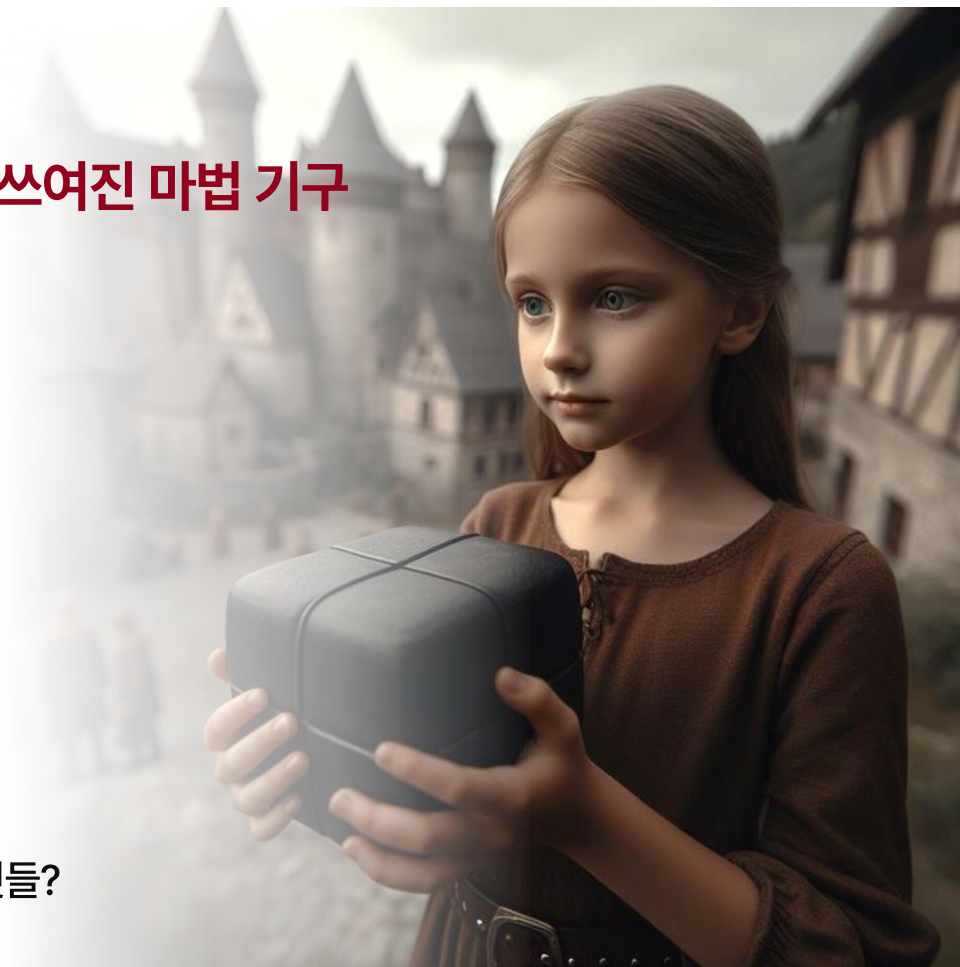
- 간략한 AI 역사
- 챗GPT 전에 AI는?
- 생성형 AI/ 언어 모델은 왜 다른가? 무엇이 바뀌는가?
  - 인풋/아웃풋
  - 일의 수행
- 빨리 바뀌지 않을 이유?
- 우리가 배워야 할 이유?



# 코스 내용 소개

## 어느날 우리 마을에 나타난, LLM 이라 쓰여진 마법 기구

- 그대로 써보기 (챗GPT), 프롬프트 엔지니어링
- 어디에 연결 된 거야?
  - 분해 /해킹 해보기
  - 내 코드 연결 해보기
  - 기본적인 웹 아키텍처, 프로그래밍 API 이해
- 좀 더 잘 써먹기 위한 여러가지 방법들
  - RAG, langchain, plugin, tools
- 이것으로 시스템을 만들어보자
- 좀 더 효과적으로? 저렴하게?
- 본격적으로 시스템을 만들려면 걱정해야 하는 것들?



# 코스 내용 소개

## 언어 모델, 어떻게 쓰지?

**Part 1,2** 이거 뭐야? 그대로 써보기 (챗GPT), 프롬프트 엔지니어링

---

**Part 3** 어디에 연결 된 거야? 해킹해보자

---

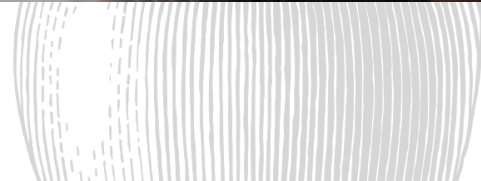
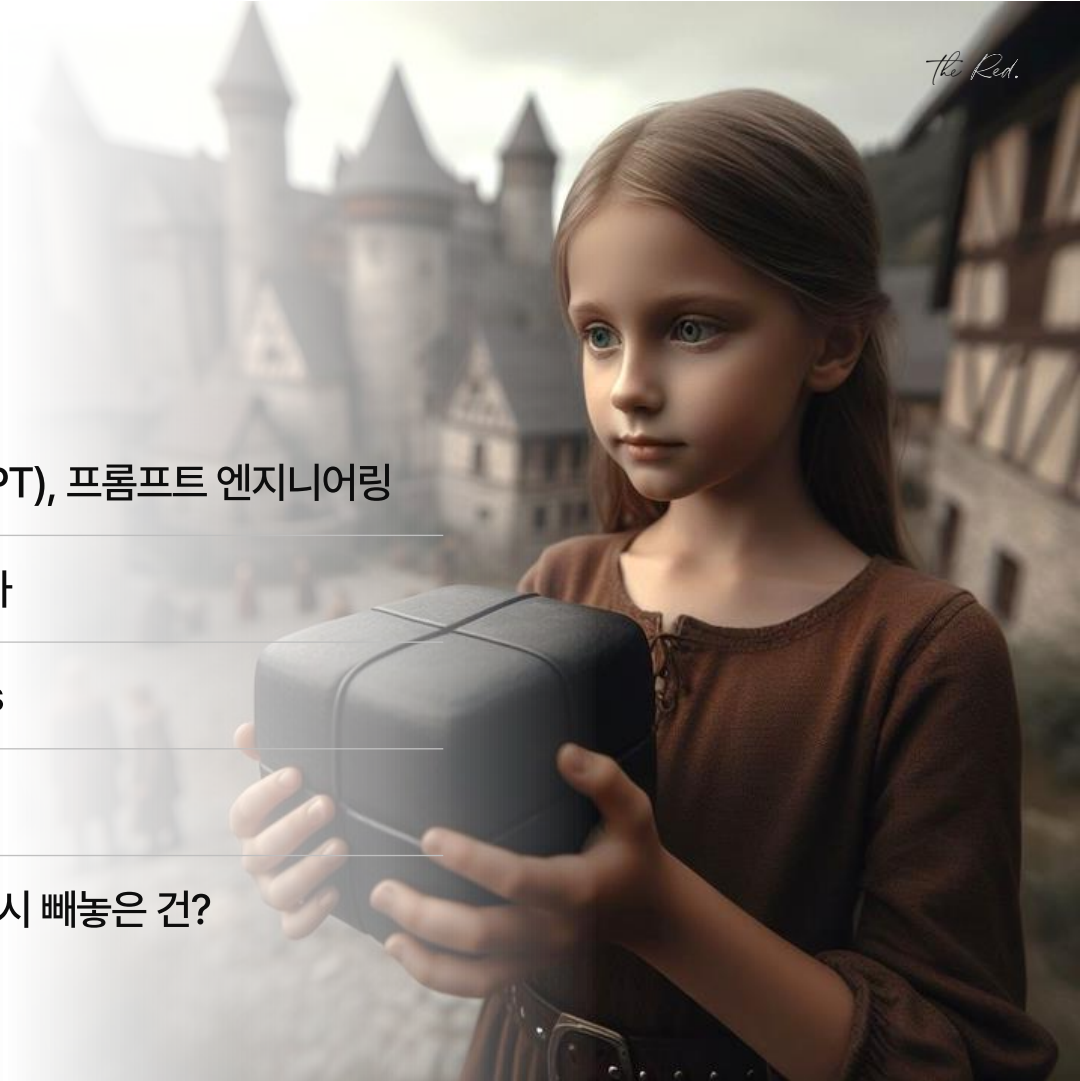
**Part 4** RAG, langchain, plugin, tools

---

**Part 5** 이것으로 시스템을 만들어보자

---

**Part 6** 좀 더 효과적으로? 저렴하게? 혹시 빼놓은 건?





# Part 1-2. LLM overview



## 거대 언어 모델 소개

# 역사, 변화, 그리고 생성형 AI 의 차이점



# 짧은 역사



1950

## Birth of AI

Alan Turing and the Turing test

Arthur Samuel - program to play checkers. First to learn the game independently.

John McCarthy - coined "AI"



1970

## AI Maturation

John McCarthy - LISP. Used for AI research

Arthur Samuel: "machine learning"

First working robot at GM

Stanford Cart

AAAI founded



1980

## AI Winter

Low interest in AI

Some developments 1990 - 2010



2010

## AGI - New Boom

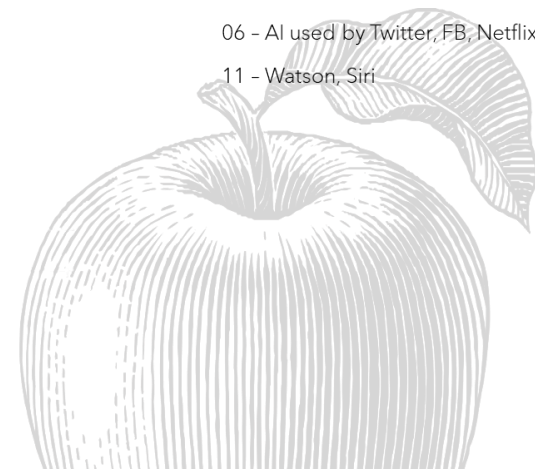
97 - Deep Blue

97 - Speech recognition

02 - Roomba

06 - AI used by Twitter, FB, Netflix

11 - Watson, Siri



# 챗GPT 이전의 AI

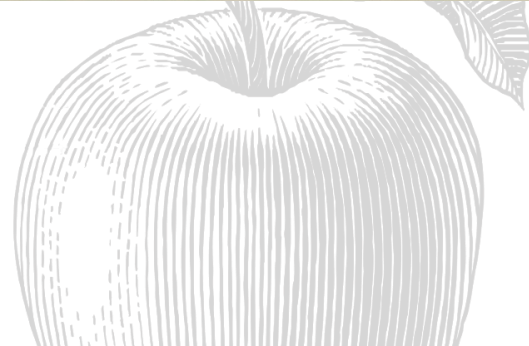
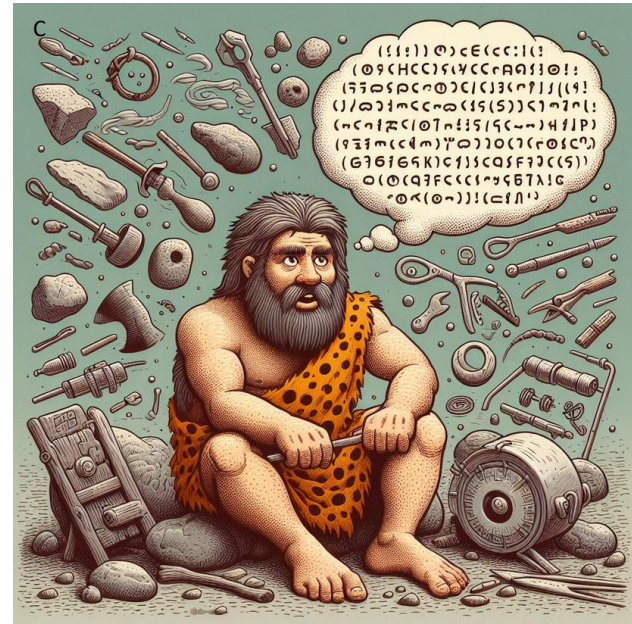
## 생성형 AI 이전에도 AI에 엄청난 발전이 있었음

스팸 필터	Naive Bayes 모델
추천 시스템	넷플릭스, 유튜브, 아마존과 같은 플랫폼들은 사용자의 이전 활동, 검색 기록 및 선호도를 분석하여 개인 맞춤형 추천을 제공하는데 AI를 사용
음성 인식	시리(Siri), 구글 어시스턴트(Google Assistant), 알렉사(Alexa)와 같은 음성 인식 기반의 개인 비서는 사용자의 명령을 이해하고 반응
자율 주행 자동차	테슬라와 같은 회사에서 자율 주행 기술 개발에 AI를 사용
의료 진단	인공지능은 의료 이미지 분석(예: X-레이, MRI)을 통해 질병을 진단하고, 환자 데이터를 분석하여 치료 계획을 제안하는 데 사용
금융 서비스	AI는 신용 평가, 주식 시장 분석, 사기 탐지 등에 사용
광고	페이스북, 인스타 등에서의 맞춤 광고
게임	AI는 게임 내 NPC의 행동을 제어하거나, 게임 플레이어의 난이도를 조절



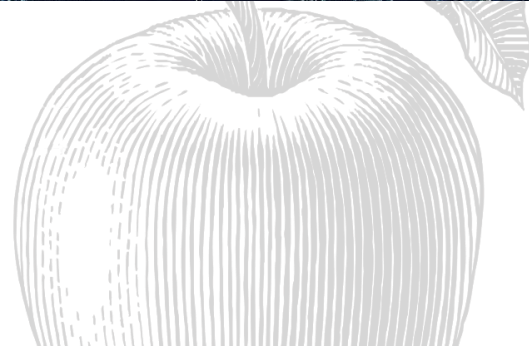
# 왜 언어 모델은 다른가

- 사람이 사용해온 도구들  
: 손도끼, 돌팔매, 자전거, 냉장고, 노트북, 자동차...
- 무엇을 쓸 지 생각하고 결정한 다음,  
그 의도에 부합하는 도구를 선택함.



# 왜 언어 모델은 다른가

- 컴퓨터 프로그램 / 웹사이트 -
  - 여전히 나의 “의도”를 이해할 수 없는 도구들
  - 여전히 사람이 언어로 사고하고 결정한 다음에 선택해야 함

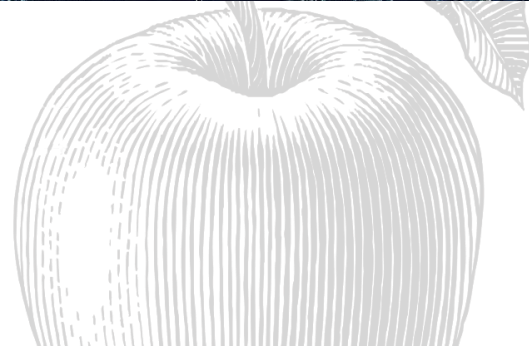




# 왜 언어 모델은 다른가

## 언어 모델

- 나의 의도를 이해하는 도구
- 나의 의도를 이해하고, 나의 실행방식과 도구를 이해하며, 알아서 해결할 수 있음
  - “초인종 울리면 확인하고 택배면 문 열어주고 옆집이면 문 열어주지 마”
  - 언어의 컨텍스트 이해, 이미지와 비디오 이해.
  - “이 400 페이지 문서에서 유튜버들 세금 매기는 거에 관한 내용 있으면 뽑아봐”
- 설명할 수 있으면, 현존하는 도구로 사람이 실행할 수 있는 임무면 실행할 수 있음



## 무엇이 바뀌는가 – 인풋 이해

### 사람이 필요했던 이유

- 언어의 이해, 언어로 발표, 언어로 검색, 언어로 판단
  - “진상민원” 이해 및 해결 → 컴퓨터 시스템
  - 6.25 시절 이야기까지 들어줘야 하는 종업원 → 주문 시스템

### UI디자인이 필요했던 이유

- “네가 뭘 좋아할지 몰라서 다 주문했어” 식의 메뉴
- 사용자가 의도에 부합하는 메뉴가 있어야 함.

### 검색이 완전하지 않았던 이유

- 사용자가 뭘 해야 할지 잘 모르면 어려움
  - KTX 를 모르는 외국인이 한국 사이트에서 TGV 로 검색
  - “떡볶이” 대신 “쫄깃 매운 음식”으로 검색
- 지금: “대강 비슷한 결과”도 상관 없음





## 무엇이 바뀌는가 – 일의 실행

사람이  
필요했던 이유

- 결과를 설명해줘야 함 - 진상 민원 대응, 문의 해결
- 프로세스를 설명해야 함
- 이전 케이스를 참고해야 함

지금까지의  
대인 인터페이스  
전체가 바뀌게 됨

- 의도 이해, 결과 및 프로세스 설명에 **사람이 필요하지 않음**
- 아웃풋을 시스템 아웃풋으로 바꾸어 **전면 자동화가 가능**
- **설명할 수 있으면, 현존하는 도구로 사람이 실행할 수 있는 임무면**  
실행할 수 있음



# 무엇이 바뀌는가 – 인간 명령어를 시스템어로 통역

가장 폭발적인 변화

- 인간적으로 이해하고 기계에게 명령가능
- 현존하는 모든 기계를 사람처럼 부릴 수가 있음



- OpenAI 의 Figure 1 로봇
- 사람의 의도를 이해하고, 로봇의 인터페이스로 번역하여 명령를 실행하고 설명도 가능함
- <https://www.youtube.com/watch?v=GiKvPJSOUmE>



## 지금 기업에서 가장 빨리 바뀌는 부분

- 모든 시스템의 UI
- 모든 시스템의 UI 방식이 바뀜 - 자연어
  - UI 흐름 디자인이 단순해지고 복잡해짐
- 

- 모든 시스템의 자연어 처리
- 기계어 **output** - 로그에서 API call
  - 자연어 **input** - 일처리 결과를 전화로
- 



## 개인적인 변화?

### 개인 생산성

- 언어 장벽 없어짐
- "생성형" 노력이 쉬워짐
- 신입이 필요 없어짐
- 내용 요약 정리가 편리해짐

---

### AI는 이 시대의 "전기"

- 무엇을 하던지 **알아야 함**
- 무엇을 하던지 **배워야 함**
- **생산성의 차이**가 10-50% 가 아닌, 수십배, 수백배로 벌어질 수 있음



## 빨리 바뀌지 않는 이유

- 코딩은 여전히 장인이 한땀 한땀...
- Hallucination
- 시스템 통합은 멀고도 험난
- AGI 는 멀었다



## 아주 빨리 바뀌는 부분

### 개인 생산성

- 언어 장벽 없어짐
- "생성형" 노력이 쉬워짐
- 신입이 필요 없어짐
- 내용 요약 정리가 편리해짐

### AI는 이 시대의 "전기"

- 무엇을 하던지 **알아야 함**
- 무엇을 하던지 **배워야 함**
- 생산성의 차이가 10-50% 가 아닌, 수십배, 수백배로 벌어질 수 있음



# 코스 내용 소개

# 코스 내용 복습



# 코스 내용 소개

## 어느날 우리 마을에 나타난, LLM 이라 쓰여진 마법 기구

- 그대로 써보기 (챗GPT), 프롬프트 엔지니어링
- 어디에 연결 된 거야?
  - 분해 /해킹 해보기
  - 내 코드 연결 해보기
  - 기본적인 웹 아키텍처, 프로그래밍 API 이해
- 좀 더 잘 써먹기 위한 여러가지 방법들
  - RAG, langchain, plugin, tools
- 이것으로 시스템을 만들어보자
- 좀 더 효과적으로? 저렴하게?
- 본격적으로 시스템을 만들려면 걱정해야 하는 것들?





**Part 1-3.**  
**단어, 컨셉 정리**



# 단어, 컨셉 정리

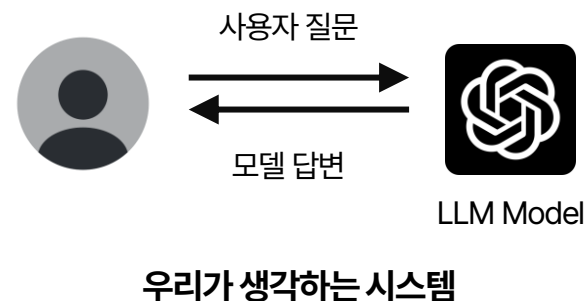
## 앞으로 계속 나올 용어와 컨셉 노트



## 언어 모델 족보 잠깐 정리

LLM	태블릿
아이패드 시리즈	GPT 시리즈
제일 유명한 아이패드 모델	ChatGPT
"아이패드는..."	태블릿을 의미할 때가 많음
다른 브랜드 태블릿도 많음	

- 다른 LLM 을 쓸까요?
- 가능함
- 삼성 갤럭시 태블릿 등등...



# 시스템 용어 잠깐 정리

Prompt Engineering

AI

Model / Foundation Model

Pre-training

Fine-Tuning

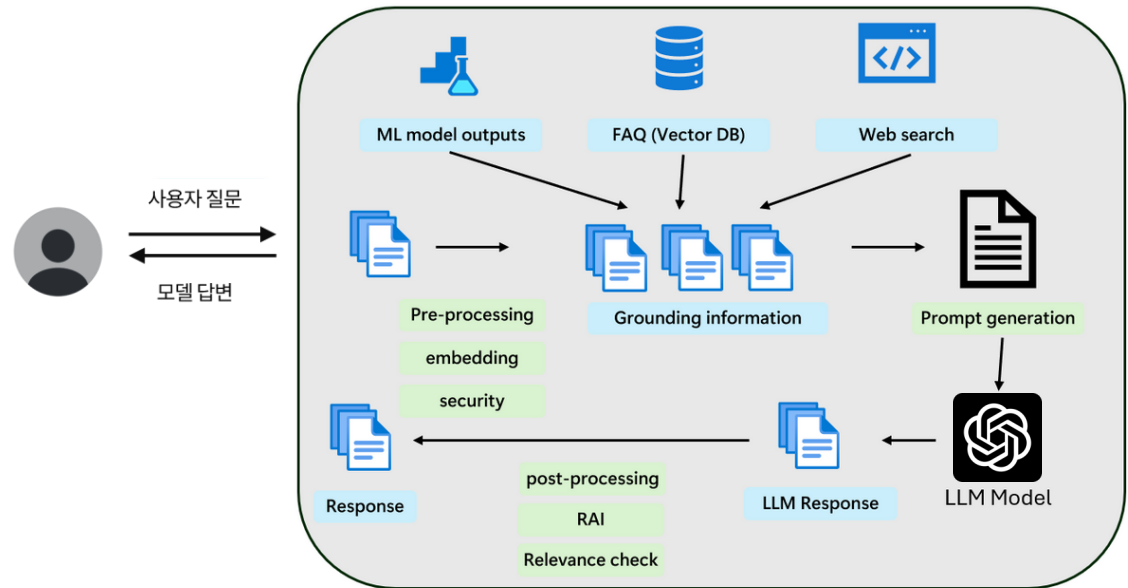
Deploy

API / "serving" the model

Post/pre processing

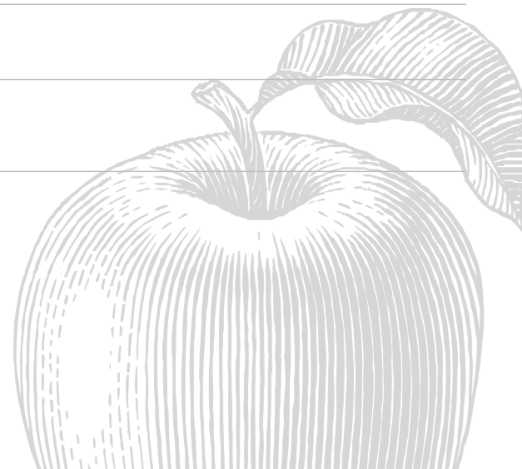
Embedding

LLM Ops



## 시스템 용어 잠깐 정리

<b>Prompt Engineering</b>	지시 똑바로 잘하기
<b>AI</b>	지난 데이터로 패턴을 공부해서 상황 딱 보면 판단 나오게 함
<b>Model / Foundation Model</b>	"판단/예측하는 기계", 기본 언어 모델, 튜닝 가능함
<b>Pre-training</b>	기본 공교육
<b>Fine-Tuning</b>	직업 교육, 연수원
<b>Deploy</b>	파견 / 배치
<b>API / "serving" the model</b>	파견된 직원이 창구에서 임무 수행함
<b>"inference", "prediction"</b>	판단 / 예측 / 답변
<b>LLM Ops</b>	식당 운영



**Part 1-4.**

**강의 시작에 앞서...**



강의 시작에 앞서...

**현재 트렌드, 분위기, 무엇을 배울까?**



## 시작하기 전에...

- AI 업계의 변화 속도
- 오늘 배우는 것을 내일 써먹을 수 있을까?
- 그냥 좀 기다렸다가 어느 정도 안정 되고 나서 쓰면 안 될까?
- 이거 배워서 ML/AI 전문가로 취업할까?  
철도 모델 이야기
- 그럼 무엇에 포커스를 두고 배울까?
- 뭐든 만들어보자!

안됩니다

딥러닝? 수학?

여러가지 모델??

