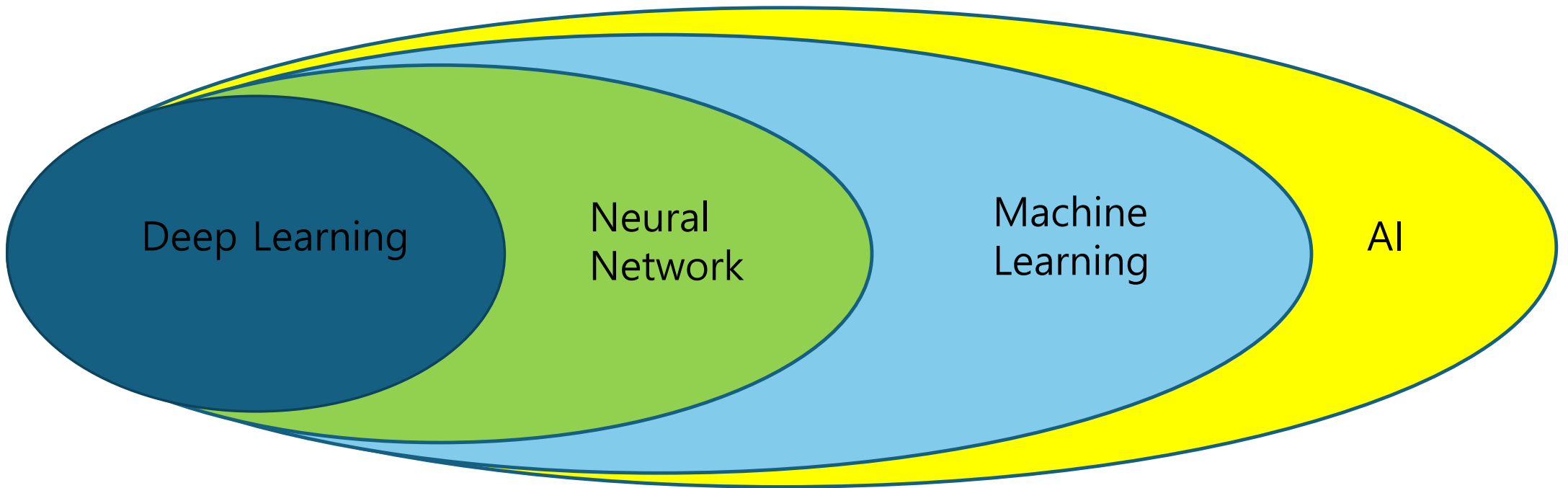
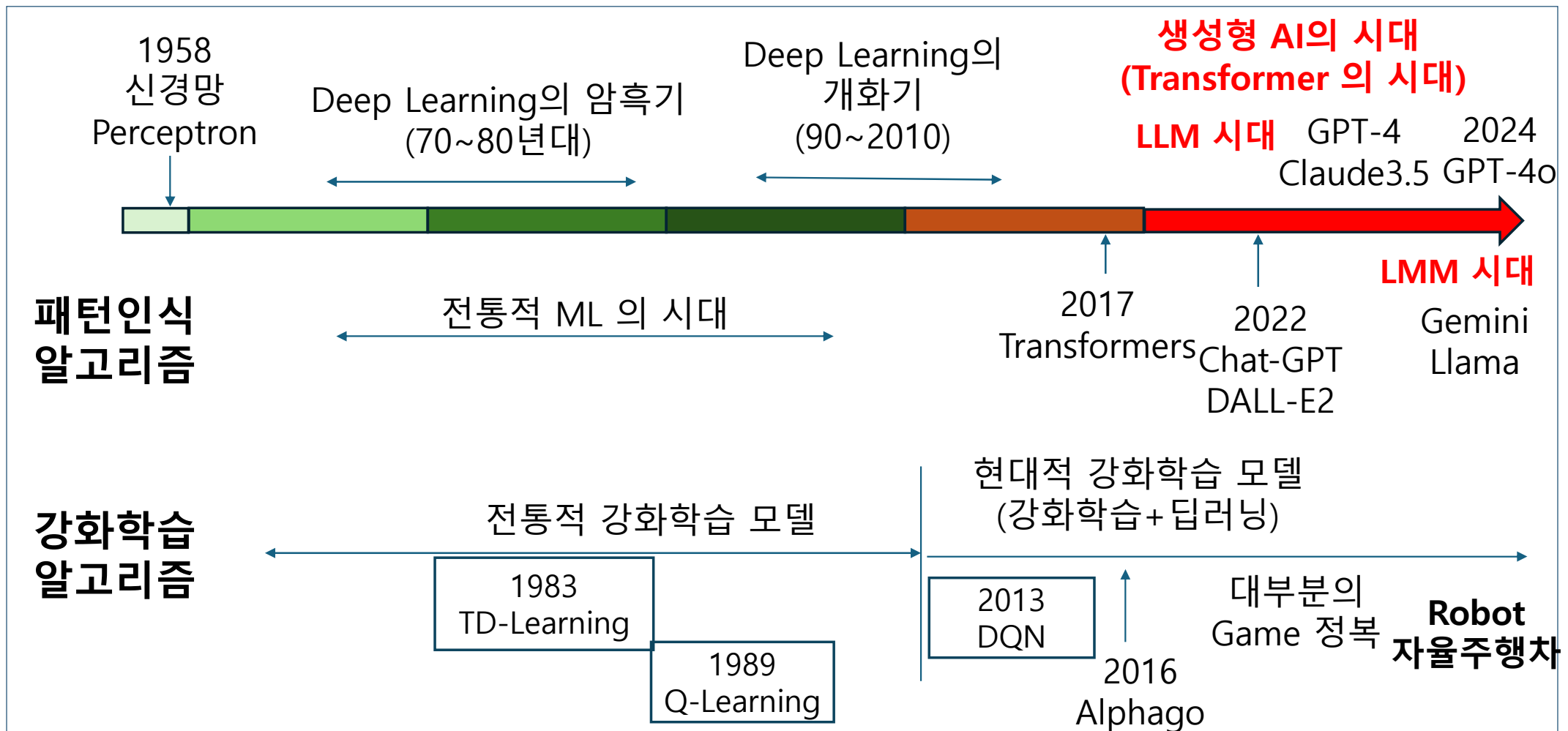


# AI 소개

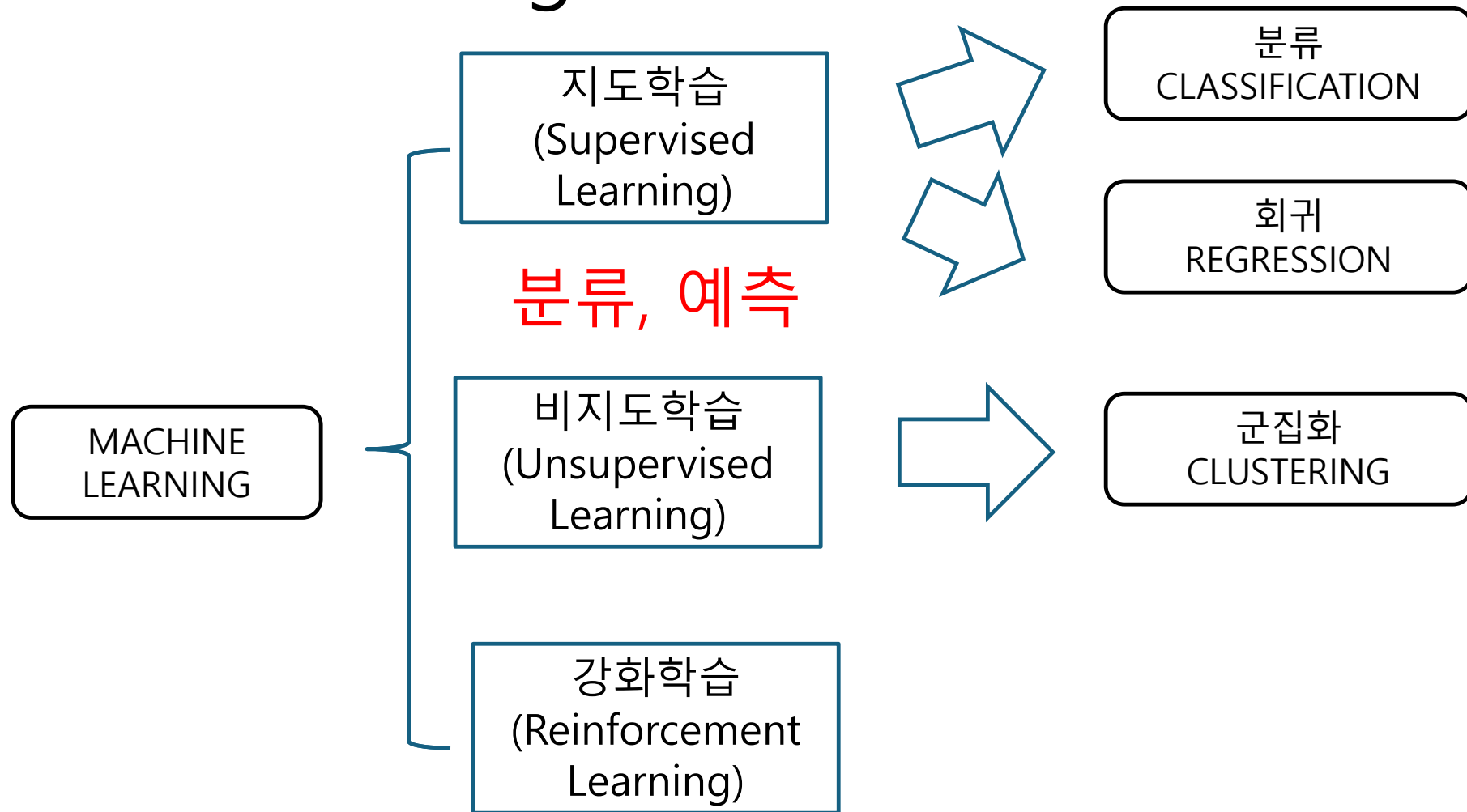
# AI vs. Machine Learning vs. Deep Learning



## AI 발전 History



# Machine Learning 의 종류



# Scikit-Learn



전통적 Machine Learning Tool:

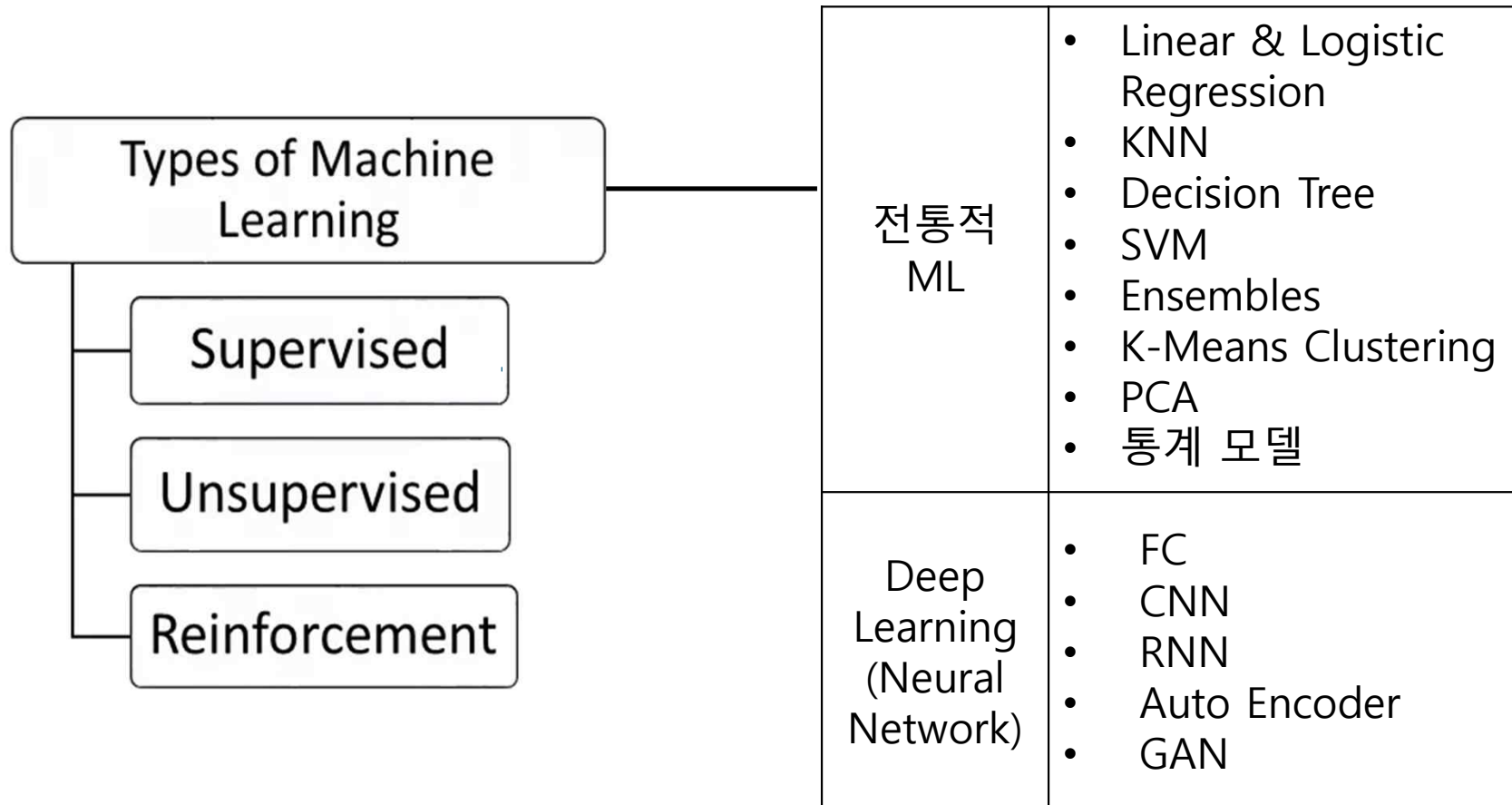
- 벤치마크용 데이터셋 예제
- 데이터 전처리(preprocessing)
- 지도 학습(Supervised learning)
- 비지도 학습(Unsupervised learning)
- 모형 평가 및 선택 (evaluation and selection)

# Tensorflow, Pytorch

Deep Learning Tool:

- Pre-train model, Sample Dataset 제공
- Deep Learning 에 필요한 각종 함수 제공
- GPU support
- 각종 language 지원 API 제공

# Machine Learning 기법의 종류



# Univariate Linear Regression (단변수선형회귀)

$$y = a + bx \rightarrow \text{Hypothesis (가설)}$$



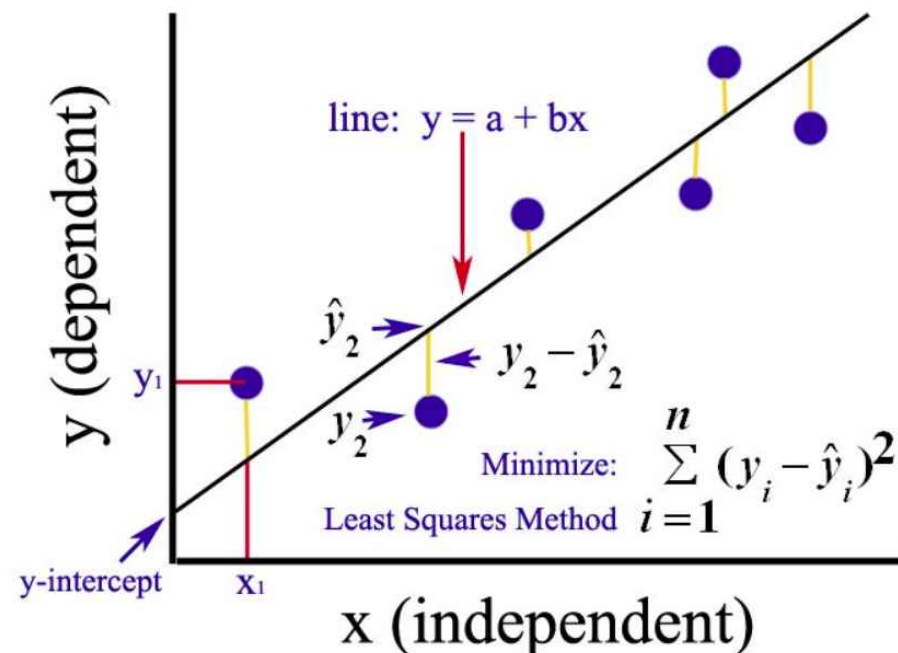
- OLS (Ordinary Least Squares, 최소자승법)

$$\text{Minimize } \sum_{i=1}^n (\text{true} - \text{prediction})^2$$




**Cost Function (비용함수)**

- 가설이 얼마나 틀렸는지 측정






# Anaconda 설치

 [Products](#) [Solutions](#) [Resources](#) [Partners](#) [Company](#) [Sign Up](#) [Sign In](#)


## Download Now


For installation assistance, refer to [Troubleshooting](#).


Download Anaconda Distribution or [Miniconda](#) by choosing the proper installer for your machine. Learn the difference from our [Documentation](#).



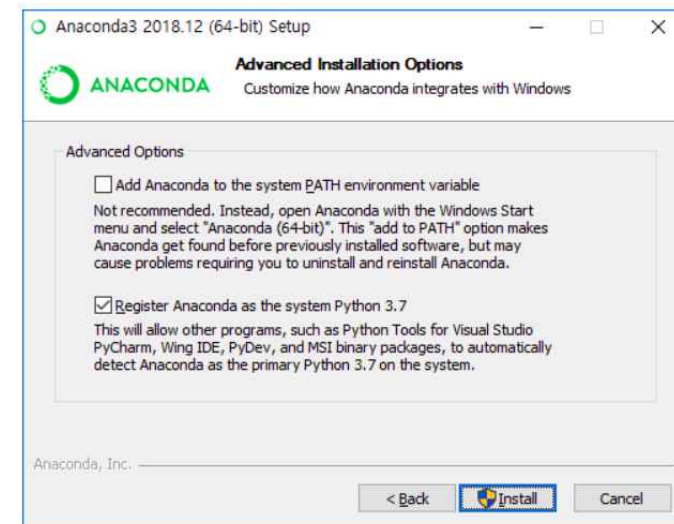
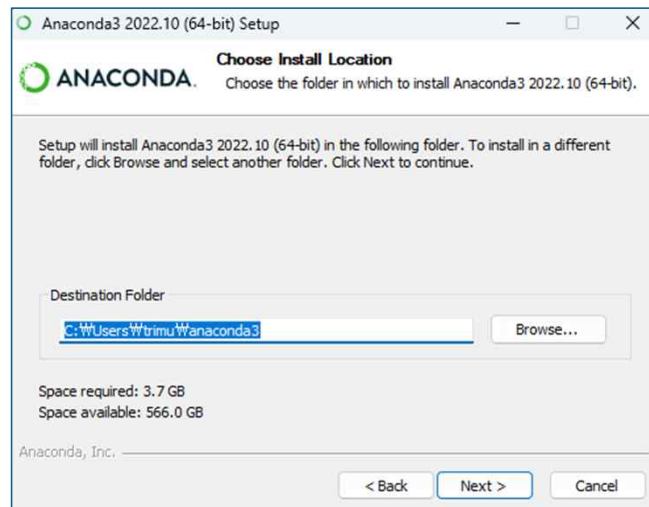
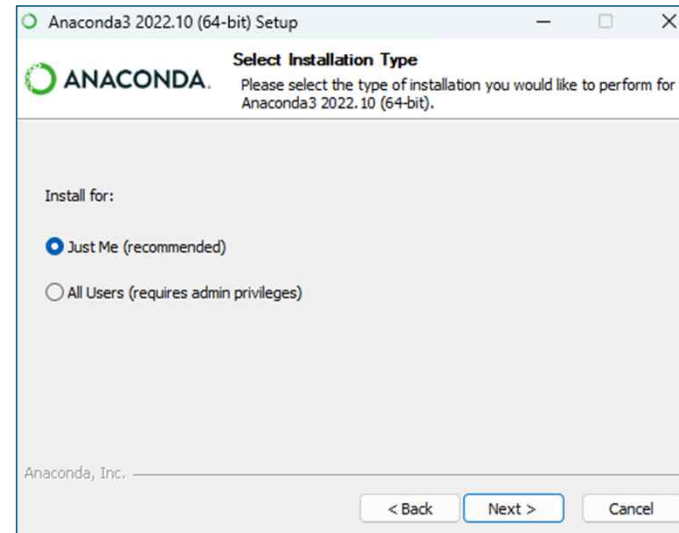
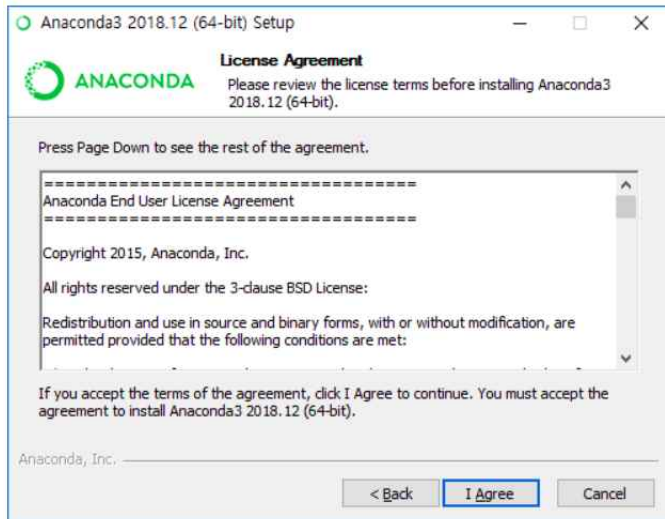
### Anaconda Installers

**Windows**  
  
**Python 3.12**  
📄 64-Bit Graphical Installer (912.3M)

**Mac**  
  
**Python 3.12**  
📄 64-Bit (Apple silicon) Graphical Installer (704.7M)

**Linux**  
  
**Python 3.12**  
📄 64-Bit (x86) Installer (1007.9M)  
📄 64-Bit (AWS Graviton2 / ARM64)

Next 를 눌러 설치 시작



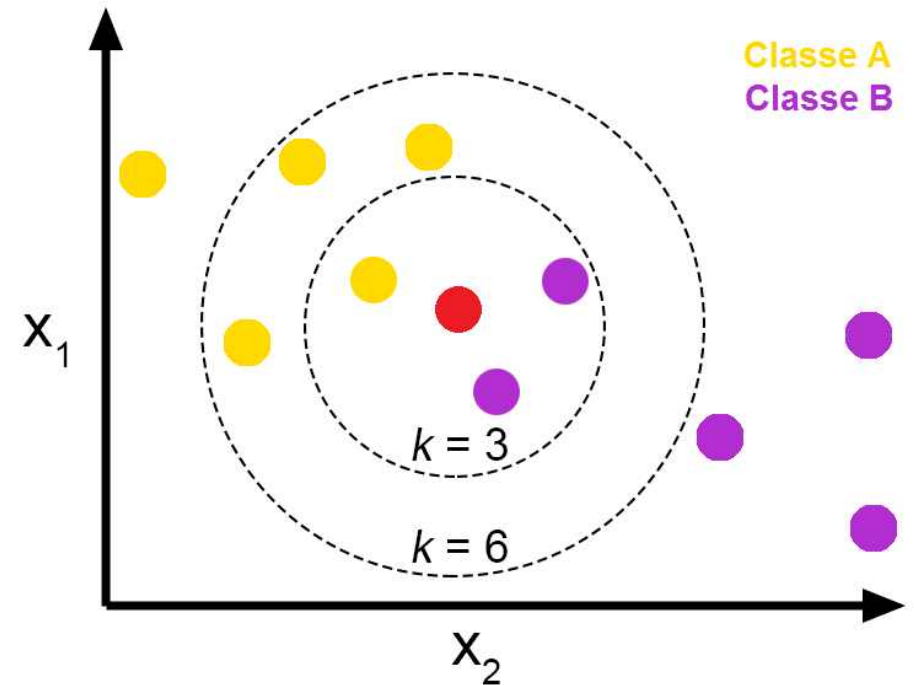
완료

## 실습: 캘리포니아 주택가격 data를 이용한 선형회귀

- Dataset : `sklearn.datasets.fetch_california_housing()`
- Feature : 주택가격 관련 8개 특성 → `MinMaxScaler`로 정규화
- Target : 주택가격
- Model : `linear_model.LinearRegression()`
- 평가기준 1 : MSE (Mean Squared Error),  $R^2$

# KNN (K-Nearest Neighbors, K 최근접 이웃)

- 다른 observation (관측치, X data) 과의 유사성에 따라 분류 (classify)
- 서로 가까이 있는 data 들을 "이웃" (neighbor) 이라고 부른다.
- 가까이 있는 이웃의 label 들 중 가장 많은 것을 unknown case 의 prediction 으로 응답한다.
- 장점 : simple and easy to implement
- 단점 : dataset 이 커지면 slow.  
outlier/missing value 의 영향이 크다.



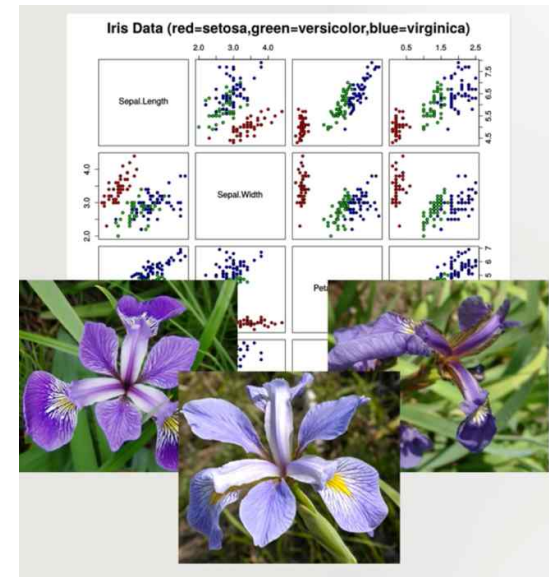
# 실습 : KNN(K-Nearest Neighbors)을 이용한 분류

1. sklearn 에서 제공하는 iris (붓꽃) 분류 dataset 사용 :

꽃잎의 각 부분의 너비와 길이 등을 측정한 데이터이며 150개의 레코드로 구성

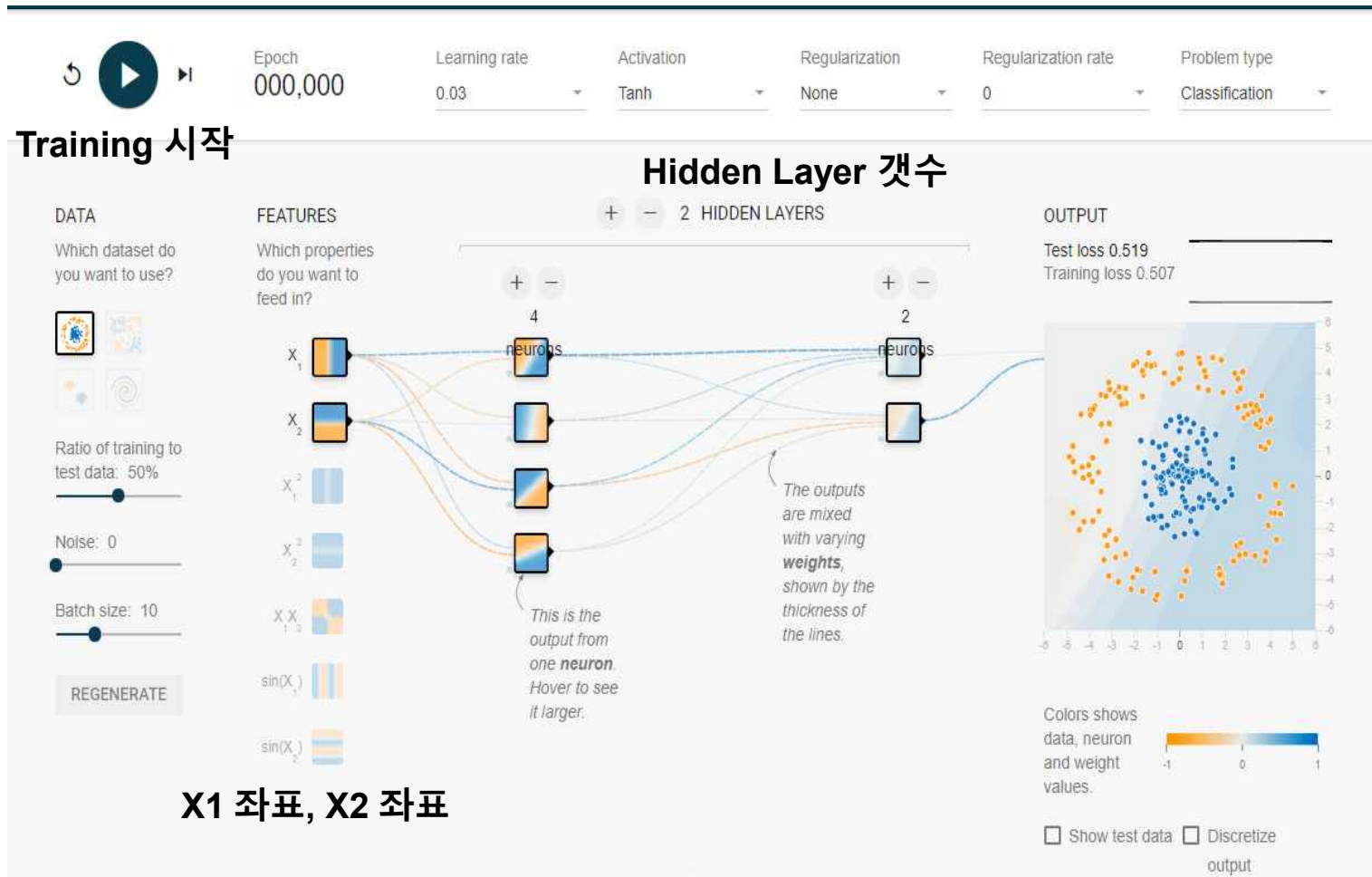
2. Data 의 내용 :

Sepal Length : 꽃받침 길이  
Sepal Width : 꽃받침 너비  
Petal Length : 꽃잎 길이  
Petal Width : 꽃잎 너비



# Neural Network

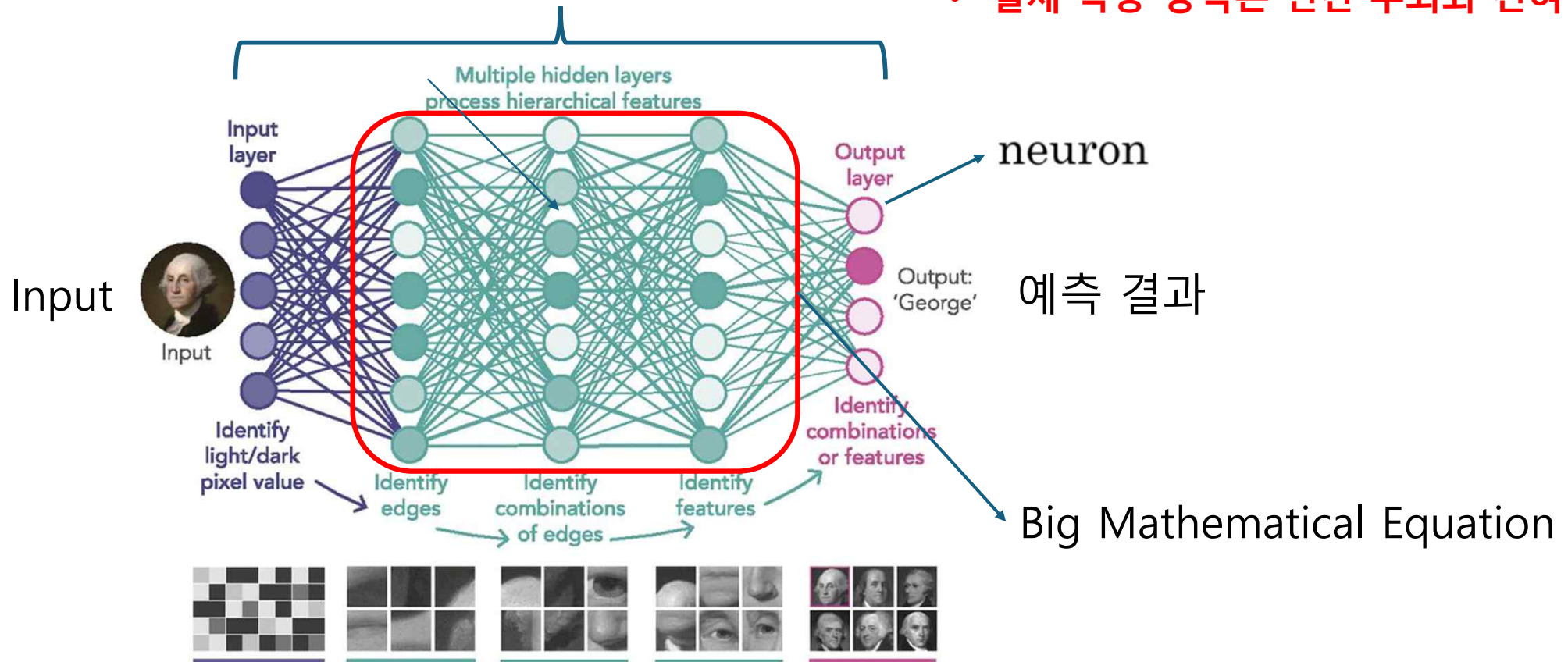
# Tensorflow Playground



# Deep Learning

## Artificial Neural Network

- 인간의 두뇌의 신경망을 모방
- 실제 작동 방식은 인간 두뇌와 전혀 다름





# Artificial Neuron (Perceptron)

구성요소:

**Pre-Activation :**

$$a(x) = b + \sum_i w_i x_i = b + w^T X$$

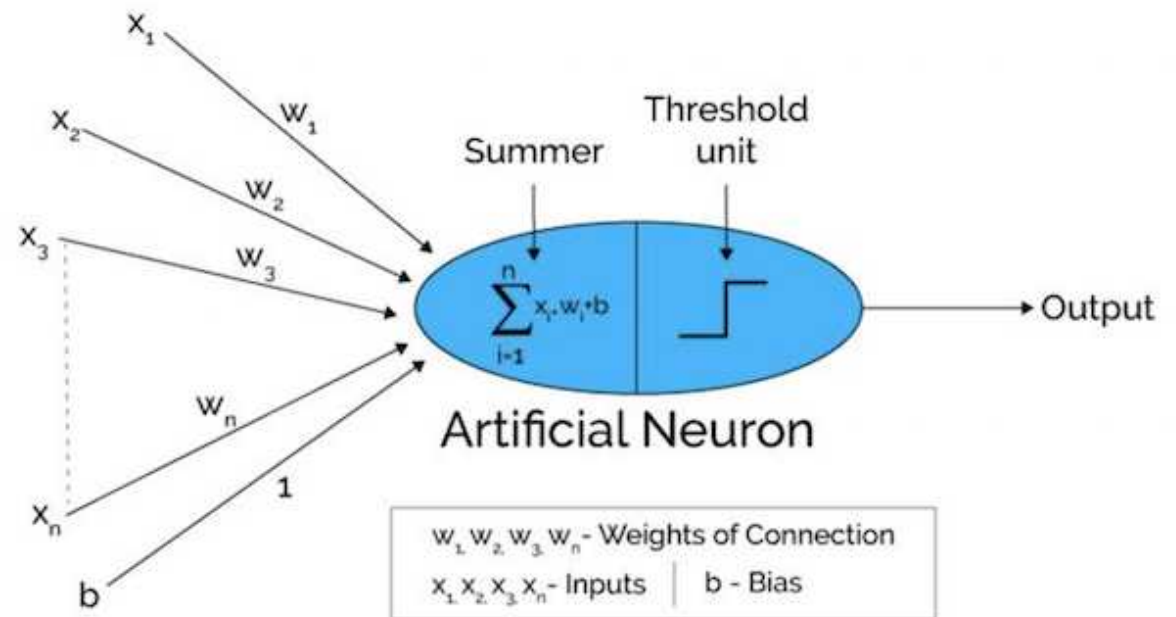
**Activation :**

$$h(x) = g(a(x)) = g( b + \sum_i w_i x_i )$$

$w$  : connection weights

$b$  : bias

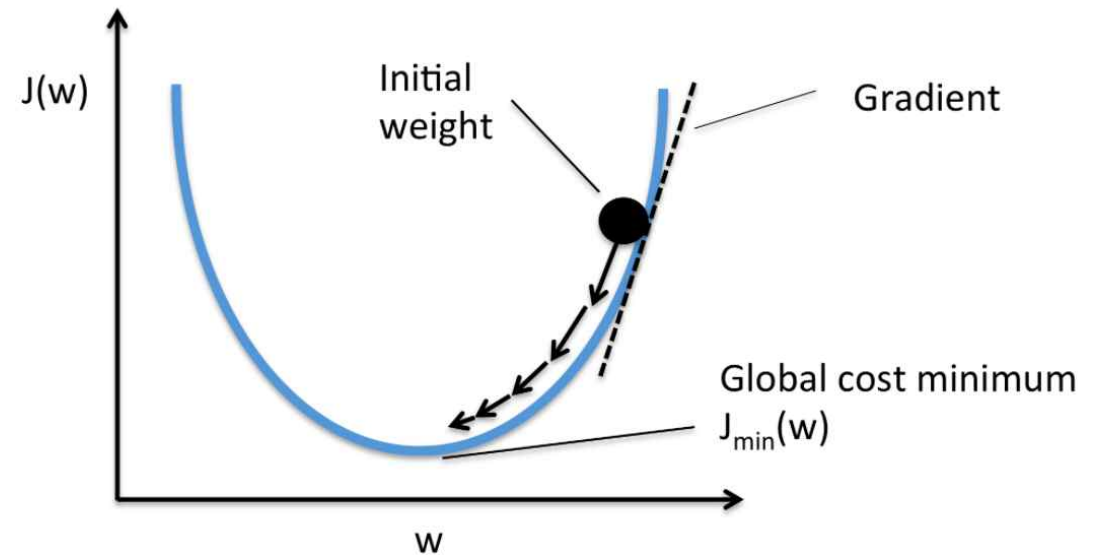
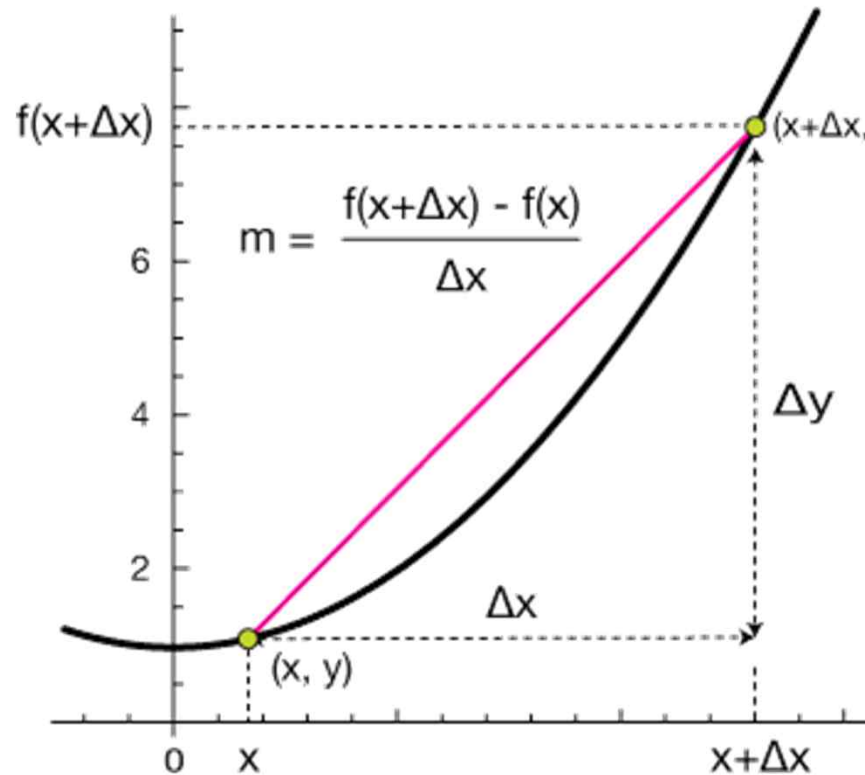
$g$  : activation function



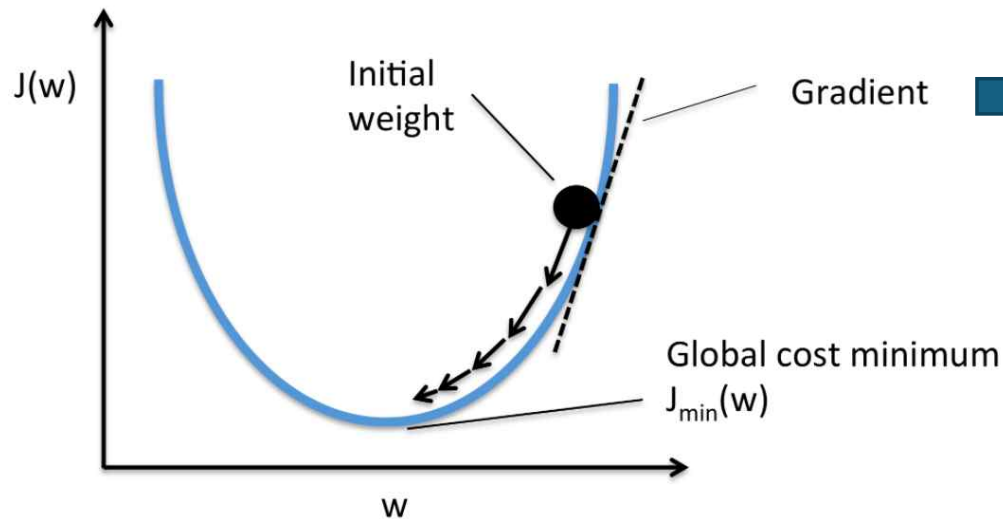
# Deep Neural Network 훈련의 핵심

- Gradient Descent (경사하강법)
  - 목적 – 실제값과 예측값의 차이를 최소화 하는  $\text{parameter}(\theta)$  발견
  - 방법 – 손실함수를 정의하여 손실함수의 값이 0 으로 수렴하도록  $\text{parameter}(\theta)$  조절
- Backpropagation (오차역전파)
  - 손실함수를 최소화 하는 방향으로 신경망 전체의  $\text{parameter}$  가 update 되도록 하는 기법
- 손실함수
  - 비용함수(cost function), 목적함수(object function) 등으로도 불림
  - 경사하강법이 가능하도록 미분 가능한 함수를 정의

# Derivative (도함수, 미분, 접선의 기울기)



# Gradient Descent (경사하강법) Optimization



**방향 – Gradient**  
**(derivative of Cost Function)**

**이동 속도 – Learning Rate**

$$\text{New } W = \text{old } W - (\text{Learning Rate}) * (\text{Gradient})$$

# Computer Vision

# Multiple Levels of Abstraction

Pixel 정보 인식



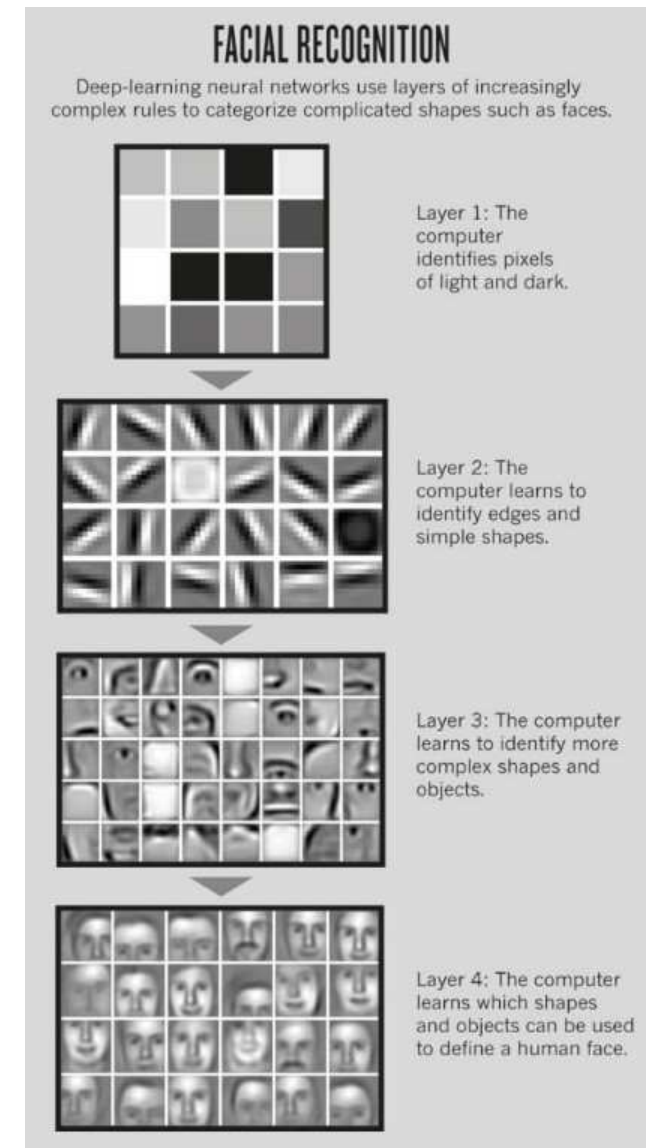
Edge/Simple Shape 특성 학습



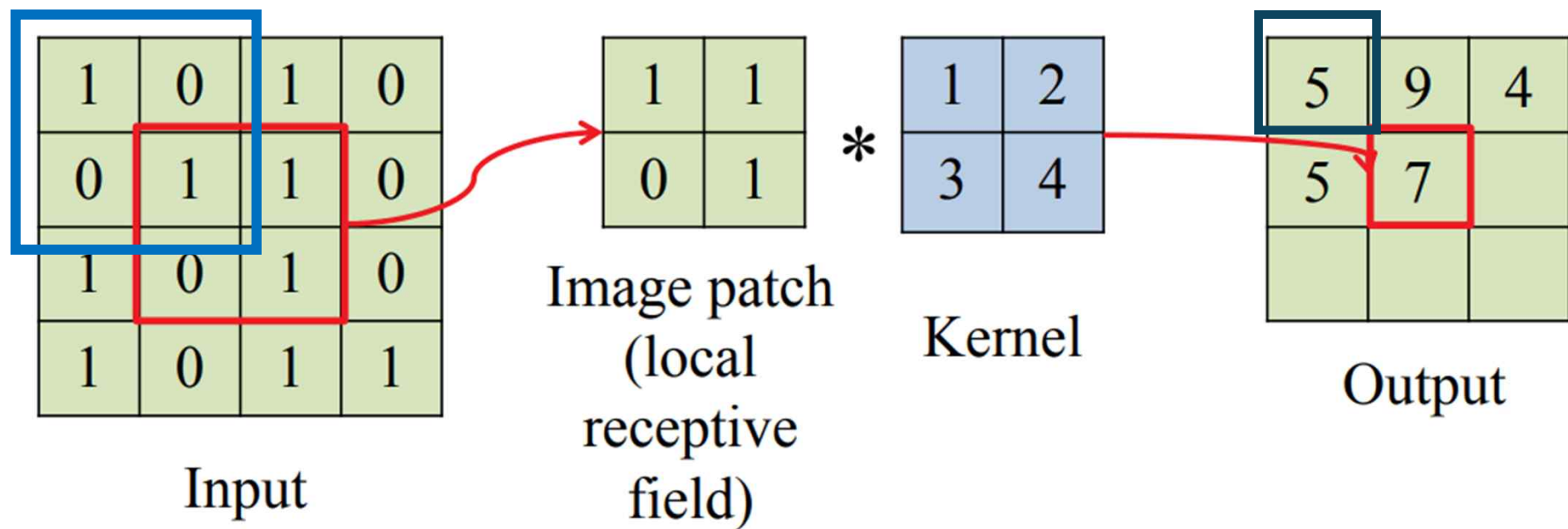
Complex Shape 특성 학습



얼굴인식에 필요한 특성 학습



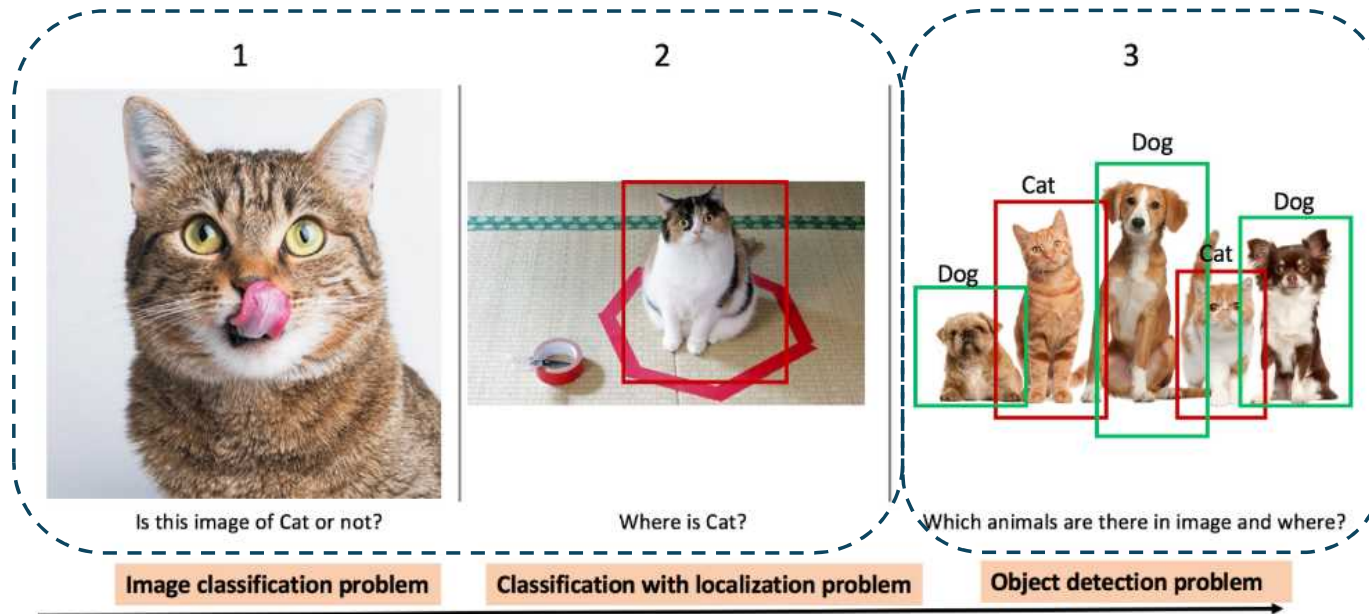
# How Convolution works ?



# Object Recognition (물체 인식) category

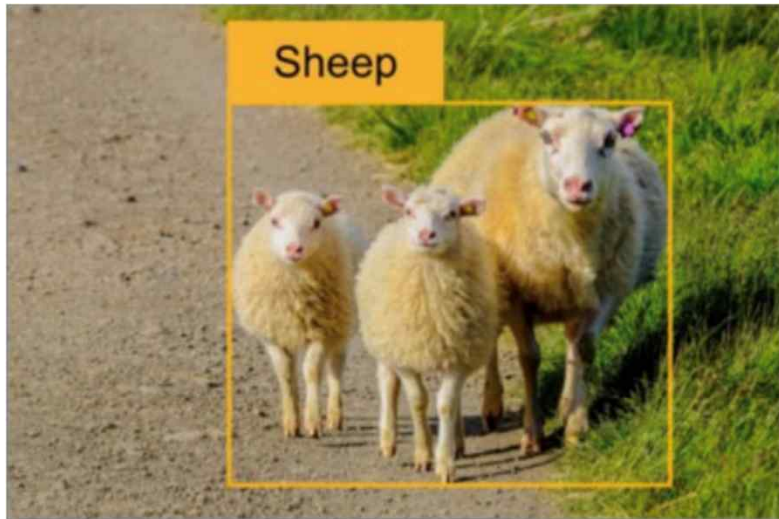
- 1) Image Classification
- 2) Object Localization (물체 위치 감지) – bounding box 그리기
- 3) Object Detection (물체 검출) – (1) + (2)

single object

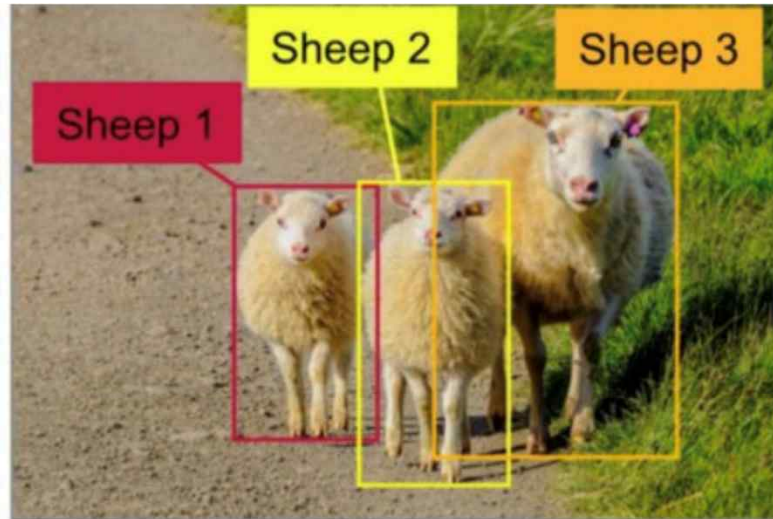


multiple objects



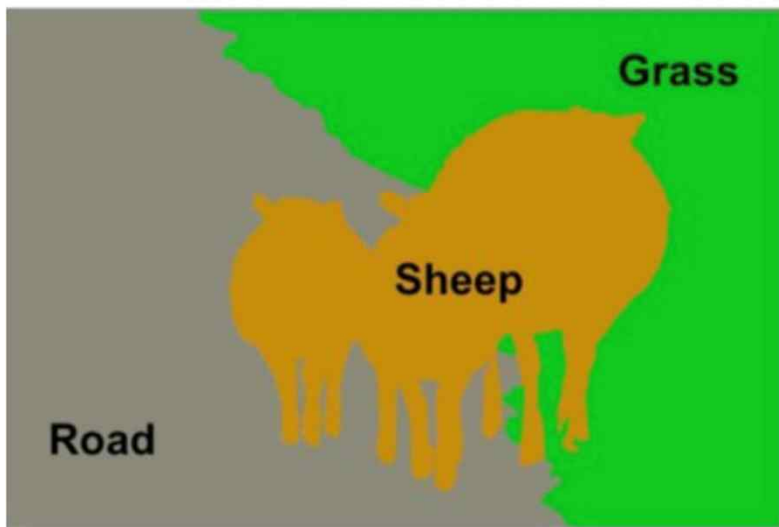


**Classification + Localization**

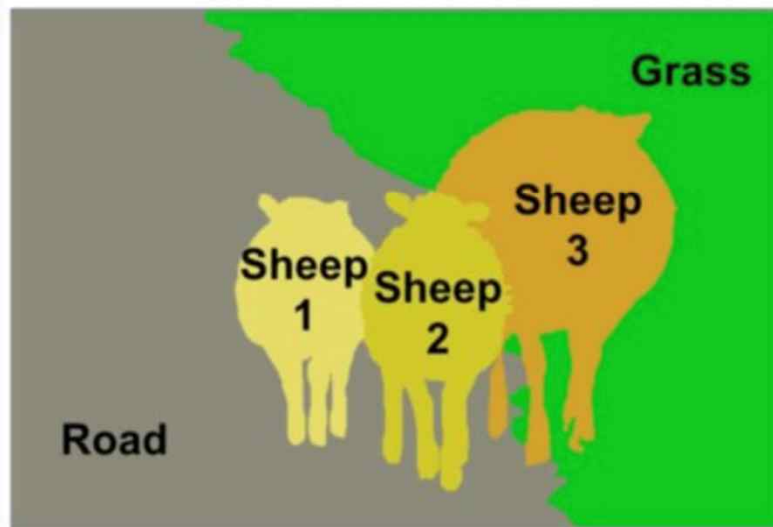


**Object Detection**

← **Localization**  
Location of the Object



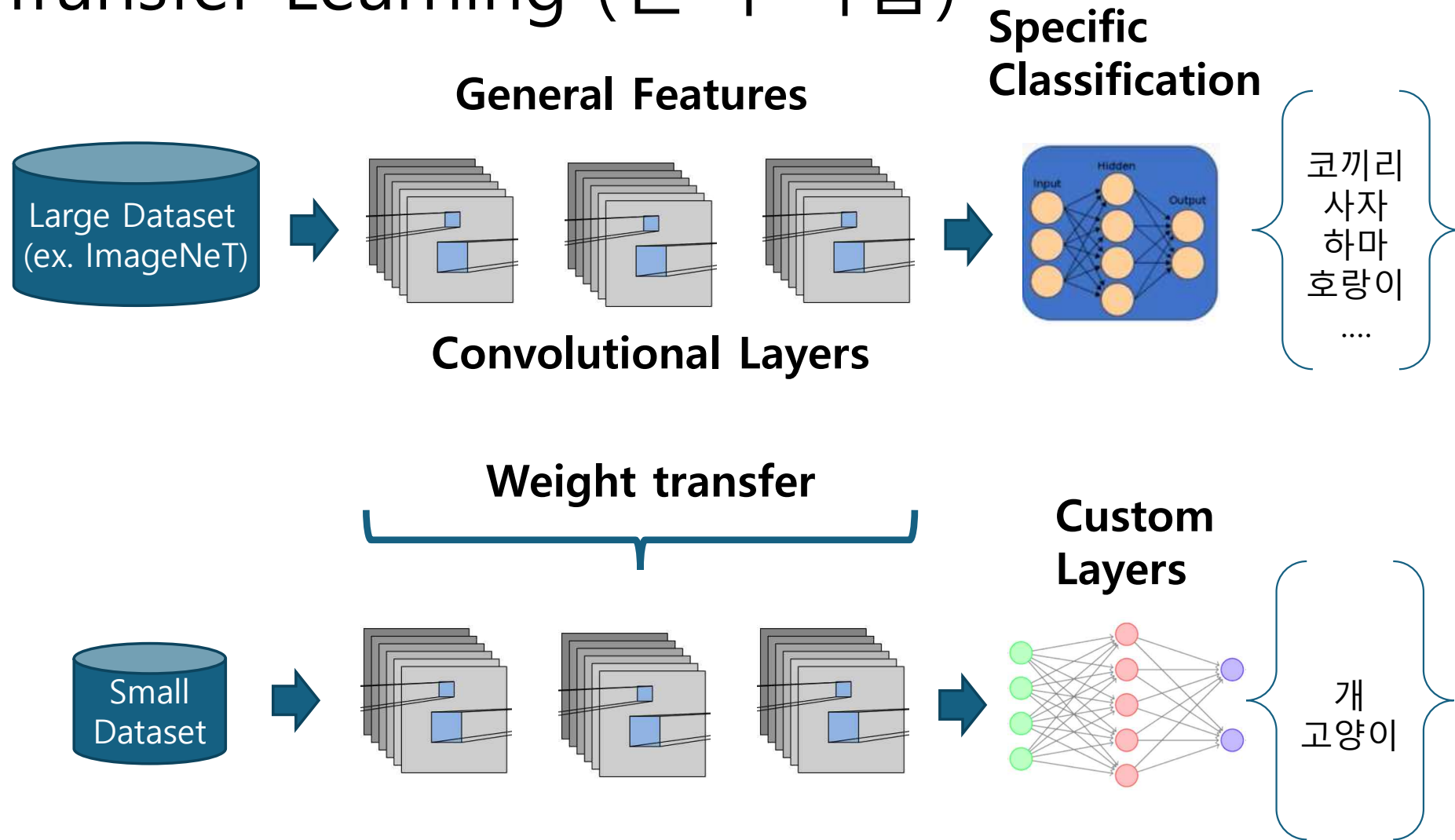
**Semantic Segmentation**



**Instance Segmentation**

← **Segmentation**  
Each pixel belongs to each labels

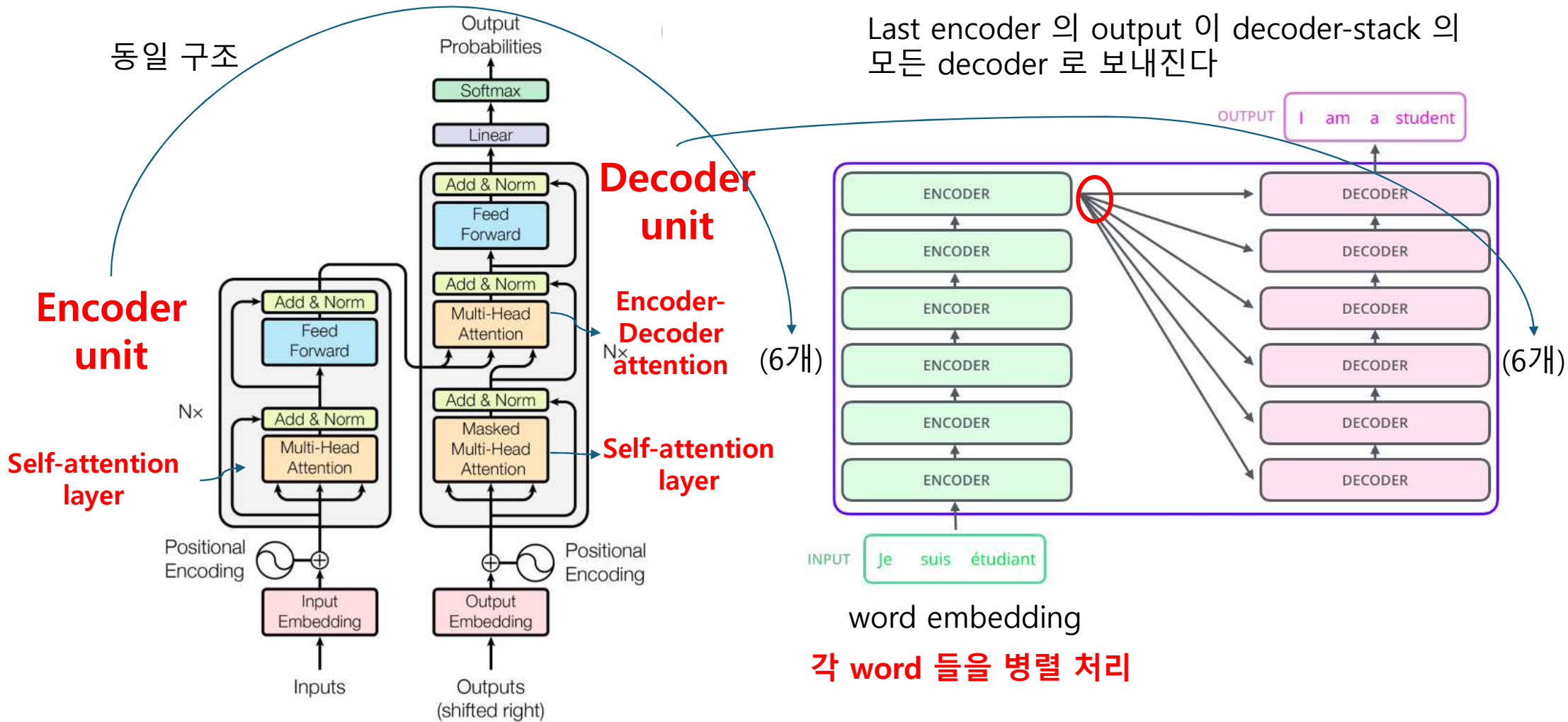
# Transfer Learning (전이 학습)



# 실습 : 전이 학습 - COVID-19 X-ray Detection

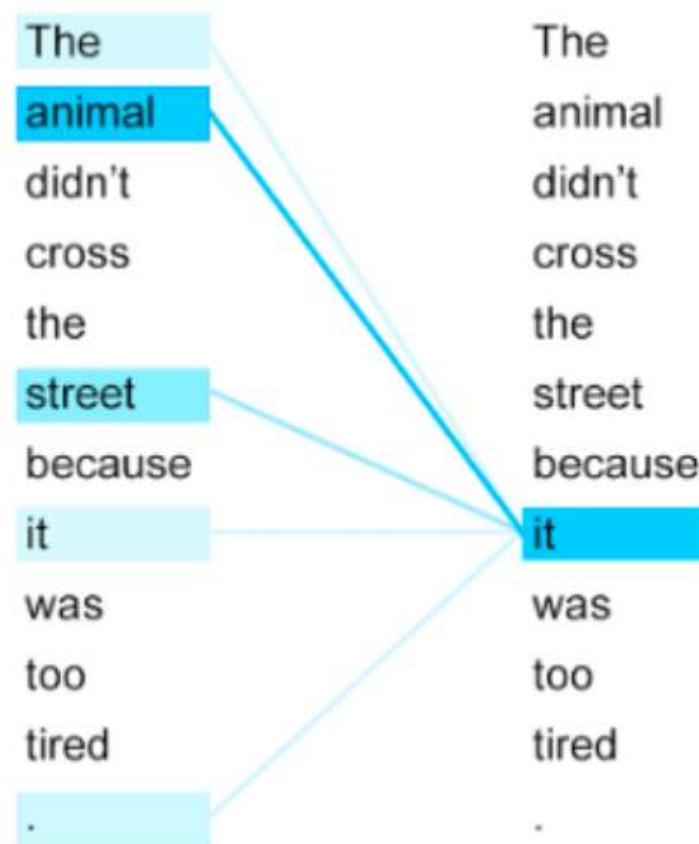
- 흉부 X-ray 이미지를 기반으로 Normal, Viral Pneumonia, COVID-19 분류
- Kaggle의 COVID-19 Radiography Dataset 이용
  - COVID-19 양성: 3616장
  - 정상: 10,192장
  - 바이러스성 폐렴: 1345장
- 데이터 전처리
  - 이미지 크기 조정: 224x224 픽셀
  - 랜덤 수평 뒤집기 및 정규화 수행
- Pretrained ResNet18 을 전이 학습으로 미세 조정

# Transformer Architecture

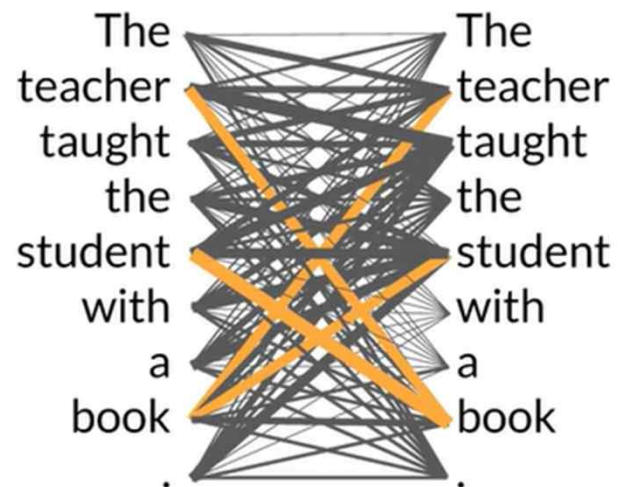
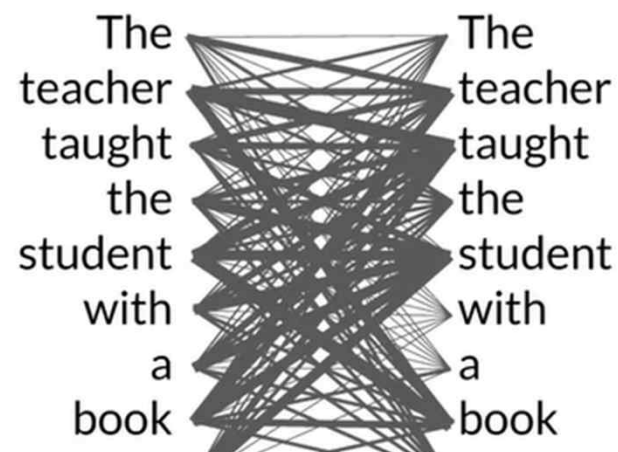
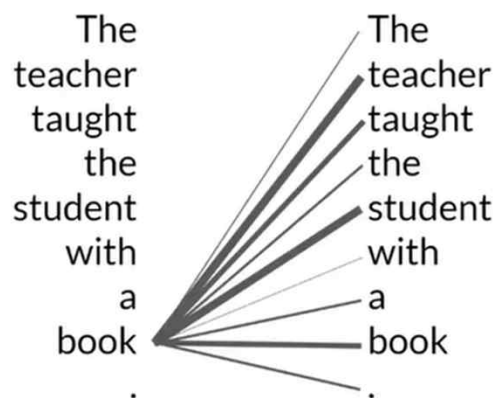
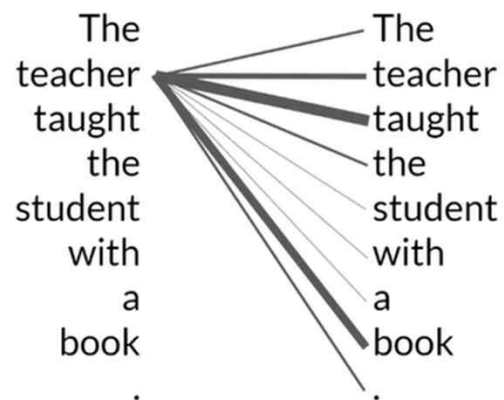


# Self-Attention (Intra-Attention)

- Attention 을 자기 자신에 대해 수행  
→ 문장 내 단어들 간의 유사도를 구함
- Self-attention 계산  
3 개의 vector 필요 (훈련 과정에서 스스로 학습)
  - Query Vector
  - Key Vector
  - Value Vector



# Self-Attention (Intra-Attention)



# What is Word Embedding ?

- 숫자화된 단어의 나열로 부터 sentiment 추출
- 연관성 있는 단어들을 군집화하여 multi-dimension 공간에 vector 로 표시  
→ 단어나 문장을 **vector space** 로 끼워 넣음 (**embedding**)
- 예를 들어, 호감(positive), 비호감(negative) 두가지 label 에 따라 관련 단어들을 두개의 category 로 군집화
  - ex) boring, bad, unfunny → negative
  - funny, good, interesting → positive



# Word Embedding (Feature 화 표시)

dimension	Man (5391)	Woman (9853)	King (4914)	Queen (7157)	Apple (456)	Orange (6257)
성별	-1	1	-0.95	0.97	0.00	0.01
귀족	0.01	0.02	0.93	0.95	-0.01	0.00
나이	0.03	0.02	0.7	0.69	0.03	-0.02
음식	0.04	0.01	0.02	0.01	0.95	0.97



Man (5391) 의 4 dimension vector 표시



King (4914) 의 4 dimension vector 표시



# Embedding matrix (example)

학습된 300 차원 features

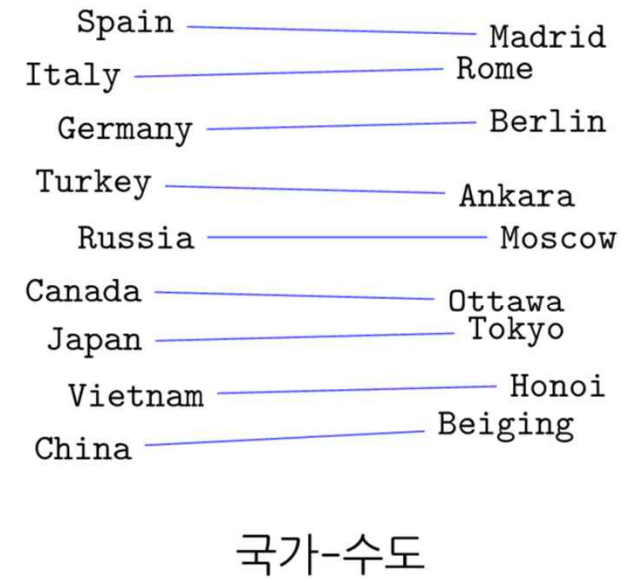
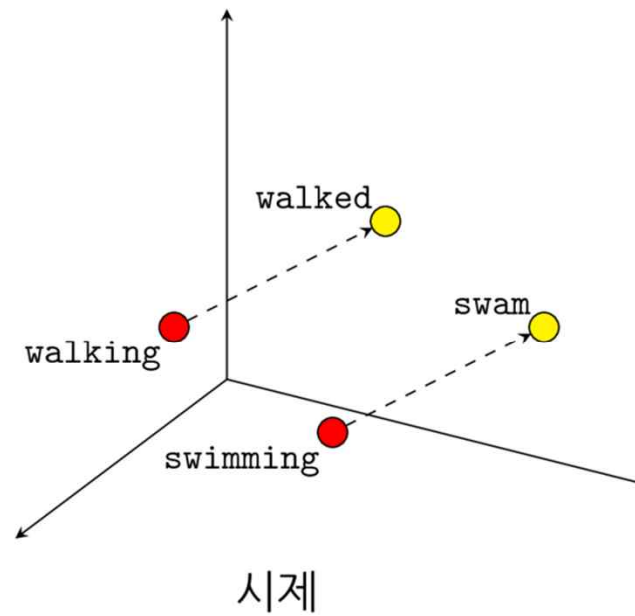
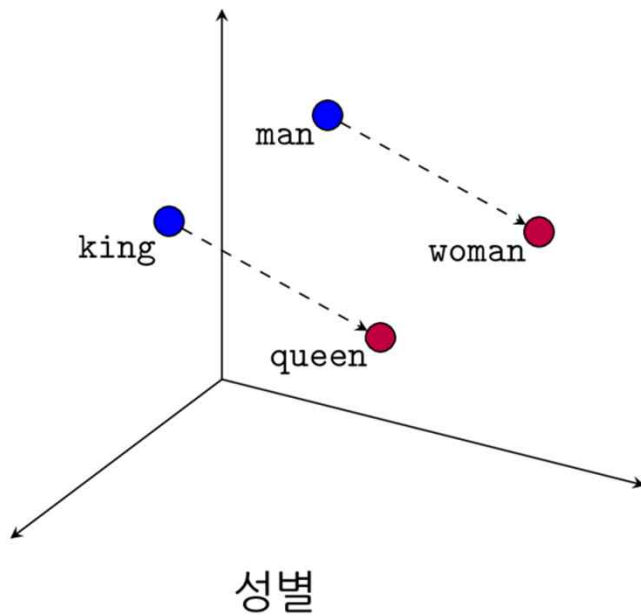
words

	0	1	2	3	4	5	6	7	8	9	...	290	291	292	
fox	-0.348680	-0.077720	0.177750	-0.094953	-0.452890	0.237790	0.209440	0.037886	0.035064	0.899010	...	-0.283050	0.270240	-0.654800	0.105
ham	-0.773320	-0.282540	0.580760	0.841480	0.258540	0.585210	-0.021890	-0.463680	0.139070	0.658720	...	0.464470	0.481400	-0.829200	0.354
brown	-0.374120	-0.076264	0.109260	0.186620	0.029943	0.182700	-0.631980	0.133060	-0.128980	0.603430	...	-0.015404	0.392890	-0.034826	-0.720
beautiful	0.171200	0.534390	-0.348540	-0.097234	0.101800	-0.170860	0.295650	-0.041816	-0.516550	2.117200	...	-0.285540	0.104670	0.126310	0.120
jumps	-0.334840	0.215990	-0.350440	-0.260020	0.411070	0.154010	-0.386110	0.206380	0.386700	1.460500	...	-0.107030	-0.279480	-0.186200	-0.545
eggs	-0.417810	-0.035192	-0.126150	-0.215930	-0.669740	0.513250	-0.797090	-0.068611	0.634660	1.256300	...	-0.232860	-0.139740	-0.681080	-0.370
beans	-0.423290	-0.264500	0.200870	0.082187	0.066944	1.027600	-0.989140	-0.259950	0.145960	0.766450	...	0.048760	0.351680	-0.786260	-0.368
sky	0.312550	-0.303080	0.019587	-0.354940	0.100180	-0.141530	-0.514270	0.886110	-0.530540	1.556600	...	-0.667050	0.279110	0.500970	-0.275
bacon	-0.430730	-0.016025	0.484620	0.101390	-0.299200	0.761820	-0.353130	-0.325290	0.156730	0.873210	...	0.304240	0.413440	-0.540730	-0.038
breakfast	0.073378	0.227670	0.208420	-0.456790	-0.078219	0.601960	-0.024494	-0.467980	0.054627	2.283700	...	0.647710	0.373820	0.019931	-0.035
toast	0.130740	-0.193730	0.253270	0.090102	-0.272580	-0.030571	0.096945	-0.115060	0.484000	0.848380	...	0.142080	0.481910	0.045167	0.055
today	-0.156570	0.594890	-0.031445	-0.077586	0.278630	-0.509210	-0.066350	-0.081890	-0.047986	2.803600	...	-0.326580	-0.413380	0.367910	-0.265
blue	0.129450	0.036518	0.032298	-0.060034	0.399840	-0.103020	-0.507880	0.076630	-0.422920	0.815730	...	-0.501280	0.169010	0.548250	-0.318
green	-0.072368	0.233200	0.137260	-0.156630	0.248440	0.349870	-0.241700	-0.091426	-0.530150	1.341300	...	-0.405170	0.243570	0.437300	-0.461
kings	0.259230	-0.854690	0.360010	-0.642000	0.568530	-0.321420	0.173250	0.133030	-0.089720	1.528600	...	-0.470090	0.063743	-0.545210	-0.195
dog	-0.057120	0.052685	0.003026	-0.048517	0.007043	0.041856	-0.024704	-0.039783	0.009614	0.308416	...	0.003257	-0.036864	-0.043878	0.000
sausages	-0.174290	-0.064869	-0.046976	0.287420	-0.128150	0.647630	0.056315	-0.240440	-0.025094	0.502220	...	0.302240	0.195470	-0.653980	-0.295
lazy	-0.353320	-0.299710	-0.176230	-0.321940	-0.385640	0.586110	0.411160	-0.418680	0.073093	1.486500	...	0.402310	-0.038554	-0.288670	-0.245
love	0.139490	0.534530	-0.252470	-0.125650	0.048748	0.152440	0.199060	-0.065970	0.128830	2.055900	...	-0.124380	0.178440	-0.099469	0.008
quick	-0.445630	0.191510	-0.249210	0.465900	0.161950	0.212780	-0.046480	0.021170	0.417660	1.686900	...	-0.329460	0.421860	-0.039543	0.150

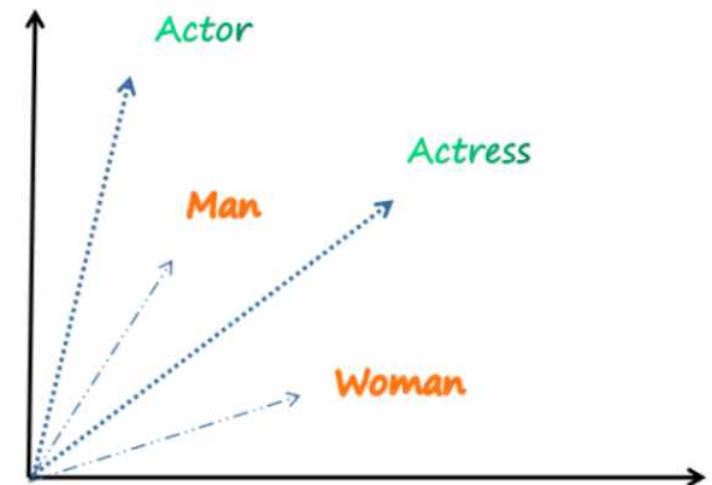
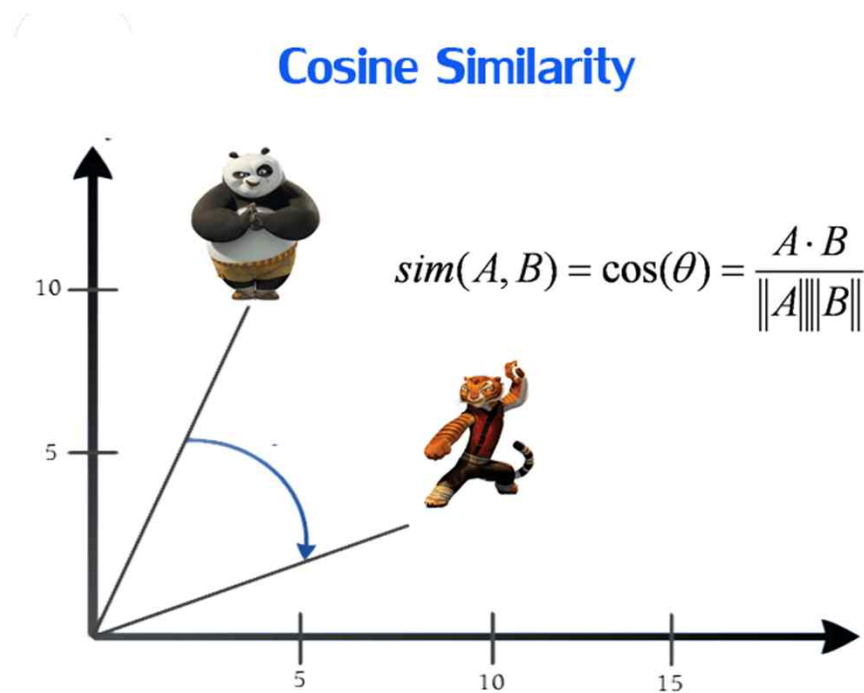
20 rows x 300 columns

# Word Embedding 의 결과

king - queen  $\approx$  man - woman



# 유사도 측정 (Cosine Similarity)



➡ A, B 유사하면  $\approx 1$   
관련 없으면  $\approx 0$

# 실습 : Huggingface Embedding 모델

- Embedding 모델을 이용하여 embedding 변환

```
sentences = [  
    '나는 인공지능 공부를 좋아한다.',  
    '인공지능은 매우 흥미롭다.',  
    '오늘 날씨가 흐리고 비가 온다.'  
]
```

→

	0	1	2	3	4	5	\
나는 인공지능 공부를 좋아한다.	-0.021738	-0.012001	-0.013173	0.011548	0.049148	-0.015339	
인공지능은 매우 흥미롭다.	-0.015396	0.010621	-0.042270	0.030104	0.010643	-0.035229	
오늘 날씨가 흐리고 비가 온다.	0.019867	-0.001774	-0.074277	-0.024289	0.029764	0.045755	

- 문장간의 cosine 유사도 계산

문장 간 코사인 유사도:

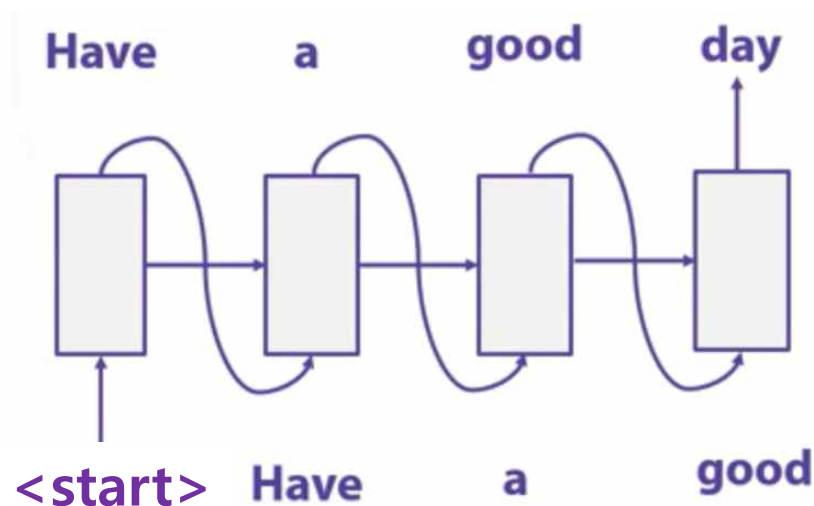
나는 인공지능 공부를 좋아한다. vs 인공지능은 매우 흥미롭다.: 0.5847

나는 인공지능 공부를 좋아한다. vs 오늘 날씨가 흐리고 비가 온다.: 0.1017

인공지능은 매우 흥미롭다. vs 오늘 날씨가 흐리고 비가 온다.: 0.1661

# How to generate language ?

- 이전 time step 의 output 을 next time step 의 input 으로 feed 하고, 각 step 에서 가장 높은 확률의 다음 단어를 선택 (greedy selection)하거나 혹은 확률 분포에 따라 sampling





# 실습 : autoregressive\_language\_generation

- Autoregressive 문장 생성
- Huggingface의 transformers library 사용
- Pre-trained 모델 이용

```
1 # 문장 시작 부분
2 input_text = "Once upon a time"
3 input_ids = tokenizer.encode(input_text, return_tensors="pt")
4 input_ids
```

```
tensor([[7454, 2402, 257, 640]])
```

```
tensor([[7454, 2402, 257, 640, 11]])
tensor([[7454, 2402, 257, 640, 11, 612]])
tensor([[7454, 2402, 257, 640, 11, 612, 373]])
tensor([[7454, 2402, 257, 640, 11, 612, 373, 257]])
tensor([[7454, 2402, 257, 640, 11, 612, 373, 257, 582]])
tensor([[7454, 2402, 257, 640, 11, 612, 373, 257, 582, 508]])
tensor([[7454, 2402, 257, 640, 11, 612, 373, 257, 582, 508, 5615]])
tensor([[7454, 2402, 257, 640, 11, 612, 373, 257, 582, 508, 5615, 287]])
tensor([[7454, 2402, 257, 640, 11, 612, 373, 257, 582, 508, 5615, 287,
257]])
tensor([[7454, 2402, 257, 640, 11, 612, 373, 257, 582, 508, 5615, 287,
257, 7404]])
tensor([[7454, 2402, 257, 640, 11, 612, 373, 257, 582, 508, 5615, 287,
257, 7404, 1444]])
tensor([[7454, 2402, 257, 640, 11, 612, 373, 257, 582, 508, 5615, 287,
257, 7404, 1444, 509]])
tensor([[7454, 2402, 257, 640, 11, 612, 373, 257, 582, 508,
5615, 287, 257, 7404, 1444, 509, 17716]])
tensor([[7454, 2402, 257, 640, 11, 612, 373, 257, 582, 508,
5615, 287, 257, 7404, 1444, 509, 17716, 322]])
tensor([[7454, 2402, 257, 640, 11, 612, 373, 257, 582, 508,
5615, 287, 257, 7404, 1444, 509, 17716, 322, 13]])
tensor([[7454, 2402, 257, 640, 11, 612, 373, 257, 582, 508,
5615, 287, 257, 7404, 1444, 509, 17716, 322, 13, 679]])
Once upon a time, there was a man who lived in a village called Krakow. He
```

ChatGPT

# What is ChatGPT ?

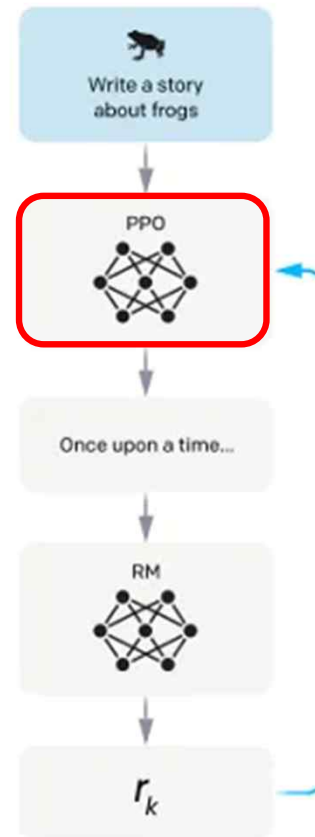
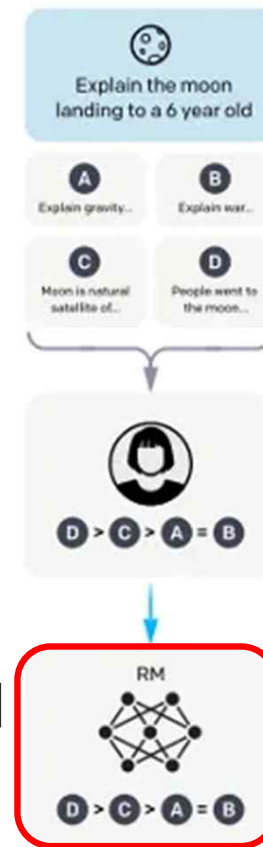
- ChatGPT라고 불리는 GPT-3.5는 GPT-3의 작지만 대화에 특화된 버전.  
이 모델은 채팅의 맥락, 구조 및 어조를 더 깊이 이해하면서 자연스럽게 대화를 이어가는 유일한 목적으로 훈련. 따라서 채팅과 관련하여 범용 GPT-3에 비해 훨씬 더 정확하고 빠른 모델을 만든다.
- GPT-4는 ChatGPT의 진화된 Multi-Modal 버전.  
텍스트나 이미지 입력을 받아 텍스트를 출력하는 대규모 다중 모드 모델로, 이전 모델들보다 더 넓은 일반 지식과 진보된 추론 능력을 통해 더 정확하게 어려운 문제를 해결할 수 있다.



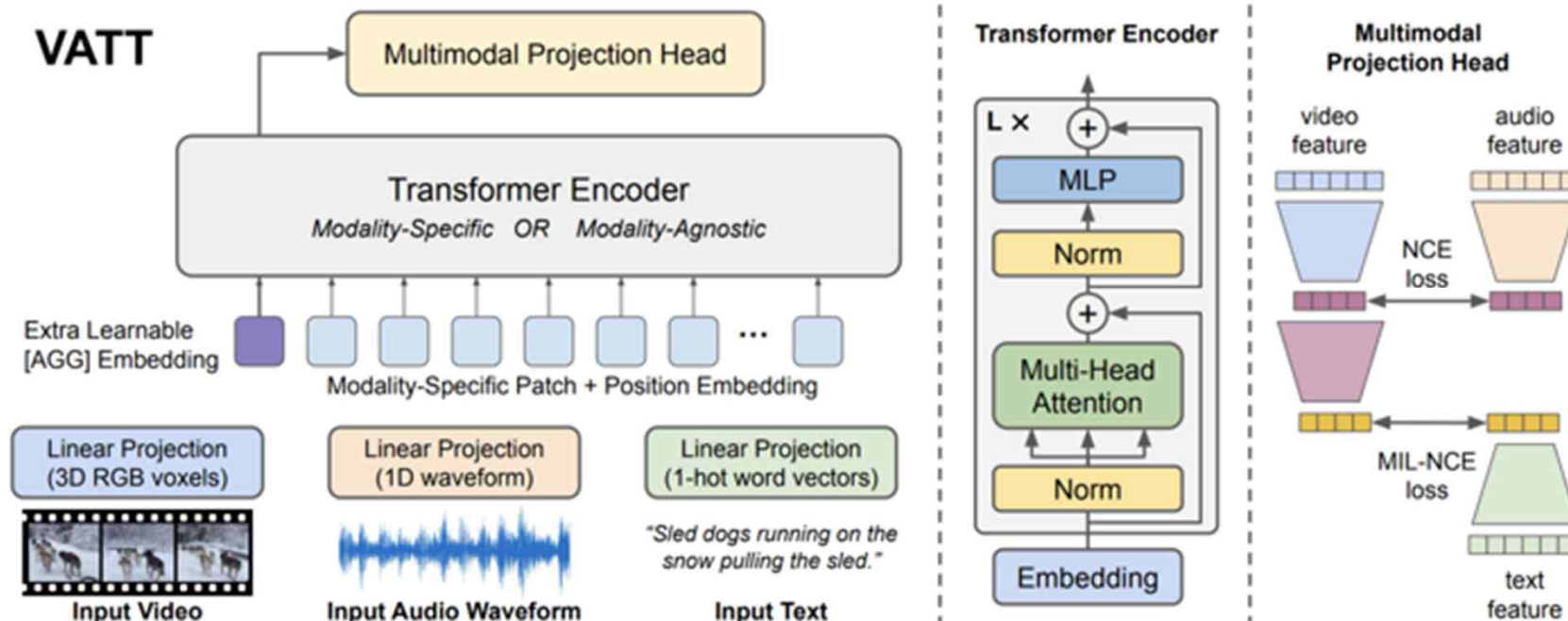
# ChatGPT Fine Tuning 과정

## RLHF(Reinforcement Learning from Human Feedback)

- 1단계 : **Supervised Fine Tuning(SFT) model** 작성
  - GPT-3를 초기 model로 사용
  - 인간이 의도하는 정책을 학습 시키기 위해 인간 labeler가 모델이 생성한 output 에 점수 부여
  - 이렇게 얻어진 데이터를 이용하여 지도학습 방법으로 Fine Tuning
- 2단계 : **Reward Model** 작성(인간 선호도 모방)
  - 1단계에서 Fine Tuning된 SFT 모델의 여러 답변 후보에 인간 labeler가 ranking을 매긴 데이터셋 수집
  - 이 데이터셋을 이용하여 새로운 reward model 학습
- 3단계 : **STF 모델을 강화학습**



# VATT(Vision, Audio, Text Transformer) 모델



Transformer Encoder를 통해 얻어진 특징 벡터들은 공통 공간으로 투영하여 대조학습을 시킵니다.

Input Video: 3D RGB Voxel로 변환된 비디오 프레임. 각 비디오 프레임은 3차원 형태로 입력됩니다.  
 Input Audio Waveform: 1D 파형으로 변환된 오디오 데이터. 오디오 입력은 파형 형태로 처리됩니다.  
 Input Text: 1-hot word vectors로 변환된 텍스트 데이터. 텍스트 입력은 단어 벡터 형태로 처리됩니다.  
 → 각 모달리티의 입력 데이터는 선형 투영(Linear Projection)을 통해 임베딩 벡터로 변환

# LLM (Large Language Model)

활용 방법

# LLM 개요

- LLM(Large Language Model)은 대규모 데이터셋을 학습하여 자연어를 이해하고 생성할 수 있는 딥러닝 모델
- 일반적으로 Transformer 기반의 신경망 구조를 사용하며, 사전 훈련(pre-training)과 미세 조정(fine-tuning)을 통해 다양한 자연어 처리(NLP) 작업을 수행
- **LLM의 특징**
  - 방대한 텍스트 데이터를 학습하여 다양한 언어적 패턴을 익힘
  - 질문 응답, 번역, 문서 요약, 코드 생성 등 다양한 작업 자연어 작업 수행 가능
  - Few-shot & Zero-shot Learning - 제한된 예제만으로도 새로운 작업을 수행
  - 멀티모달 - 텍스트뿐만 아니라 이미지, 음성, 동영상과 같은 다양한 입력을 처리

# 최신 LLM 모델 비교

모델명	발표	주요 특징	특징
GPT-5	2025 OpenAI	- 다중 모달 지원 (텍스트, 이미지, 음성) - 깊은 추론	AI 챗봇(ChatGPT)
Claude Opus4 /Sonnet 4	2024 Anthropic	- 안전성을 강화한 AI 모델 - 강력한 추론 및 논리적 사고 능력	AI 윤리 및 법률 도구 교육 및 연구 보조 데이터 분석
Gemini 2.5	2025 Google	- 멀티모달 처리 지원 - 1백만 토큰 이상의 컨텍스트 윈도우 지원	복잡한 대화형 AI 시스템 다국어 번역 및 문서 생성
Llama 4	2024 Meta	- 오픈소스 LLM	대규모 문맥 처리를 가능하게 하는 연구 중심 모델
Grok 4	2025 xAI	- X 플랫폼 통합	팬덤/비평 반응이 공존하는 독특한 포지션
DeepSeek	2025 DeepSeek	- 오픈 소스 LLM (MIT 라이선스) - 혼합전문가(Mixture of Experts) 구조와 강화학습으로 모델 개선	개선저비용 고효율 추구

# LLM API (Application Programming Interface)

- API란 응용 프로그램 간의 상호 작용을 위한 인터페이스
- 복잡한 모델을 직접 구축하지 않고도 LLM의 기능 활용 가능
- 최신 모델의 기능을 손쉽게 통합 가능
- 주요 LLM API 제공 업체 - OpenAI, Anthropic, Google DeepMind 등
- API 활용 시 고려사항
  - 요금 구조
  - 요청 제한
  - 보안 및 개인정보 보호


# LangChain

- LLM을 활용한 애플리케이션 개발을 지원하는 프레임워크
- 자연어 처리, 데이터 연결, 메모리 관리, API 통합 등 다양한 기능 제공
- LangChain의 장점
  - 복잡한 LLM 기반 애플리케이션을 모듈화하여 개발
  - 다양한 외부 데이터 소스 및 API와의 통합 인터페이스 지원
  - 오픈소스로 활발한 커뮤니티와 지속적인 업데이트
- 단점
  - 복잡성 및 과도한 추상화로 개발 효율성을 저하
  - 추가적인 연산 부담으로 성능 저하.
  - 학습 곡선이 높아 초보자나 작은 프로젝트를 진행하는 개발자들에게 부담
  - 잦은 버전 변경으로 업그레이드 부담

# Hugging Face (<https://github.com/huggingface>)


- NLP 와 관련된 다양한 패키지를 제공


Search or jump to... Pull requests Issues Marketplace Explore


 **Hugging Face**  
The AI community building the future.  
NYC + Paris <https://huggingface.co/> Verified


Overview Repositories 52 Packages People 26 Projects Sponsoring 4


Pinned


 **transformers**  
🤖 Transformers: State-of-the-art Natural Language Processing for Pytorch, TensorFlow, and JAX.  
Python ☆ 50.4k 🍴 12k

 **datasets**  
🤖 The largest hub of ready-to-use datasets for ML models with fast, easy-to-use and efficient data manipulation tools  
Python ☆ 8.8k 🍴 1.1k

 **tokenizers**  
🚀 Fast State-of-the-Art Tokenizers optimized for Research and Production  
Rust ☆ 4.8k 🍴 375

 **knoockknoock**  
🔔 Knock Knock: Get notified when your training ends with only two additional lines of code  
Python ☆ 2.2k 🍴 190

 **accelerate**  
🚀 A simple way to train and use PyTorch models with multi-GPU, TPU, mixed-precision  
Python ☆ 1.8k 🍴 89

 **huggingface\_hub**  
all the open source things related to the Hugging Face Hub.  
Python ☆ 196 🍴 34



# Hugging Face

- Transformers
  - Transformer 기반 (masked) language model 알고리즘 제공
  - 기 학습된 (pretrained) 모델 배포
- Tokenizers
  - transformers package 에서 사용할 수 있는 subword 토크나이저 학습
  - transformers 와 분리되어 다른 목적에도 이용할 수 있다.
- dataset → tokenizer → models 의 언어 모델 학습에 필요한 전 과정을 지원

# 실습: Text 생성 및 Prompt 예제

- OpenAI Response API 사용 또는,
- Hugging Face 에서 ["naver-hyperclovax/HyperCLOVAX-SEED-Text-Instruct-1.5B"](https://huggingface.co/naver-hyperclovax/HyperCLOVAX-SEED-Text-Instruct-1.5B) 모델을 download 하여 사용

Cursor AI

# Python 설치

Python PSF Docs PyPI Jobs Community

python™ Donate Search GO Socialize

About Downloads Documentation Community Success Stories News Events

Python >>> Downloads >>> Windows

## Python Releases for Windows

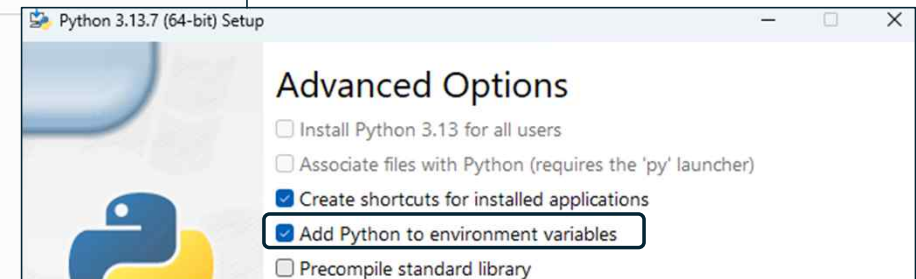
- [Latest Python install manager - Python install manager 25.0 beta 14](#)
- [Latest Python 3 Release - Python 3.13.7](#)

### Stable Releases

- [Python 3.13.7 - Aug. 14, 2025](#)  
**Note that Python 3.13.7 cannot be used on Windows 7 or earlier.**
  - [Download Windows installer \(64-bit\)](#)
  - [Download Windows installer \(32-bit\)](#)
  - [Download Windows installer \(ARM64\)](#)
  - [Download Windows embeddable package \(64-bit\)](#)
  - [Download Windows embeddable package \(32-bit\)](#)
  - [Download Windows embeddable package \(ARM64\)](#)
  - [Download Windows release manifest](#)

### Pre-releases

- [Python install manager 25.0 beta 14 - Aug. 27, 2025](#)
  - [Download Installer \(MSIX\)](#)
  - [Download MSI package](#)
- [Python install manager 25.0 beta 13 - Aug. 14, 2025](#)
  - [Download Installer \(MSIX\)](#)
  - [Download MSI package](#)
- [Python 3.14.0rc2 - Aug. 14, 2025](#)
  - [Download Windows installer \(64-bit\)](#)
  - [Download Windows installer \(32-bit\)](#)



# uv 설치

- <https://docs.astral.sh/uv/getting-started/installation/>

## Installation methods

↑ Back to top

Install uv with our standalone installers or your package manager of choice.

### Standalone installer

uv provides a standalone installer to download and install uv:

macOS and Linux

**Windows**

Use `irm` to download the script and execute it with `iex`: **powershell 에서 실행**

```
PS> powershell -ExecutionPolicy ByPass -c "irm https://astral.sh/uv/install.ps1 | iex"
```

Changing the [execution policy](#) allows running a script from the internet.

```
Windows PowerShell
PS> powershell -ExecutionPolicy ByPass -c "irm https://astral.sh/uv/install.ps1 | iex"
```

```
명령 프롬프트
Microsoft Windows [Version 10.0.26100.4946]
(c) Microsoft Corporation. All rights reserved.

C:\Users\trimu> uv
An extremely fast Python package manager.

Usage: uv [OPTIONS] <COMMAND>

Commands:
  auth  Manage authentication
  run   Run a command or script
  init  Create a new project
  add   Add dependencies to the project
  remove Remove dependencies from the project
  version Read or update the project's version
  sync  Update the project's environment
  lock  Update the project's lockfile
  export Export the project's lockfile to an alternate format
  tree  Display the project's dependency tree
  format Format Python code in the project
  tool  Run and install commands provided by Python packages
```

# uv 프로젝트 만들기

- New folder 생성 → “uv init” or “uv init folder-name”

```
포트 문제 터미널 디버그 콘솔 SPELL CHECKER + v C:\ cmd ...
Microsoft Windows [Version 10.0.26100.4946]
(c) Microsoft Corporation. All rights reserved.

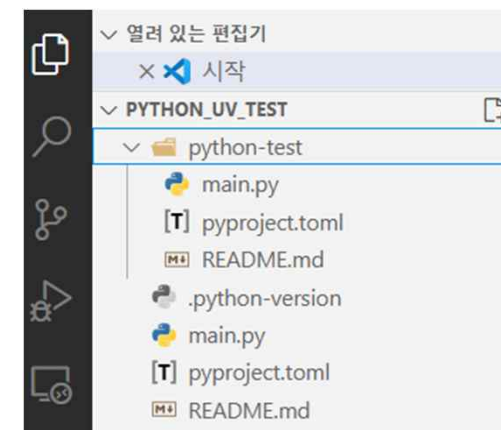
C:\Users\trimu\OneDrive\python_uv_test>uv init
Initialized project `python-uv-test`

C:\Users\trimu\OneDrive\python_uv_test>
```

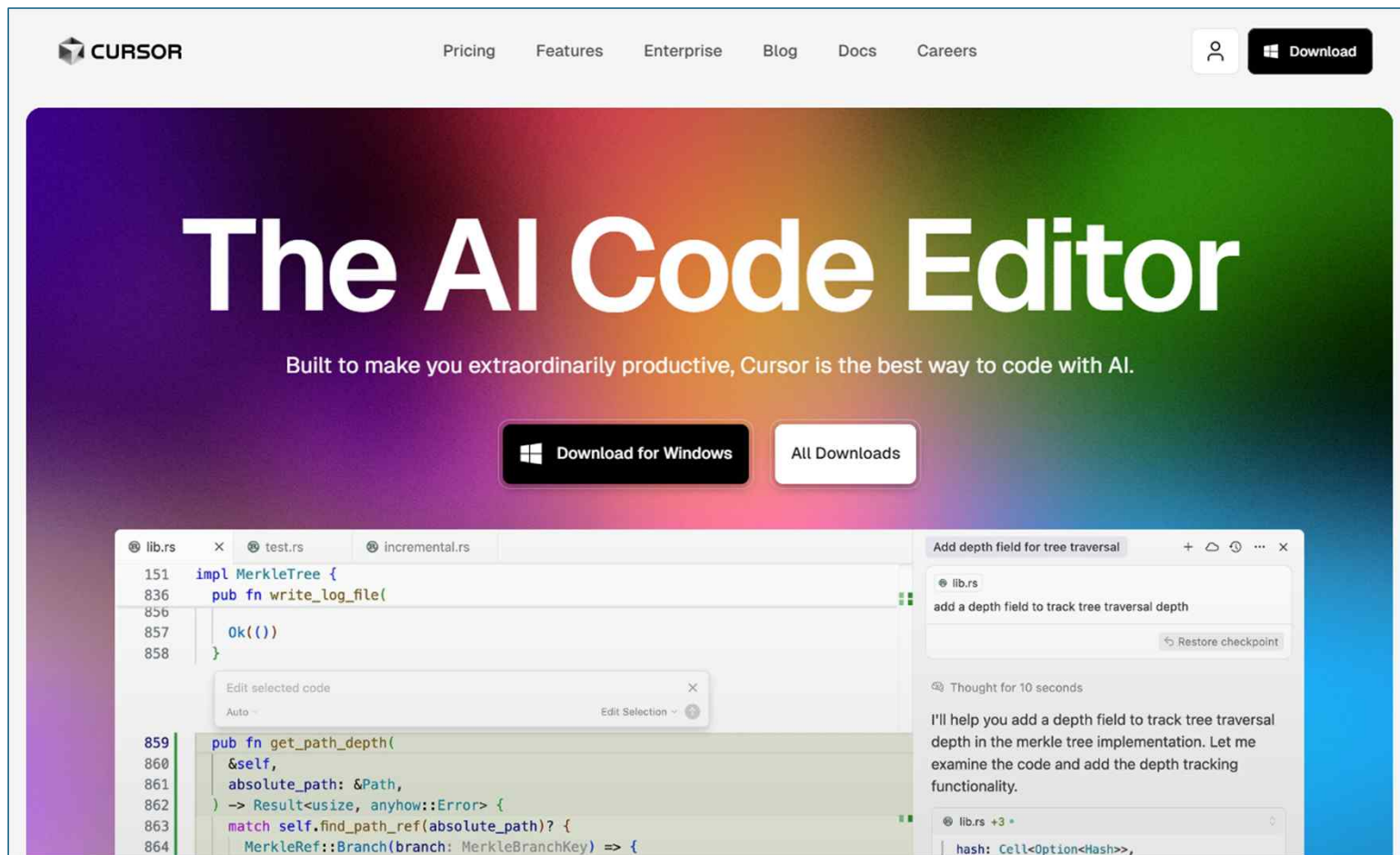


```
C:\Users\trimu\OneDrive\python_uv_test>uv init python-test
Adding `python-test` as member of workspace `C:\Users\trimu\OneDrive\python_uv_test`
Initialized project `python-test` at `C:\Users\trimu\OneDrive\python_uv_test\python-test`

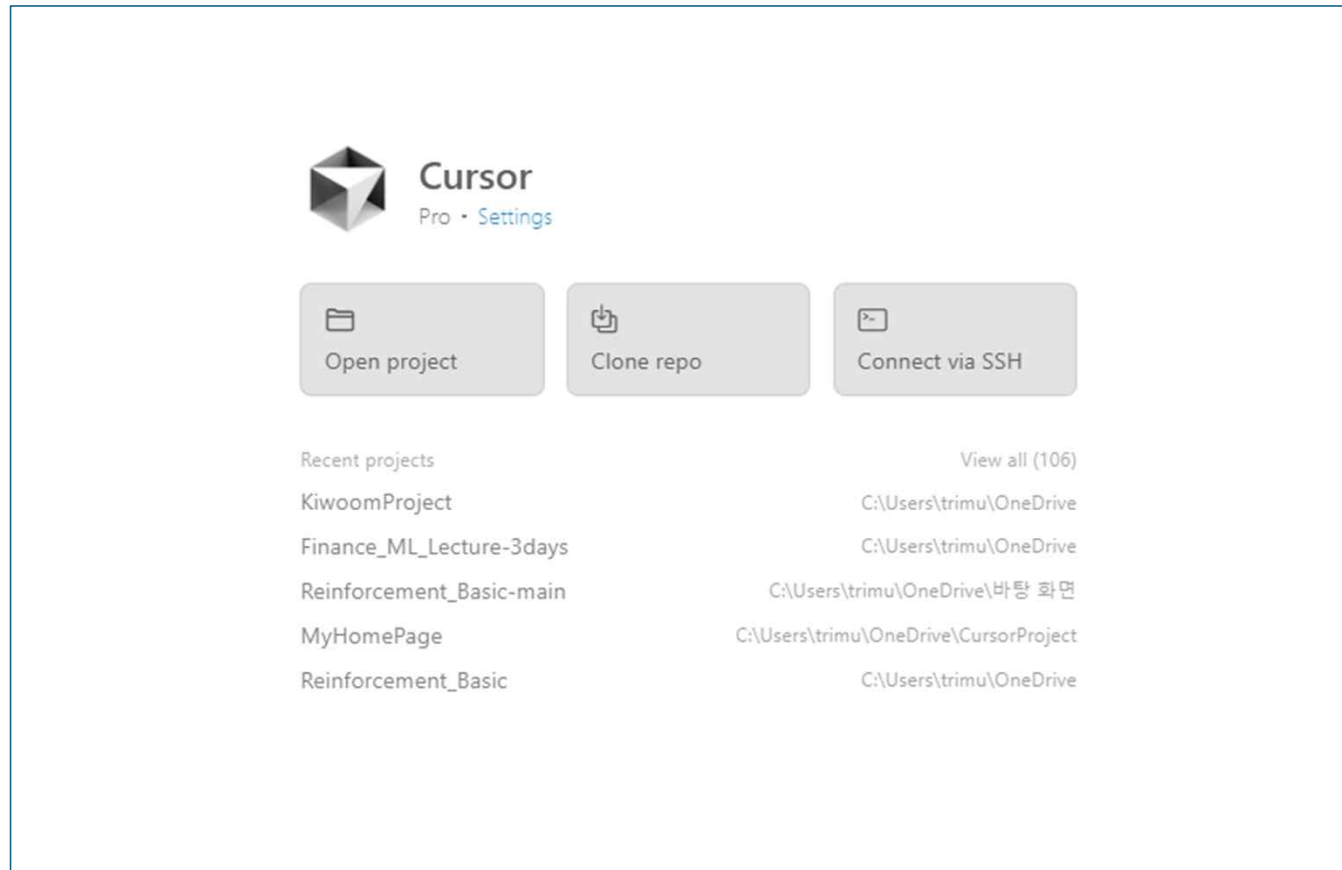
C:\Users\trimu\OneDrive\python_uv_test>
```



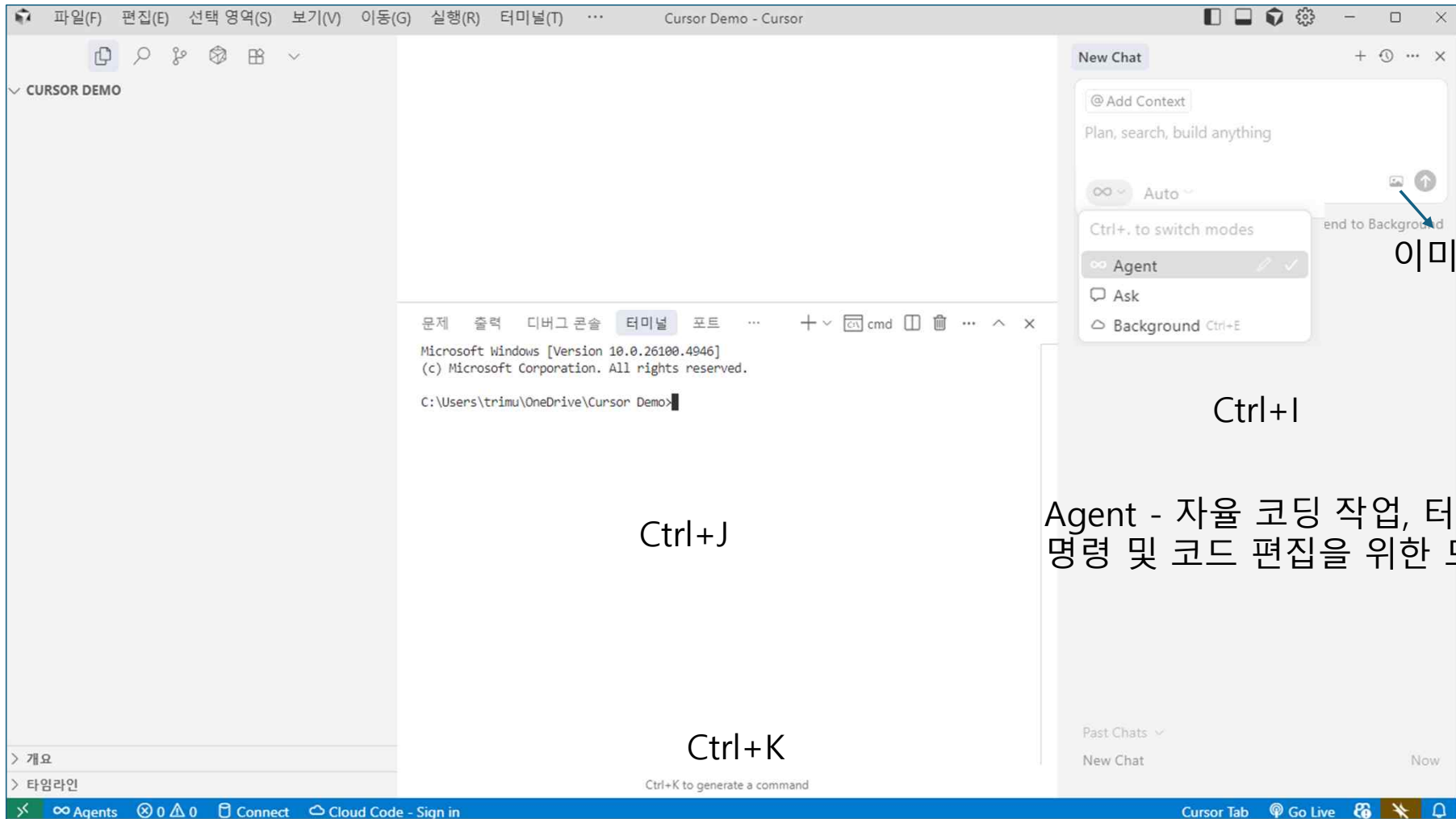
# Cursor AI 설치



# New Project Open







Ctrl+J

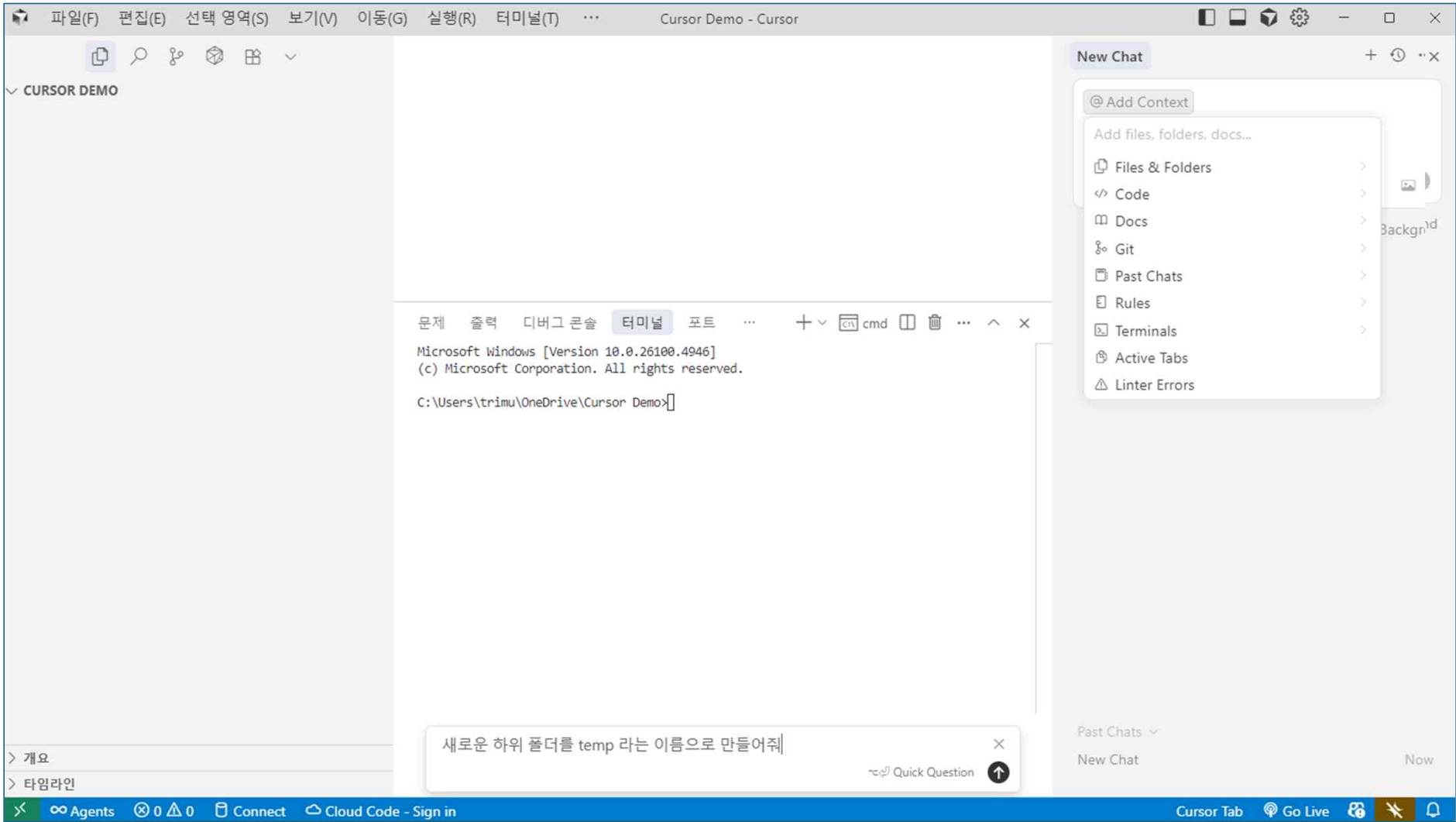
Ctrl+K

Ctrl+K to generate a command

이미지 추가

Ctrl+I

Agent - 자율 코딩 작업, 터미널 명령 및 코드 편집을 위한 도우미



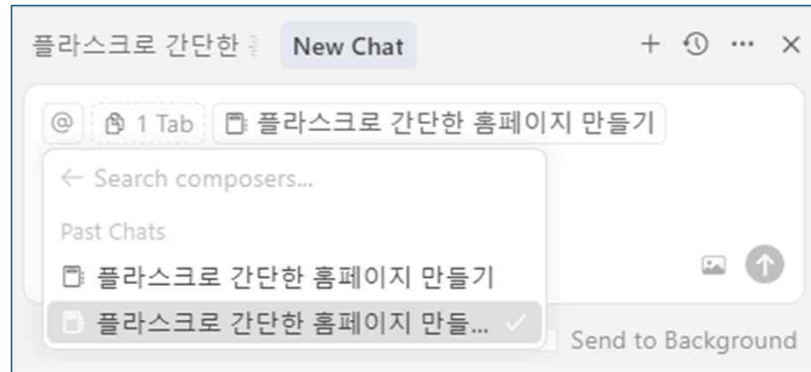
#### ◆ Add Context **의 역할**

- AI가 코드를 이해하거나 답변할 때, 추가 자료를 함께 고려하도록 연결해주는 기능
  - 기본적으로 AI는 현재 열려 있는 코드 파일만 이해하지만, 이 기능을 쓰면 다른 코드·문서·환경을 대화에 포함시킬 수 있음
- 

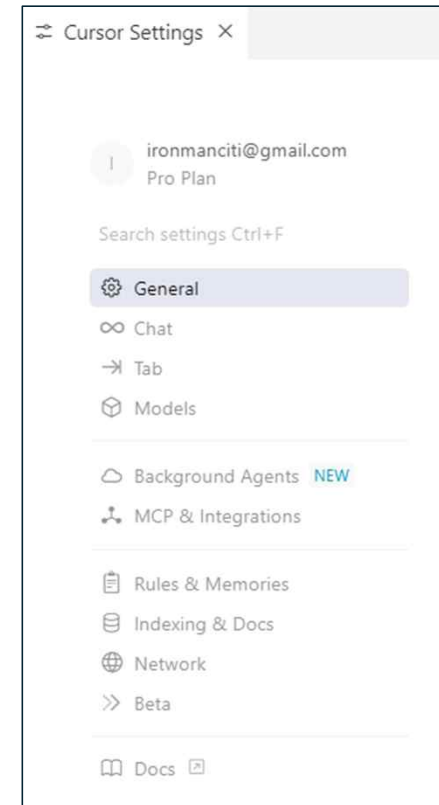
#### ◆ **선택 가능한 항목**

- **Files & Folders:** 특정 프로젝트 폴더/파일을 통째로 AI에게 전달 → “이 코드 베이스를 참고해서 답변해줘”
- **Code:** 개별 코드 스니펫 지정 → 함수나 클래스 단위로 맥락 제공
- **Docs:** 프로젝트 내 문서(README, API 문서 등)를 연결해 참고
- **Git:** Git 로그/브랜치/커밋 메시지 같은 버전 이력 참고
- **Past Chats:** 이전 대화 기록을 불러와 이어서 활용
- **Rules:** 코딩 규칙, 스타일 가이드 같은 개발 원칙 제공
- **Terminals:** 터미널 출력(에러 로그 등)을 AI에게 전달
- **Active Tabs:** 현재 열려 있는 파일 탭 전체를 Context로 제공
- **Lint Errors:** 린터(Linter)가 잡아낸 에러/경고를 참고

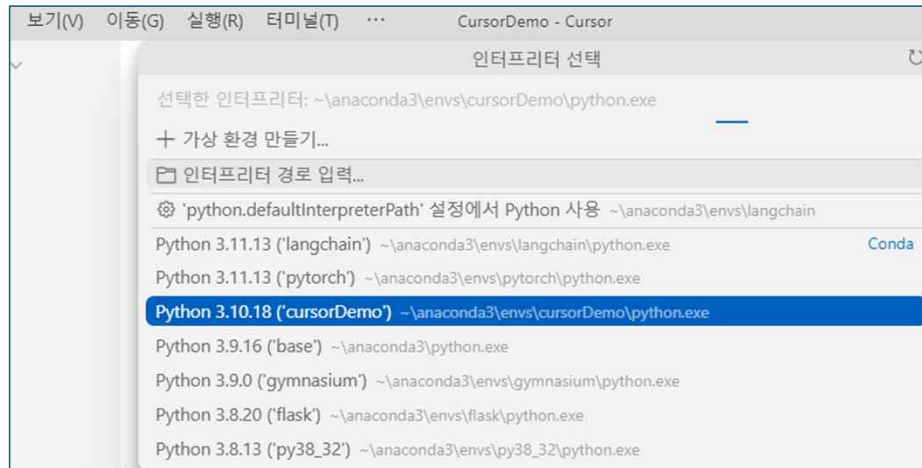
## 이전 Chat History 가져오기



## Cursor 설정



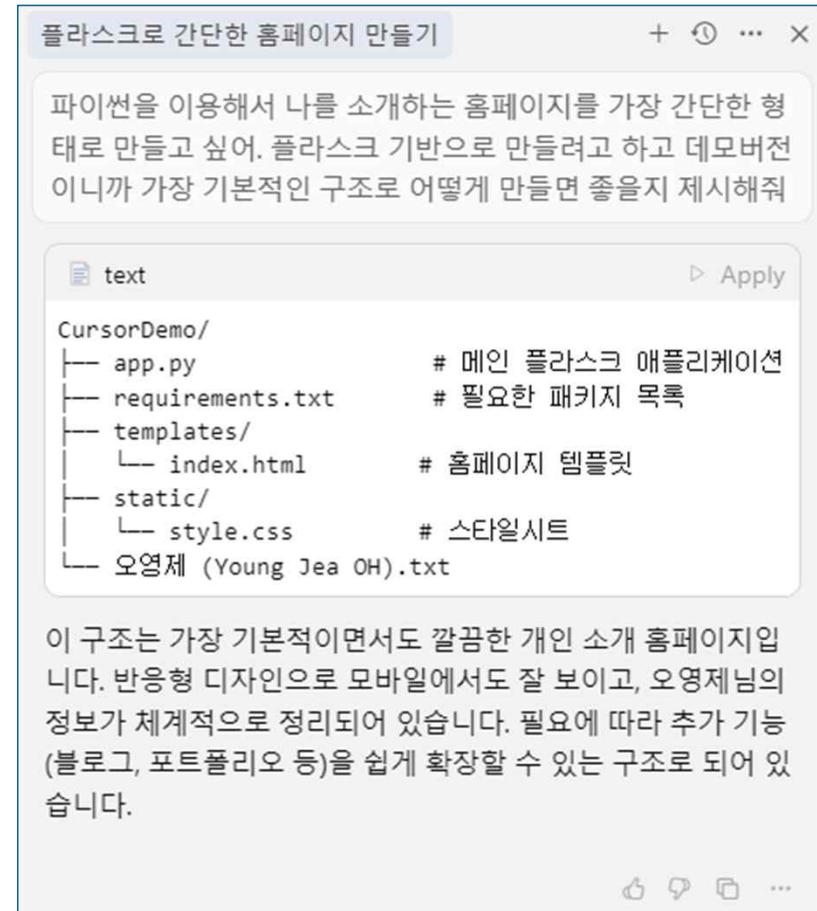
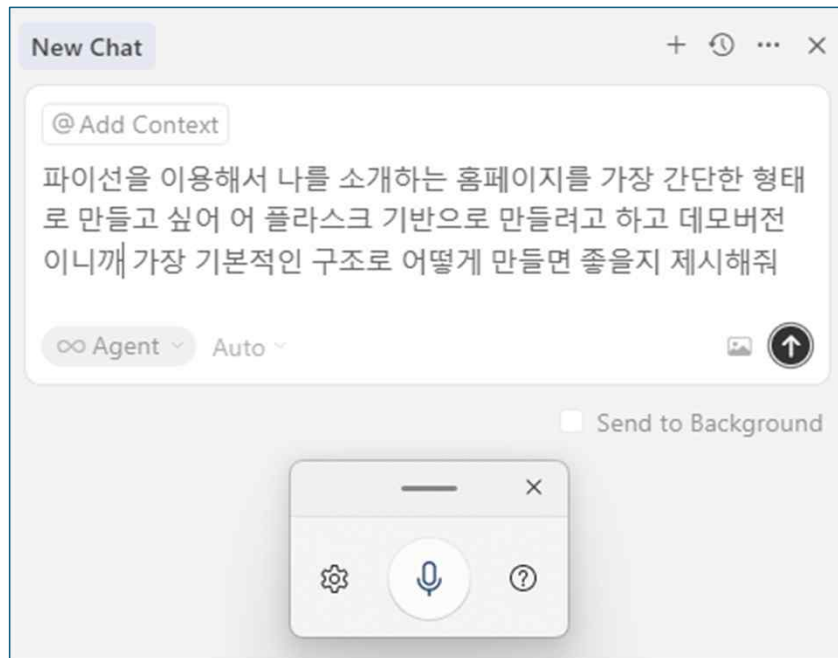
## 파이썬 경로 지정

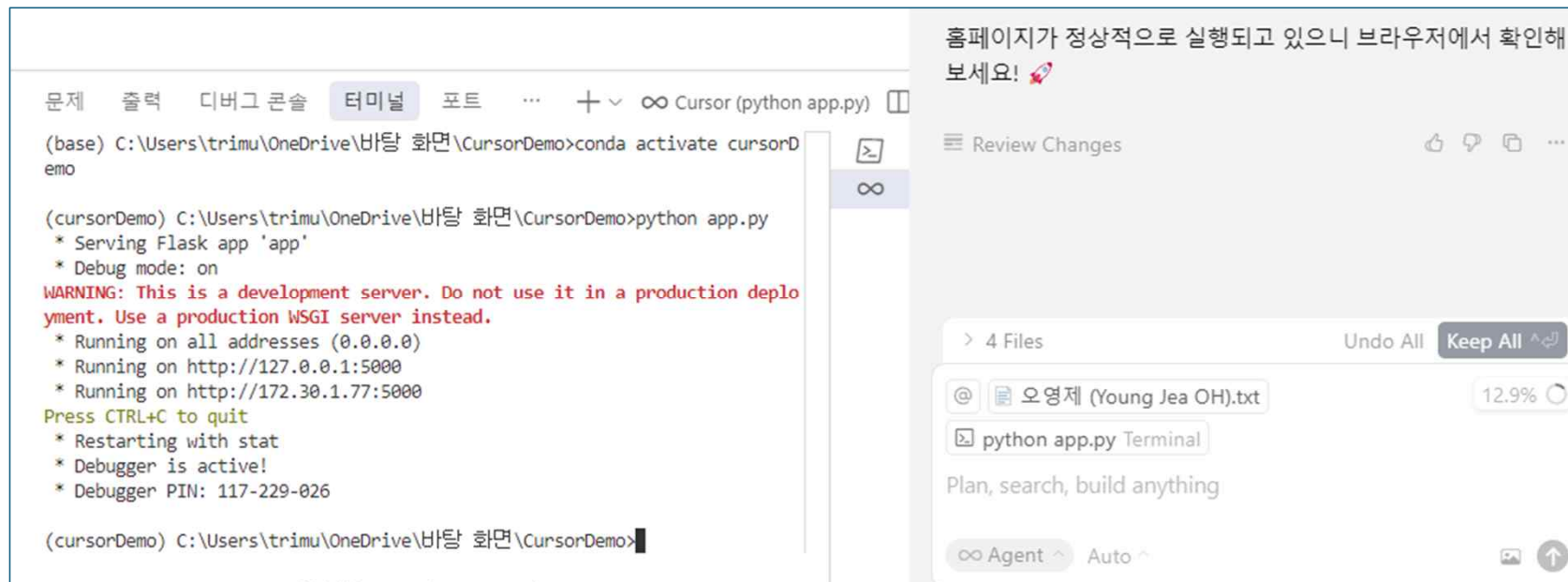
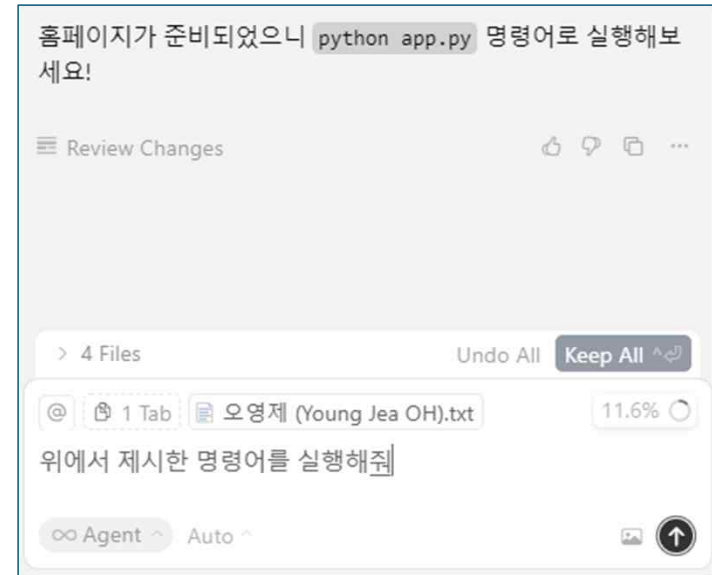
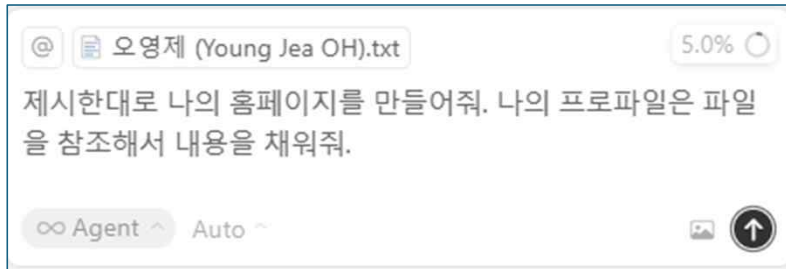


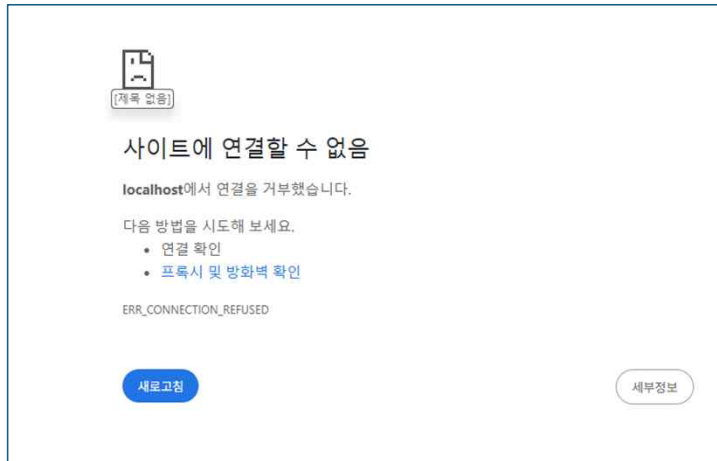
# 실습 : 개인 소개 웹사이트 만들기

1. 새로운 프로젝트 폴더 생성 - MyHomePage
2. Flask 기반으로 작성
3. 기본 틀 제작 기준 - HTML + Tailwind CSS
4. 개인 정보 추가
5. 간단한 game 추가
6. 디자인 조정 → 랜딩페이지 추가

# 음성 입력 - Window+H







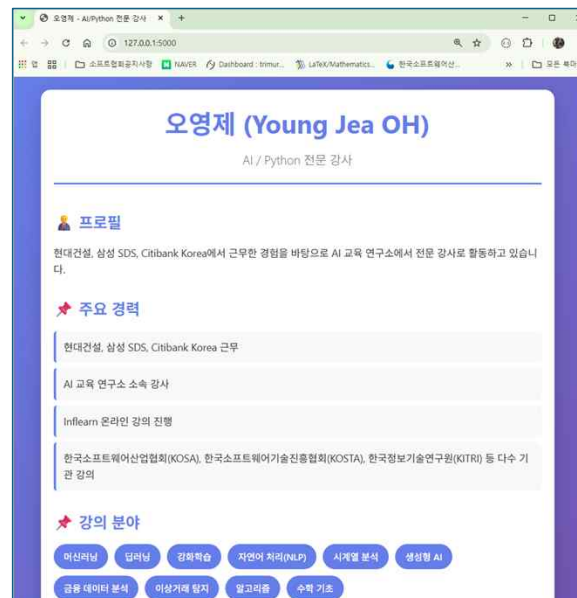
오류화면  
copy/paste

라우저에서 연결이 거부되는 오류가 발생했네요. 이는 서버가  
백그라운드에서 실행되고 있어서 발생한 문제일 수 있습니다.  
서버를 다시 실행해보겠습니다:

```
python app.py

Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 117-229-026
127.0.0.1 - - [05/Sep/2025 12:34:58] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [05/Sep/2025 12:34:59] "GET /static/style.css
127.0.0.1 - - [05/Sep/2025 12:34:59] "GET /favicon.ico HTTP
```

Open in Terminal Shift+⌘ Skip





이전 version 복구

주요 경력 부분이 너무 넓은 면적을 차지하는데 강의 분야와 순서를 바꿔줘

네! 주요 경력 부분이 너무 넓은 면적을 차지하는 문제를 해결하고, 강의 분야와 순서를 바꿔드리겠습니다.

index.html +10 -10 ✓

완료되었습니다! ✨

### 💎 변경 사항

섹션 순서 변경:

1. 🧑💎 프로필 (기존)
2. 📌 강의 분야 (이제 두 번째로 이동)
3. 📌 주요 경력 (이제 세 번째로 이동)
4. 📌 연락처 & 채널 (기존)



오영제 (Young Jea OH)

AI / Python 전문 강사

### 프로필

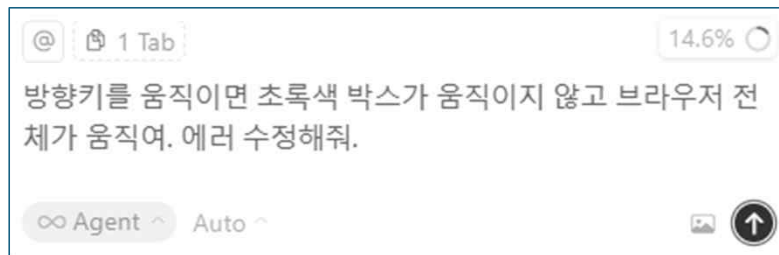
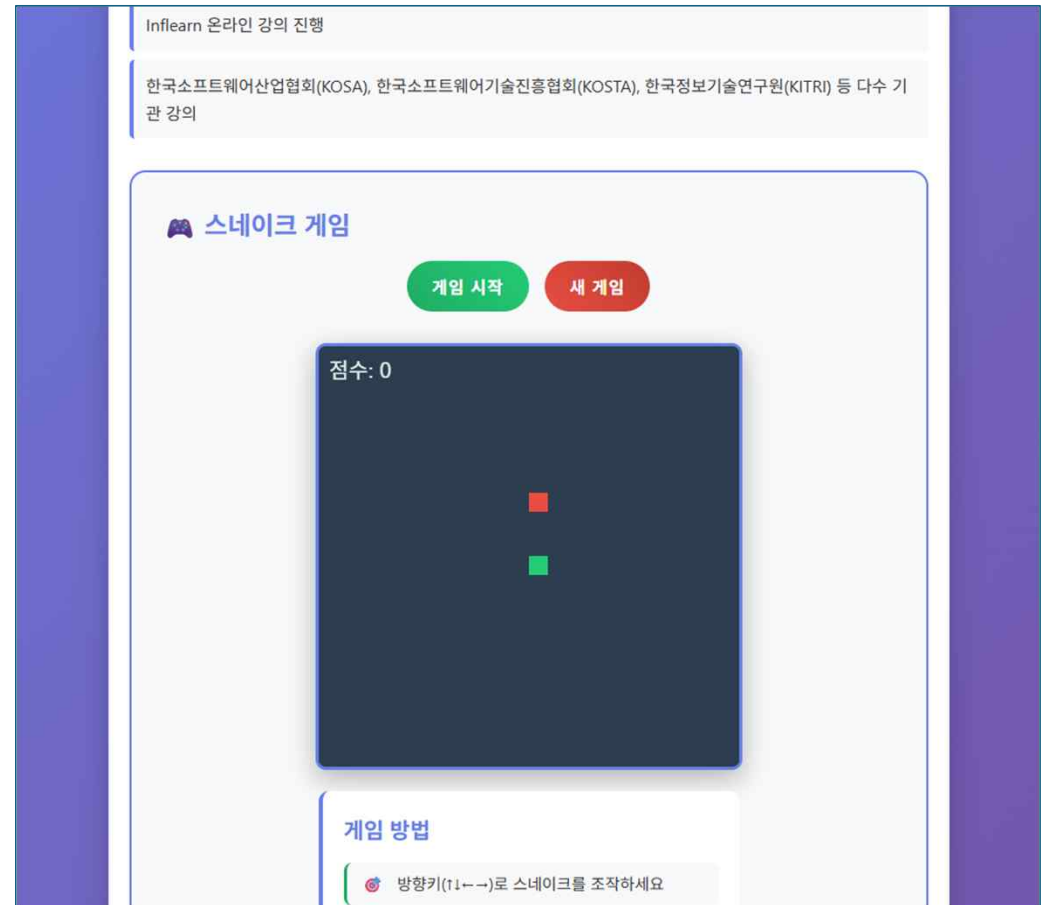
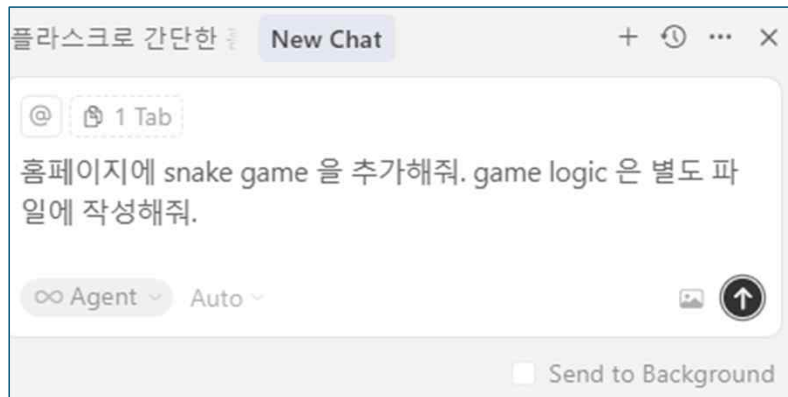
현대건설, 삼성 SDS, Citibank Korea에서 근무한 경험을 바탕으로 AI 교육 연구소에서 전문 강사로 활동하고 있습니다.

### 강의 분야

머신러닝 딥러닝 강화학습 자연어 처리(NLP) 시계열 분석 생성형 AI  
금융 데이터 분석 이상거래 탐지 알고리즘 수학 기초

### 주요 경력

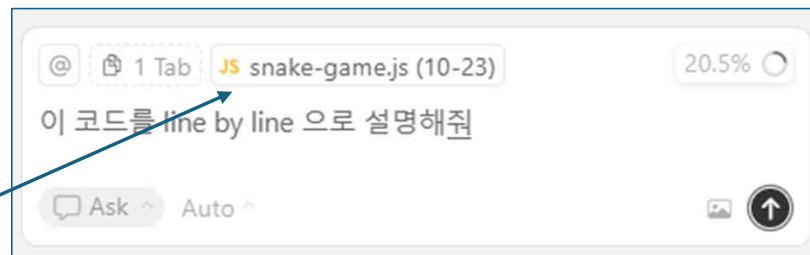
- 현대건설, 삼성 SDS, Citibank Korea 근무
- AI 교육 연구소 소속 강사
- Inflearn 온라인 강의 진행
- 한국소프트웨어산업협회(KOSA), 한국소프트웨어기술진흥협회(KOSTA), 한국정보기술연구원(KITRI) 등 다수 기관 강의



```
// 스네이크 초기 설정
this.snake = [
  {x: 10, y: 10}
];
this.dx = 0;
this.dy = 0;

// 음식 위치
this.food = this.generateFood();

// 게임 상태
this.gameRunning = false;
this.score = 0;
this.gameSpeed = 150;
```



### 라인 23: 게임 속도 설정

JS javascript

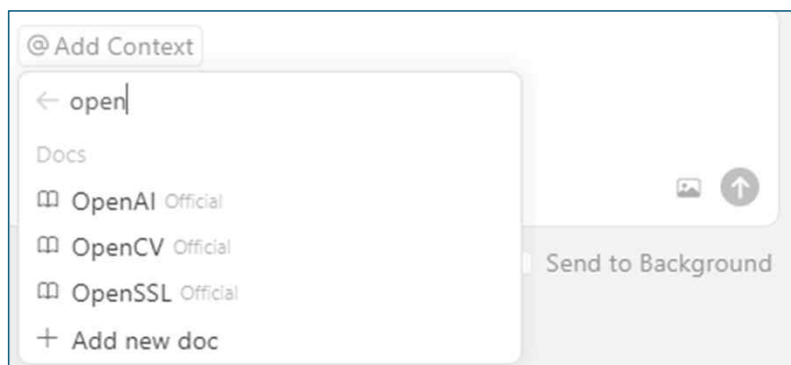
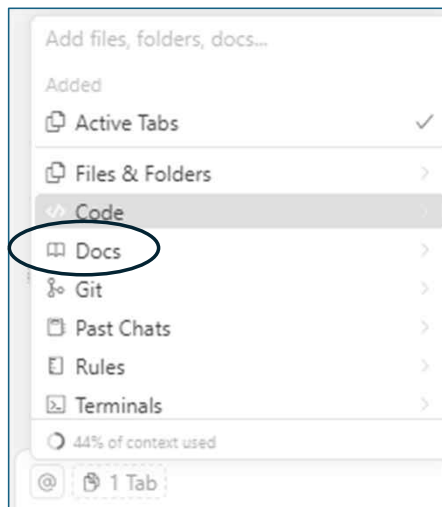
Apply to snake-game.j...

```
this.gameSpeed = 150;
```

- `this.gameSpeed`: 게임의 업데이트 속도를 밀리초 단위로 설정
- `150`: 150밀리초(0.15초)마다 게임이 한 번씩 업데이트됨
- 값이 작을수록 게임이 빨라집니다
- 점수가 올라갈수록 이 값을 줄여서 게임을 더 빠르게 만들 수 있습니다

### 전체적인 의미

이 코드 블록은 스네이크 게임의 초기 상태를 설정하는 부분으로, 게임이 시작되기 전의 모든 기본값들을 정의합니다. 스네이크는 중앙 위치에 머리 하나만 있는 상태로 시작하고, 아직 움직이지 않으며, 점수는 0점, 게임은 정지 상태로 설정됩니다.



openai document 에서 responses api 사용법을 알려줘

## 실제 사용 예시

### 날씨 정보 조회

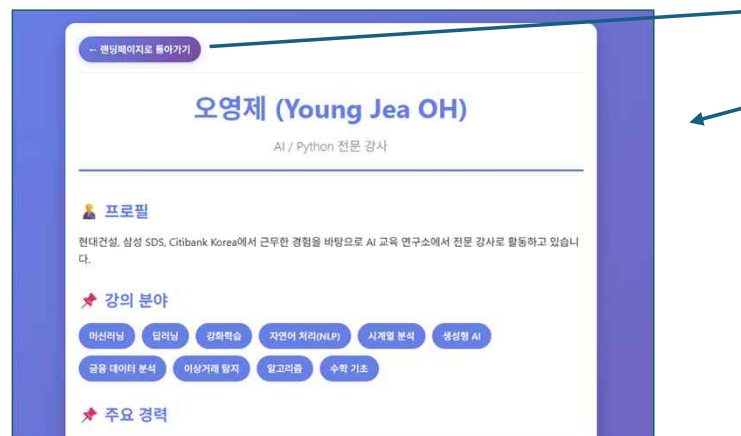
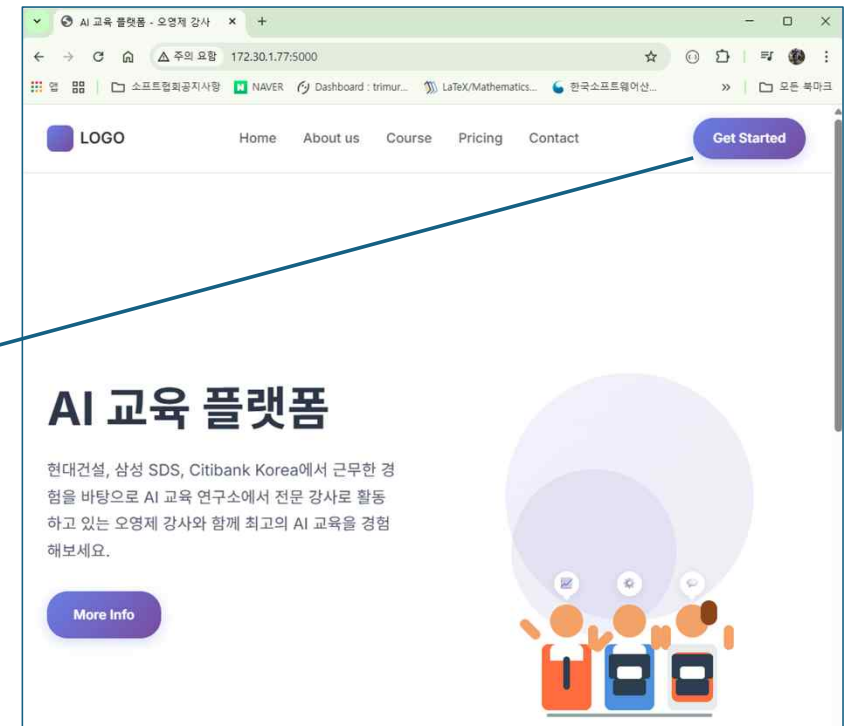
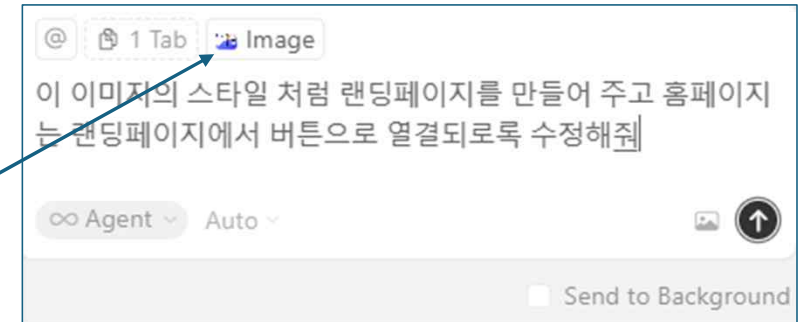
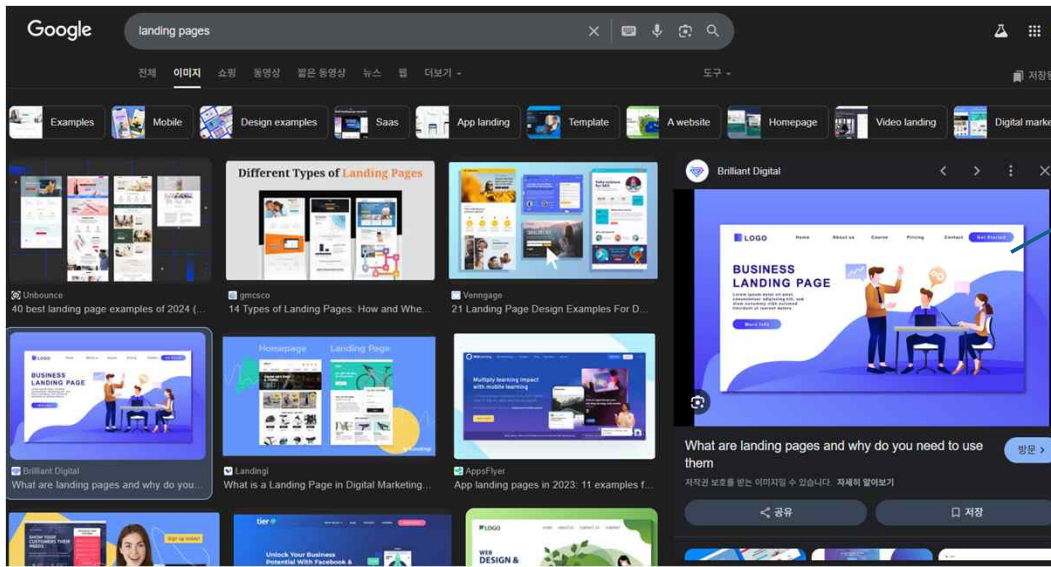
```
python ▶ Apply to index.html

response = client.responses.create(
    model="gpt-4o",
    input="서울의 현재 날씨를 알려주세요",
    tools=[{"type": "web_search"}]
)
print(response.output_text)
```

### 이미지 분석

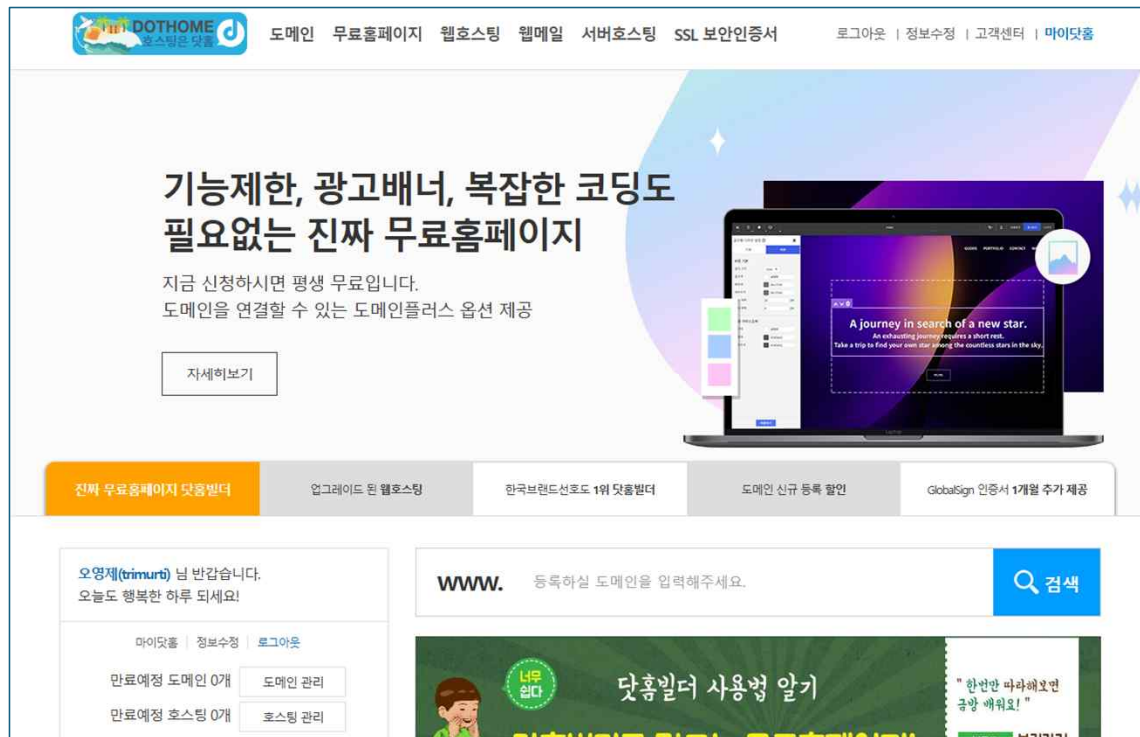
```
python ▶ Apply to index.html

response = client.responses.create(
    model="gpt-4o",
    input="이 이미지에 무엇이 보이나요?",
    tools=[{"type": "image_analysis"}]
)
```



# Web Hosting 서비스 연결

- [dothome.co.kr](https://dothome.co.kr) 가입





[도메인](#) [무료홈페이지](#) [웹호스팅](#) [웹메일](#) [서버호스팅](#) [SSL 보안인증서](#)

[무료호스팅](#) **NEW**

[이미지호스팅](#)

[쇼핑몰 이미지호스팅](#)

[웹호스팅](#) **NEW**

[단독 웹호스팅](#)

[워드프레스 웹호스팅](#)



[도메인](#) [무료홈페이지](#) [웹호스팅](#) [웹메일](#) [서버호스팅](#) [SSL 보안인증서](#)

[로그아웃](#) | [정보수정](#) | [고객센터](#)

## 무료호스팅

국내에서 가장 많은 트래픽과 용량을 제공하는 무료호스팅 서비스를 이용해보세요

[신청하기](#) [도메인플러스](#) [특징](#) [추천 서비스](#) [사양](#) [자주 묻는 질문](#)

## 무료호스팅

### 기본사양

무료

디스크 500M

트래픽(월) 15G

기본 도메인 제공 ②

DB 무제한

[신청하기](#)

도메인
무료홈페이지
웹호스팅
웹메일
서버호스팅
SSL 보안인증서
로그아웃
정보수정
고객센터
마이닷컴

01. 서비스 선택

02. 이용안내 확인

03. 서비스 신청 및 결제

04. 완료

1

결제 환경 안내

브라우저 - Internet Explorer 9 이상 / Chrome / Safari / Firefox / Opera  
OS - Windows / Mac (신한, 삼성, 현대, 하나, 롯데, NH) 결제 가능  
(Mac 환경의 Safari 브라우저에서 결제가 진행 안될 시 쿠키 설정과 자바스크립트 설정을 허용해주시기 바랍니다)

2

신청 서비스 정보

선택 서비스	무료호스팅
이용기간	6개월
만료 예정일	2026-02-11
트래픽	15G/월
디스크	500M
도메인 연결	1계정 / 최대0개

3

웹호스팅 설정 정보 입력

FTP 아이디	trimurti	중복확인
FTP 비밀번호 ?	*****	
기본제공 도메인	trimurti.dothome.co.kr	
PHP 버전	8.4	
DB명	trimurti	
DB아이디	trimurti	
DB 비밀번호 ?	*****	
CMS 자동설치	설치 안함	



이메일 인증





도메인 무료홈페이지 웹호스팅 웹메일 서버호스팅 SSL 보안인증서

로그아웃 | 정보수정 | 고객센터 | 마이닷홈

## 도메인 신규 등록 할인

할인쿠폰으로 가격부담없이 신규 도메인을 신청해보세요!  
(net/org/biz/info/pe.kr 제외)



### ➡ 웹호스팅 목록

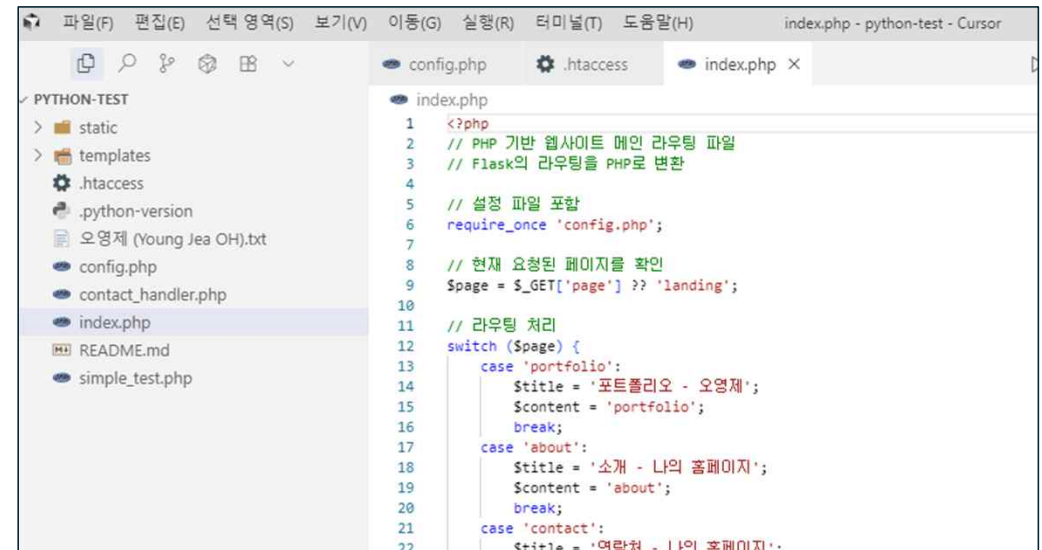
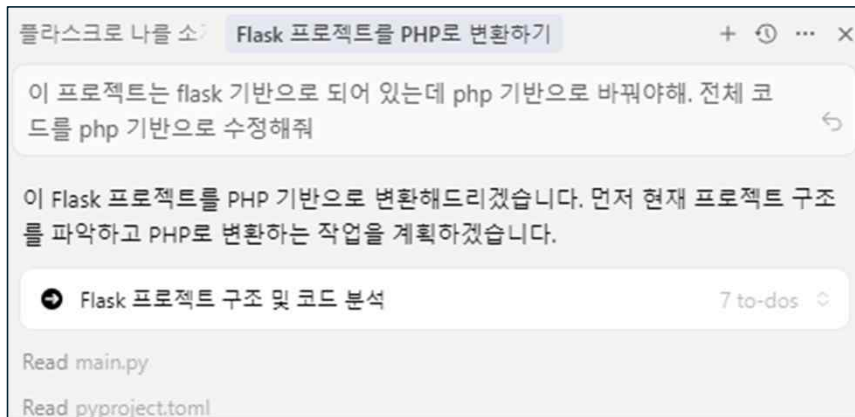
만료일순 정렬 ▼ 전체 ▼ 계정 ID ▼  검색

연장하기 서비스 변경 도메인 연결

내 홈페이지 연결

선택	계정 ID	서비스 종류	도메인	옵션	만료일	잔여일	상태	기타
<input type="radio"/>	trimurti	무료호스팅	trimurti.dothome.co.kr [112.175.185.138]		2025-12-04	114	사용	<a href="#">상세보기</a> <a href="#">유료전환</a>

# Flask → php 기반으로 변경



FileZilla download (<http://filezilla-project.org/>)



**FileZilla** The free FTP solution

Home

**FileZilla**

- Features
- Screenshots
- Download
- Documentation
- FileZilla Pro
- RemoteDrive

**FileZilla Server**

- Download

**Community**

- Forum
- Wiki

**General**

- FAQ
- Support

## Download FileZilla Client

The latest stable version of FileZilla Client is 3.69.3

Please select the file appropriate for your platform below.

**Windows (64bit x86)** 

- [FileZilla\\_3.69.3\\_win64-setup.exe](#)  (recommended)
- [FileZilla\\_3.69.3\\_win64.zip](#) 

The 64bit versions of Windows 10 and 11 are supported.

**Windows (32bit x86)** 

- [FileZilla\\_3.69.3\\_win32-setup.exe](#)  (recommended)
- [FileZilla\\_3.69.3\\_win32.zip](#) 

The 32bit version of Windows 10 is supported.

**FileZilla**

파일(F) 편집(E) 보기(V) 전송(T) 서버(S) 북마크(B) 도움말(H)

호스트(H): 사용자명(U): 비밀번호(W): 포트(P): 빠른 연결(Q) ▼

웹서버/FTP 정보

▶ 웹서버 아이피	112.175.185.136	▶ 웹서버 정보	Apache 2.4
▶ PHP 버전	PHP 8.4	▶ PHP 버전 변경	8.4 ▼ 변경
▶ 웹루트 디렉토리	/hosting/trimurti/html		
▶ FTP 아이디	trimurti	▶ FTP 암호	변경
▶ FTP 접속설정	<input checked="" type="radio"/> 접속허용 <input type="radio"/> 접속제한 변경		

도메인 정보

▶ 기본 제공 도메인	trimurti.dothome.co.kr	
▶ 연결 도메인		도메인 연결
▶ SSL 보안인증서 적용 도메인		SSL 보안인증서 신청



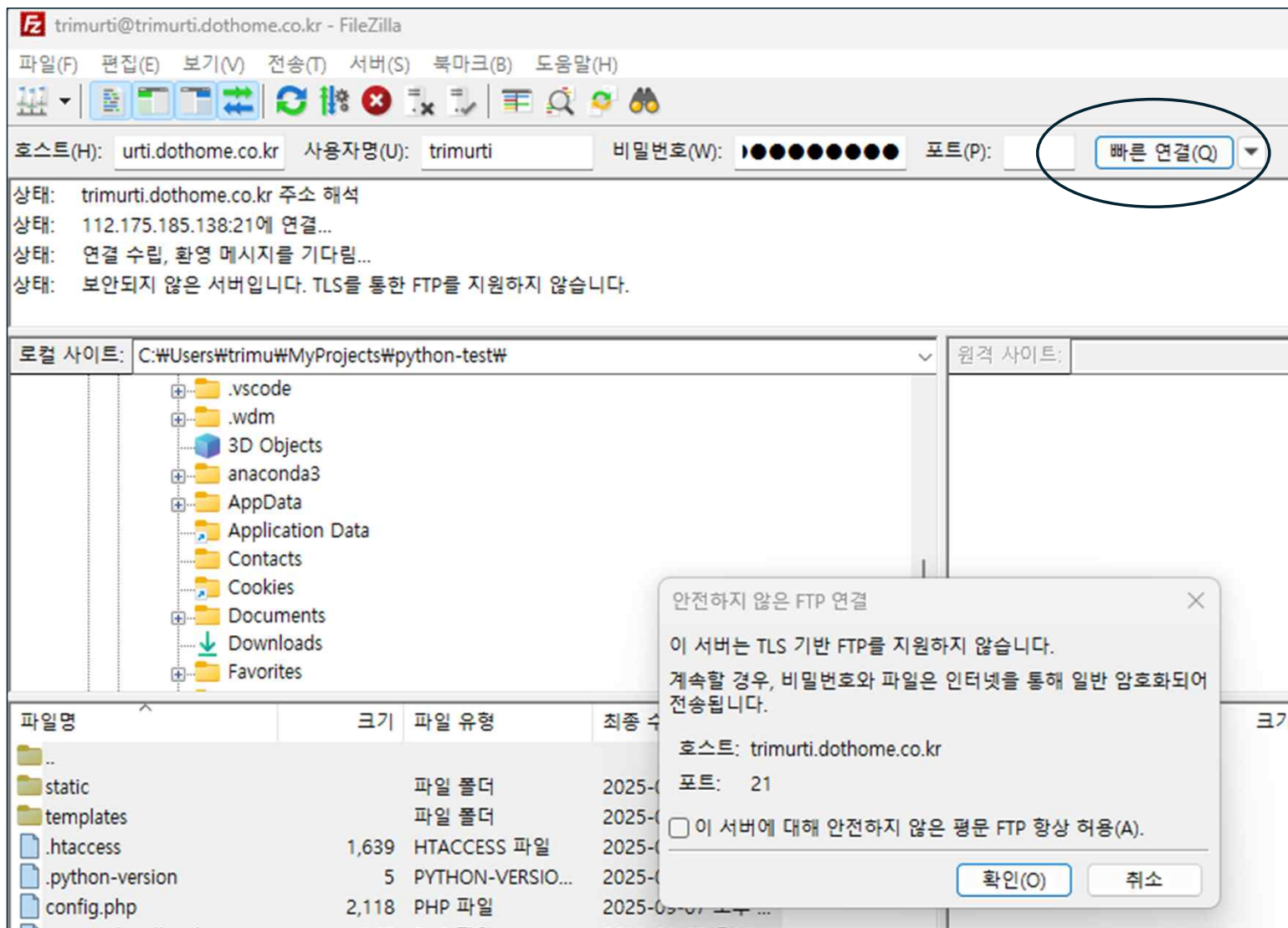
호스트(H): urti.dothome.co.kr 사용자명(U): trimurti 비밀번호(W): ●●●●●●●● 포트(P): 빠른 연결(Q) ▼

상태: 보안되지 않은 서버입니다. TLS를 통한 FTP를 지원하지 않습니다.

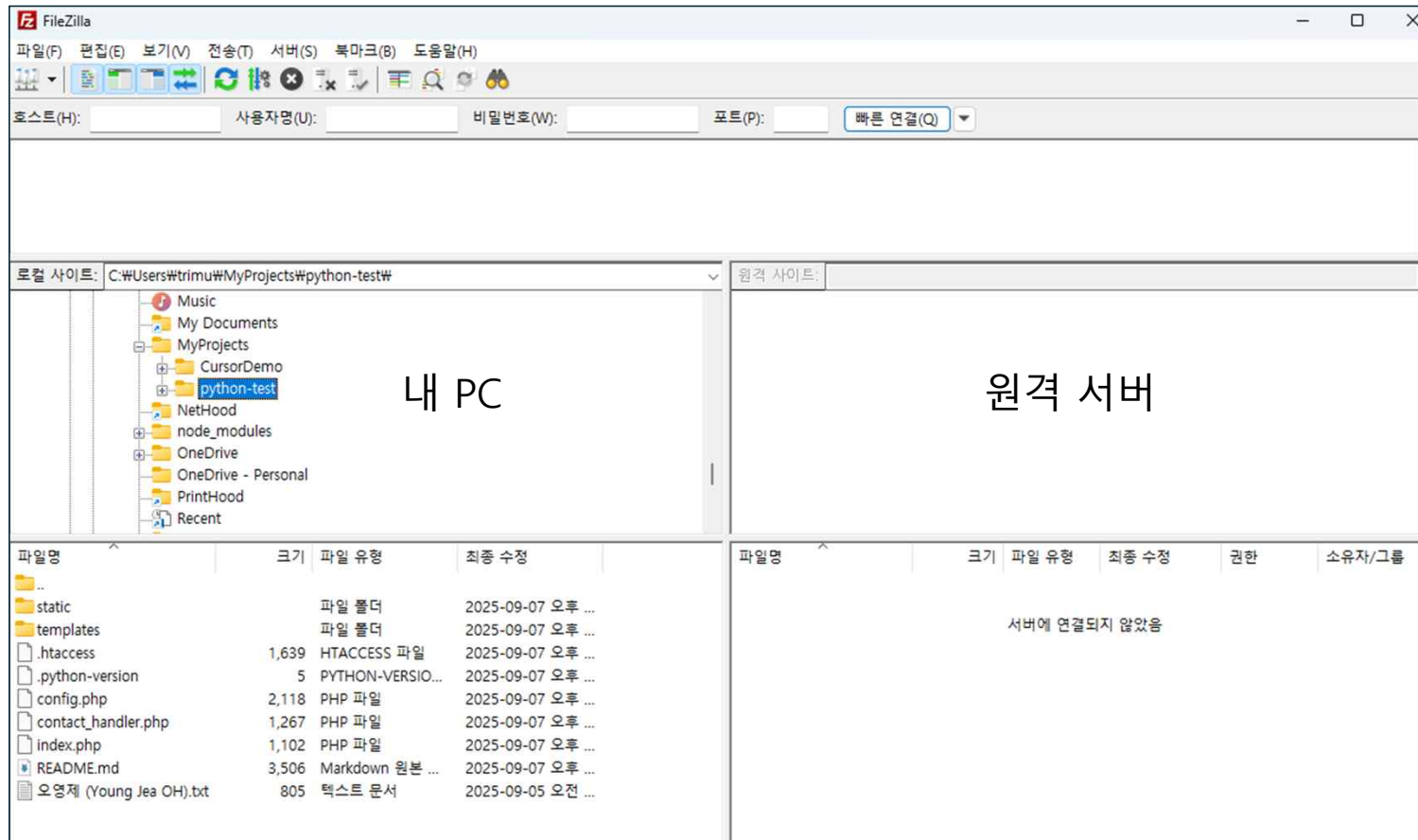
상태: 로그인

상태: 디렉터리 목록 조회...

상태: "/" 디렉터리 목록 조회 성공



# FTP 로 server 전송





trimurti@trimurti.dothome.co.kr - FileZilla

파일(F) 편집(E) 보기(V) 전송(T) 서버(S) 북마크(B) 도움말(H)

호스트(H): urti.dothome.co.kr 사용자명(U): trimurti 비밀번호(W): [mask] 포트(P): 빠른 연결(Q)

상태: 파일 전송 성공, 5,060 바이트를 1 초에 전송  
 상태: 파일 전송 성공, 3,404 바이트를 1 초에 전송  
 상태: 서버와의 연결이 종료됨  
 상태: 서버와의 연결이 종료됨

로컬 사이트: C:\Users#trimu#MyProjects#python-test#

- python-test
  - NetHood
  - node\_modules
  - OneDrive
  - OneDrive - Personal
  - PrintHood
  - Recent
  - Saved Games
  - scikit\_learn\_data
  - seaborn-data
  - Searches

원격 사이트: /html

- /
  - .webalizer
  - html

파일명	크기	파일 유형	최종 수정
..			
static		파일 폴더	2025-09-07 오후 ...
templates		파일 폴더	2025-09-07 오후 ...
.htaccess	1,639	HTACCESS 파일	2025-09-07 오후 ...
.python-version	5	PYTHON-VERSIO...	2025-09-07 오후 ...
config.php	2,118	PHP 파일	2025-09-07 오후 ...
contact_handler.php	1,267	PHP 파일	2025-09-07 오후 ...
index.php	1,102	PHP 파일	2025-09-07 오후 ...
README.md	3,506	Markdown 원본 ...	2025-09-07 오후 ...
오영제 (Young Jea OH).txt	805	텍스트 문서	2025-09-05 오전 ...

→

파일명	크기	파일 유형	최종 수정	권한	소유자/그룹
..					
static		파일 폴더	2025-09-07 ...	flcdmpe (0...	trimurti hos...
templates		파일 폴더	2025-09-07 ...	flcdmpe (0...	trimurti hos...
.htaccess	1,584	HTACCESS...	2025-09-07 ...	adfrw (064...	trimurti hos...
.python-version	5	PYTHON-...	2025-09-07 ...	adfrw (064...	trimurti hos...
config.php	2,056	PHP 파일	2025-09-07 ...	adfrw (064...	trimurti hos...
contact_handler.php	1,238	PHP 파일	2025-09-07 ...	adfrw (064...	trimurti hos...
index.php	1,055	PHP 파일	2025-09-07 ...	adfrw (064...	trimurti hos...
README.md	3,506	Markdown...	2025-09-07 ...	adfrw (064...	trimurti hos...
오영제 (Young Jea O...	772	텍스트 문서	2025-09-07 ...	adfrw (064...	trimurti hos...

감사합니다.