

1강

Why Deep Learning?

여러분은 왜 딥러닝을 배우러
오셨나요?

지능(Intelligence)이란?

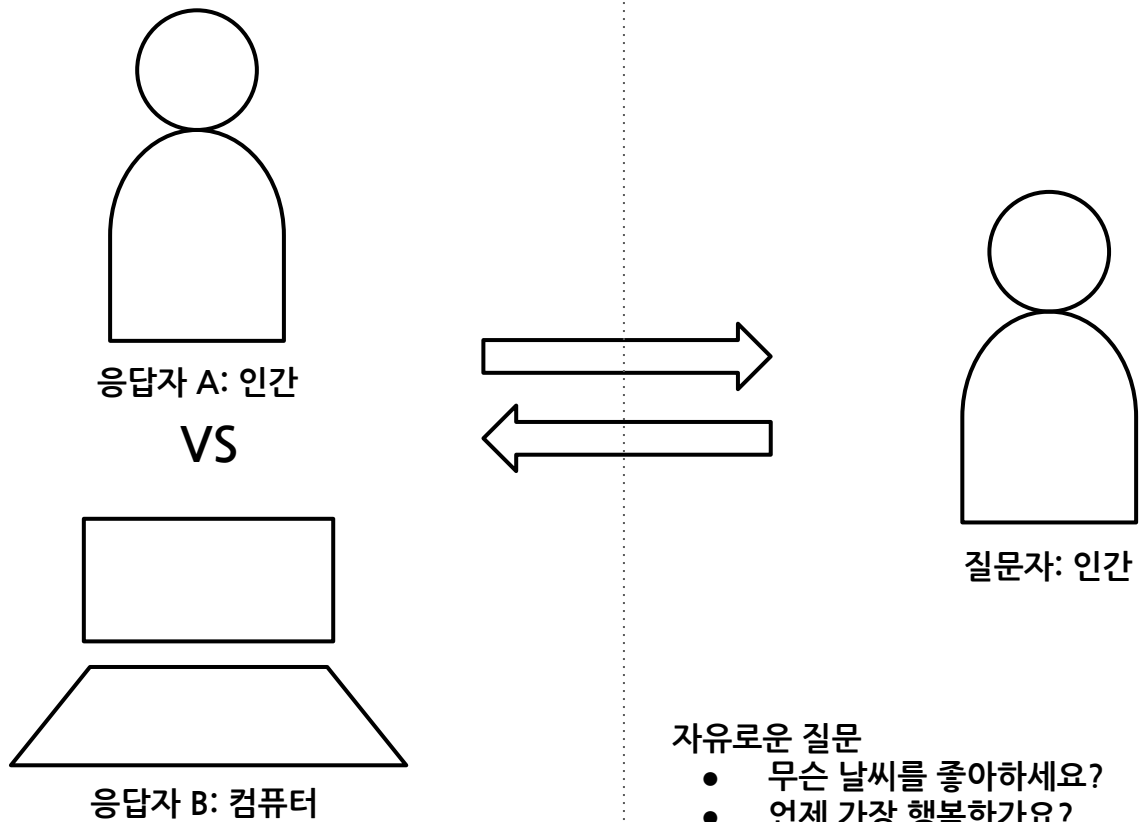
미래의 의사 결정 능력에
영향을 미칠 정보를 처리하는
능력

기계가 지능을 가질 수 있을까?

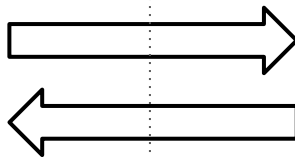
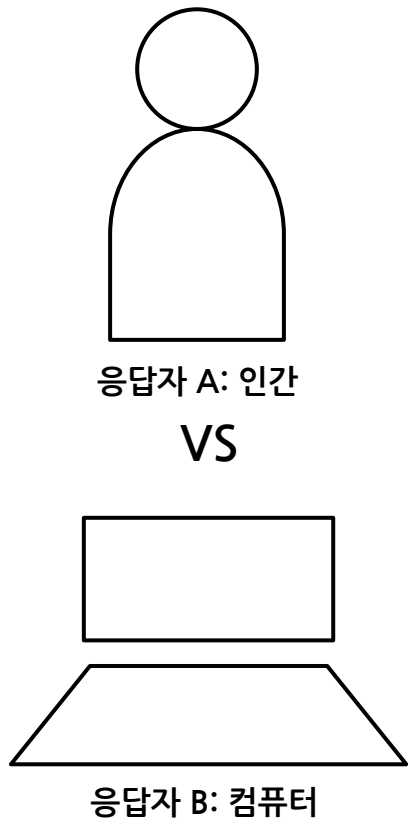
Intro.

튜링 테스트

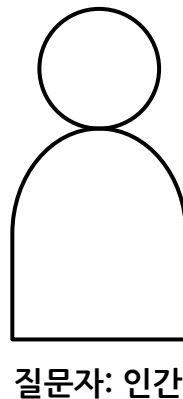
튜링 테스트



튜링 테스트



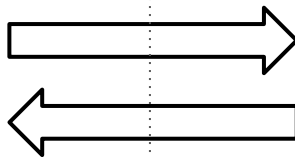
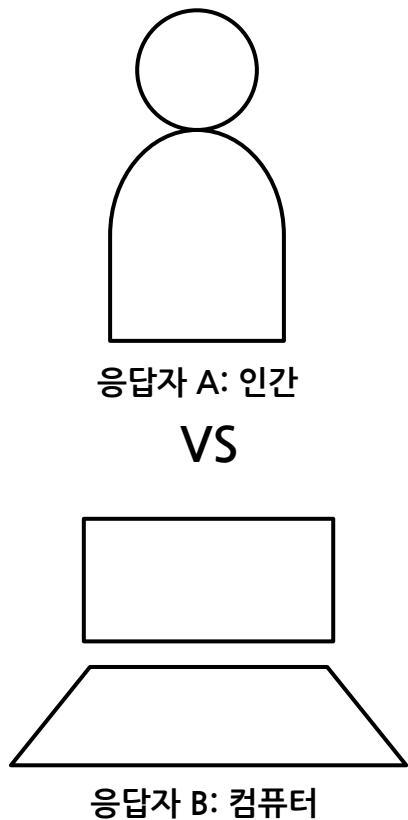
튜링 테스트 통과
X



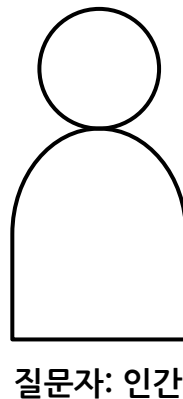
판단

- A는 인간이다.
- B는 기계다.

튜링 테스트



튜링 테스트 통과
O



판단

- A는 기계다.
- B는 인간이다.

Deep Learning 이란?

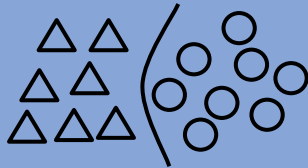
Artificial Intelligence

컴퓨터가 인간처럼
사고하고, 문제를
해결하며, 학습할 수 있게
만드는 모든 기술



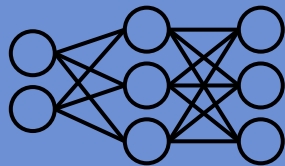
Machine Learning

사람이 명확한 규칙을
정해주지 않아도 컴퓨터가
데이터를 통해 패턴을
찾아냄



Deep Learning

여러 층으로 이루어진 인공신경망을
사용해 이미지, 음성, 자연어 같은
복잡한 데이터를 효과적으로 처리



1997



IBM의 딥블루가 체스 세계 챔피언 가리
카스파로프를 상대로 승리

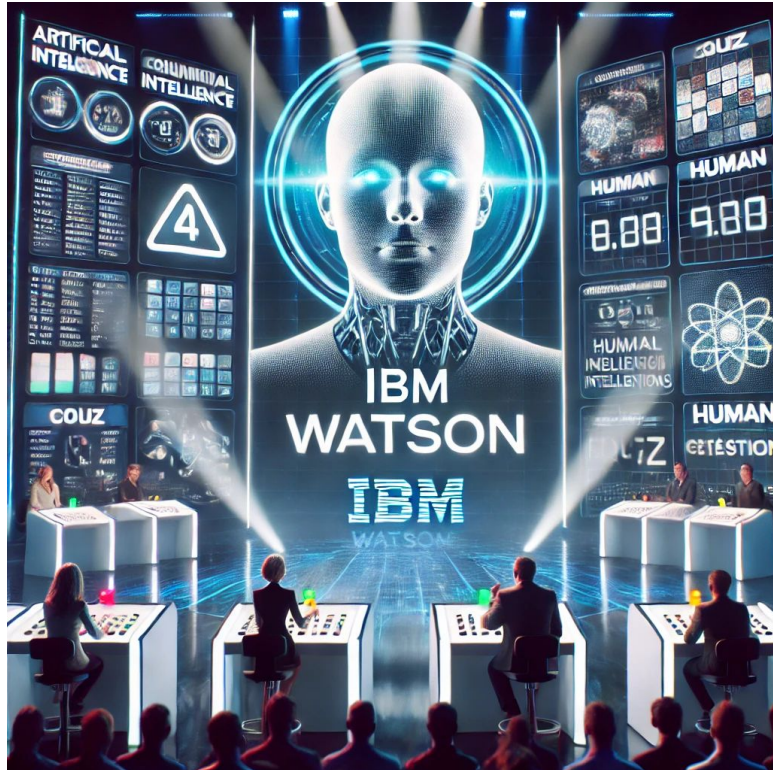
체스 경우의 수

1번째 수 : 400가지

2번째 수 : 19만 7742가지

3번째 수 : 1억 2100가지

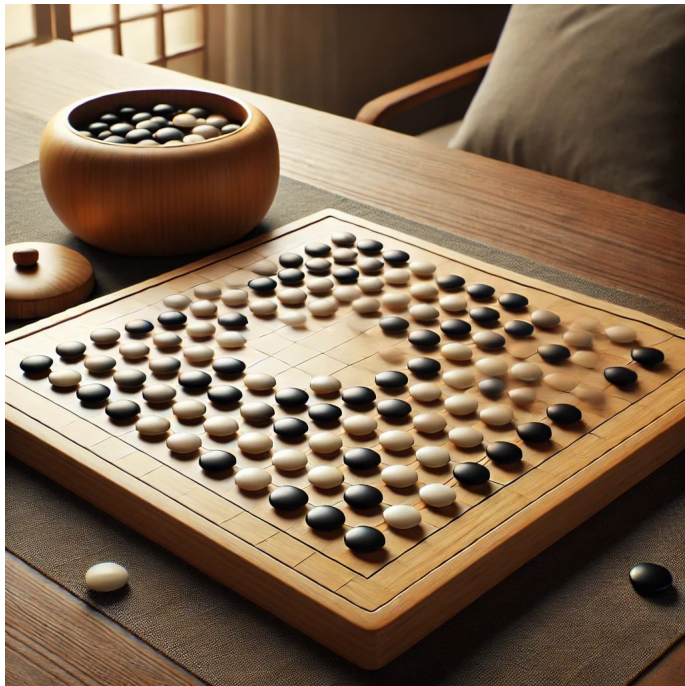
2011



IBM의 왓슨이 퀴즈 쇼 '제퍼디!'에서 인간
챔피언들을 이김

실시간으로 언어를 분석해야하고 인터넷 연결되지
않아 체스보다 어려움

2016



알파고가 이세돌 9단을 상대로 바둑 대국에서

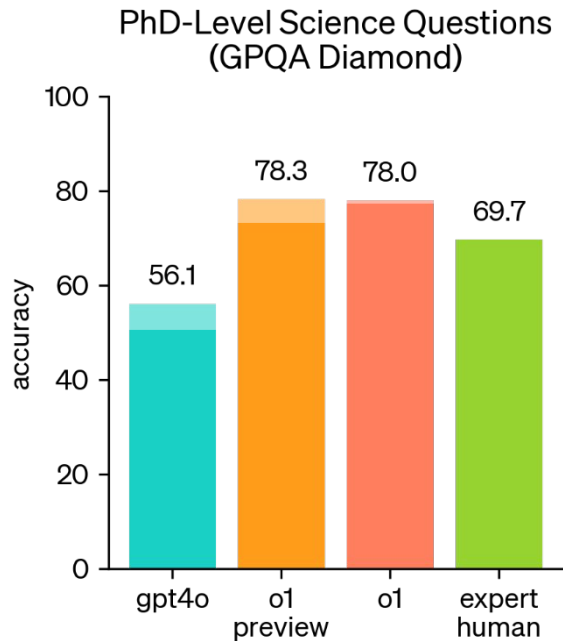
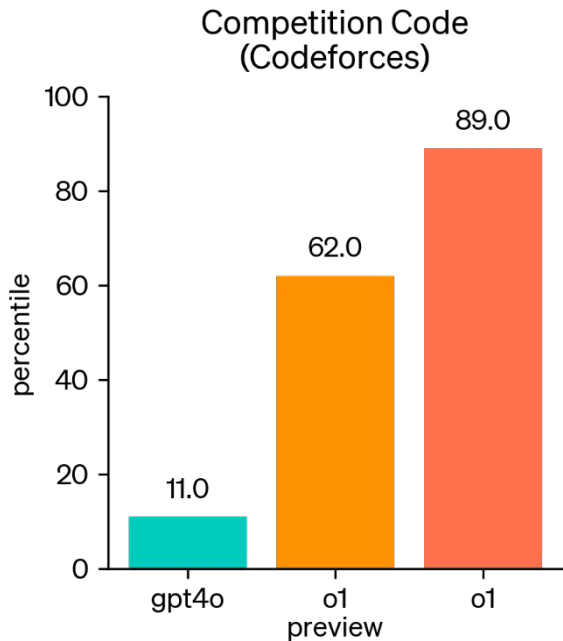
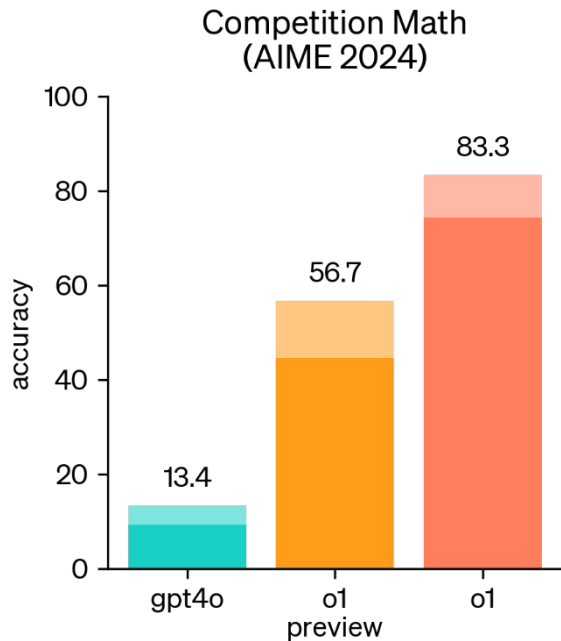
4승 1패로 최종 승리

바둑 경우의 수 (단, 돌을 따낸 경우는 고려하지 않는다.)

=> 361!

바둑이 체스보다 경우의 수가 훨씬

2024



<https://openai.com/index/learning-to-reason-with-llms/>

OpenAI의 o1 모델은 박사 수준의 물리학, 생명 등의 시험에서 인간을 상대로 승리

2030

AI의 미래는?

Why Deep Learning?

여러분은 왜 딥러닝을 배우러
오셨나요?

Why Deep Learning?

사람들이 겪는 불편한 문제를 해결하기
위하여

사람들이 겪는 불편한 문제를 해결하기 위하여

의료

의사들의 암 진단률의 정확도를 높일 수 있을까?

바이러스 변이가 발견될 때마다 적절한 항체를 즉시 만들 수는 없을까?

사람들이 노화하지 않으려면 어떻게 해야할까?

쇼핑

사이트마다 비교하지 않고 내가 원하는 조건에 딱 맞는 제품을 바로 알 수 없을까?

온라인에서 본 옷을 입었을 때 내가 잘 어울릴까?

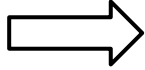
교육

기출 문제만 있고 문제 풀이는 없는데 모범 답안을 알 수 없을까?

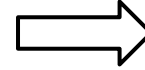
이 수학 문제를 풀기 위해서 필요한 개념을 알 수 없을까?

Deep Learning

Input



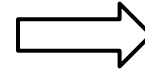
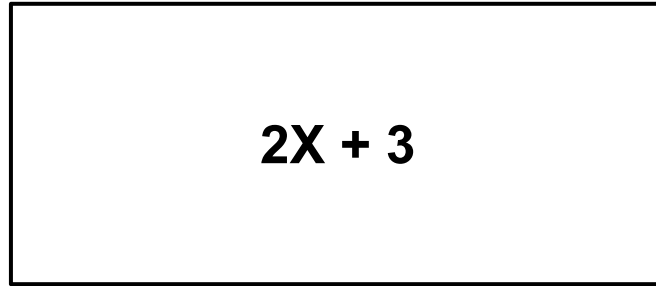
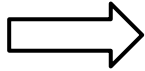
Function



output

Deep Learning

Input

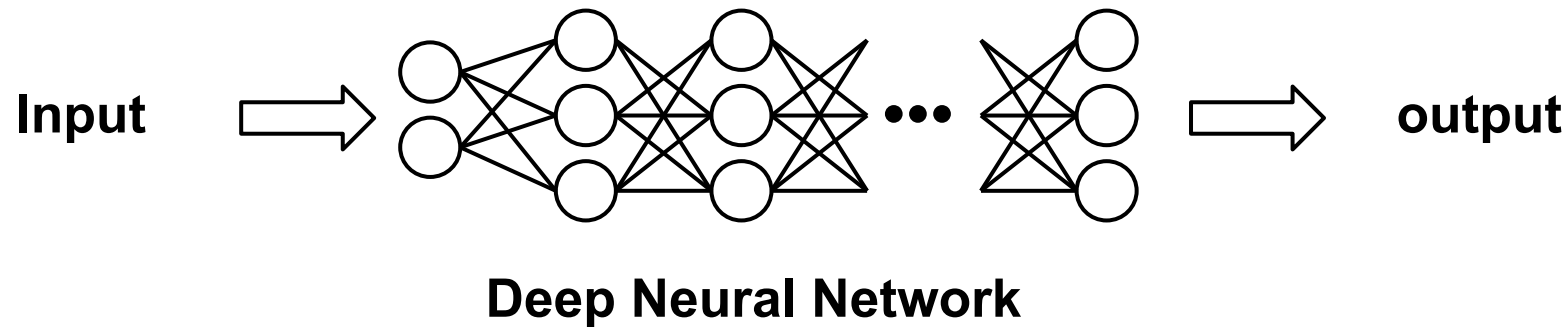


output

$x = 1$

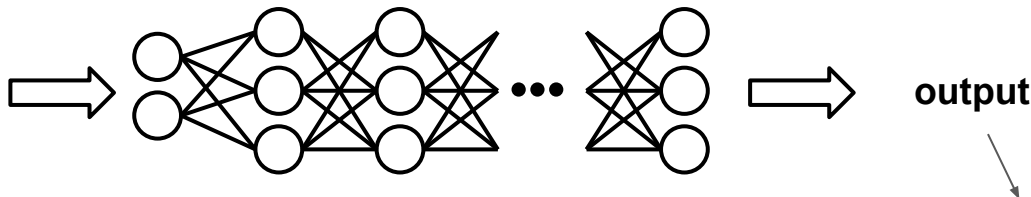
$y = 5$

Deep Learning



Deep Learning

Input



AI 미래에 대해
알려줘



1초 동안 Thought about future of AI

인공지능(AI)의 미래는 매우 다양하고 흥미로운 방향으로 발전할 것으로 예상돼. 몇 가지 중요한 흐름과 전망을 정리해볼게.

1. 더 똑똑해지는 AI 모델:

지금보다 훨씬 더 큰 규모의 데이터와 연산 능력을 바탕으로, AI 모델들은 점점 더 정교하고 사람과 유사한 인지 능력을 갖추게 될 거야. 이미 챗봇이나 번역기, 이미지 생성 AI가 상당히 발전했지만, 미래에는 이들이 더 자연스럽게 인간과 대화하고, 복잡한 문제도 효율적으로 해결하는 모습을 볼 수 있을 거야.

2. 다양한 산업 분야에서의 활용 확대:

의료, 교육, 금융, 교통, 예술, 엔터테인먼트, 농업, 제조업 등 거의 모든 산업 분야에서 AI를 활용할 수 있게 될 거야. 예를 들어, 의료 분야에서는 질병을 진단하고 치료법을 추천하는 등 의사결정을 돕고, 교육 분야에서는 학생마다 맞춤형 학습 자료를 제공하는 '개인 교사' 역할을 할 수도 있어.

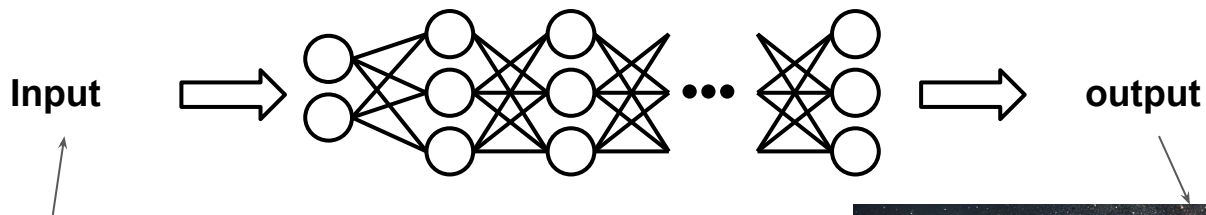
3. 인간과 AI의 협업 증가:

완전히 AI가 모든 것을 대체하기보다, 사람과 AI가 협력하는 형태가 늘어날 것으로 보여. AI가 빠르고 정확하게 데이터를 처리하고 패턴을 찾아주면, 사람은 창의성과 공감 능력을 발휘해 더 나은 결정을 내릴 수 있어. 즉, 인간과 AI가 각자의 장점을 살려 시너지를 내는 방향으로 발전할 거야.

4. 윤리적, 사회적 문제에 대한 관심 증가:

AI가 우리의 삶에 깊숙이 들어올수록, 데이터 활용 문제나 프라이버시, 알고리즘 편향(특정 그룹에게 불리한 결과를 내는 현상), 책임 소재 등에 대한 논의도 활발해질 거야. 이를 해결하기 위해 법적 규제와 윤리 가이드라인이 만들어지고, 사람들이 신뢰할 수 있는 AI 기술 개발이 중요해질 거야.

Deep Learning



달에서 우주비행사가 말을
타고 있는 모습을 그려줘



AI를 배우기 위한 필수 지식

1. 프로그래밍 언어(Python)
2. 선형대수학
3. 확률과 통계

AI를 배우기 위한 필수 지식

1. 프로그래밍 언어(Python)
2. 선형대수학
3. 확률과 통계

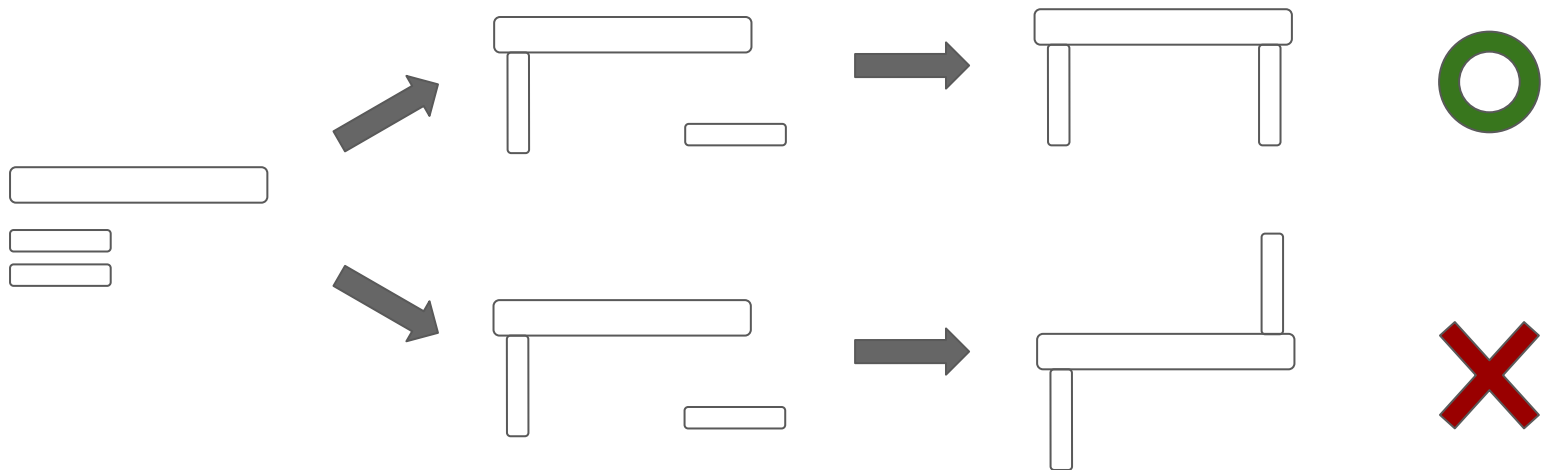
프로그래밍 언어

프로그래밍이란 ?

어떤 문제를 해결하기 위하여 그 처리 방법과 순서를 기술하여 컴퓨터에 주어지는 일련의 명령문 집합체

프로그래밍이란 ?

어떤 문제를 해결하기 위하여 그 처리 방법과 순서를 기술하여 컴퓨터에 주어지는 일련의 명령문
집합체



프로그래밍 언어

유클리드 알고리즘으로 최대공약수 구하기

1. **목표:** 두 숫자의 최대공약수(GCD)를 구한다.
2. **방법:**
 - 두 숫자 **a**와 **b**를 가지고 시작한다.
 - **a**가 **b**보다 크다고 가정한다.
 - **a%b**(나머지)를 계산한다.
 - 나머지가 0이 아닐 경우, **a**를 **b**로, **b**를 나머지로 바꿔서 과정을 반복한다.
 - 나머지가 0이 되면, **b**가 GCD이다.

예제: 56과 15의 최대공약수 구하기

1. 초기 값: **a=56, b=15**
2. 계산 과정:
 - $56 \div 15 = 3$ (몫), 나머지 $56 \% 15 = 11$
 - 새로운 값: **a = 15, b = 11**
 - $15 \div 11 = 1$ (몫), 나머지 $15 \% 11 = 4$
 - 새로운 값: **a = 11, b = 4**
 - $11 \div 4 = 2$ (몫), 나머지 $11 \% 4 = 3$
 - 새로운 값: **a = 4, b = 3**
 - $4 \div 3 = 1$ (몫), 나머지 $4 \% 3 = 1$
 - 새로운 값: **a = 3, b = 1**
 - $3 \div 1 = 3$ (몫), 나머지 $3 \% 1 = 0$
 - 나머지가 0이므로 GCD는 **b = 1**

Why Python?



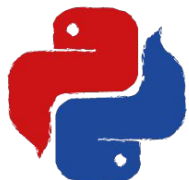
1. 풍부한 라이브러리와 프레임워크가 갖춰진 훌륭한 생태계
2. 쉬운 문법과 가독성

 PyTorch

 TensorFlow

Caffe


OpenCV


KoNLPy

 **SciPy**

 Jupyter

 pandas

 scikit
learn

 NumPy

matplot*lib*

 python™

쉬운 문법과 가독성



숫자의 크기 비교

```
if age > 18:  
    print("You are an adult.")
```

복잡한 기호가 적음

```
numbers = [1, 2, 3]  
numbers.append(4)  
print(numbers)
```

보기 좋은 구조

```
age = 20  
  
if age >= 18:  
    print("You can vote.")  
else:  
    print("You are too young to vote.")
```

불필요한 설명이 필요없음

```
total = sum([10, 20, 30])  
print(total)
```

2강. Python 기초: 기본 문법과 환경 설정

3강. Python 기초: 조건문과 반복문

4강. Python 기초: 함수와 클래스

5강. 이미지 처리 기법: OpenCV, Pillow, matplotlib

6강. AI 프로젝트: 이미지 분류, 객체 탐지 등

수업을 듣고 난 이후 여러분의 모습은?

Python을 활용하여 간단한 문제를 풀 수 있다.

- 최대공약수를 구하는 프로그램
- 특정 질문에 답할 수 있는 간단한 챗봇 만들기
- 숫자 맞추기 게임

원하는 AI 모델을 활용하여 나만의 프로젝트를 만들어볼 수 있다.

- 나만의 책읽어주는 AI 만들기
- 초상권 침해 방지를 위하여 이미지에서 사람을 찾아 자동으로 모자이크하기
- 이미지에서 인물 누끼 따기

- 오늘 이 자리에서 **AI**를 학습하고 있는 여러분들은 정말 대단한 열정을 가졌다는 것을 스스로 자부심을 느껴도 됩니다.
- 어렵더라도 괜찮습니다. 모든 것을 이해하기보다 익숙해지세요. 컴퓨터 공학을 전공하는 대학생들도 어려워합니다.
- 수업을 들으며 내가 정말 재밌고 생각만 해도 가슴이 뛰는 분야를 찾아보세요.

Google Colab(Colaboratory)

- Google에서 제공하는 무료 클라우드 기반 Jupyter 노트북 환경

시작 방법

1. Google 계정으로 로그인
2. Google Colab 열기(새 노트북 생성 또는 기존 노트북 열기)

Colab의 장단점

장점:

- 무료 제공
- 설치 필요 없음
- 하드웨어 가속기 사용 가능(GPU, TPU).

단점:

- 인터넷 연결 필요.
- 특정 런타임 제한(최대 12시간).
- 하드웨어 사양이 제한적(유료 Colab Pro에서 확장 가능).

✓ 클라우드 기반:

- Google Colab은 설치가 필요 없으며, 웹 브라우저에서 바로 실행됩니다.
- Google 드라이브와 통합되어 프로젝트 파일을 저장하고 관리할 수 있습니다.

✓ 무료 GPU & TPU 지원:

- Colab은 무료로 GPU와 TPU(하드웨어 가속기)를 지원하여 딥러닝과 데이터 과학 프로젝트를 실행할 수 있습니다.

✓ Python 지원:

- 파이썬 프로그래밍을 기본으로 지원하며, 다양한 데이터 과학 라이브러리 (예: Pandas, NumPy, TensorFlow, PyTorch)를 사용할 수 있습니다.

✓ Jupyter 노트북과 호환:

- Jupyter 노트북과 완전히 호환되어 기존 `.ipynb` 파일을 열고 편집할 수 있습니다.

✓ 실시간 협업:

- Google 문서처럼 여러 사용자가 동시에 노트북을 편집하고 주석을 달 수 있습니다.

✓ *Python-AI-Tutorial* [Link](#)에서 Colab 클릭

Python-AI-Tutorial

1강

- [LectureNote](#)

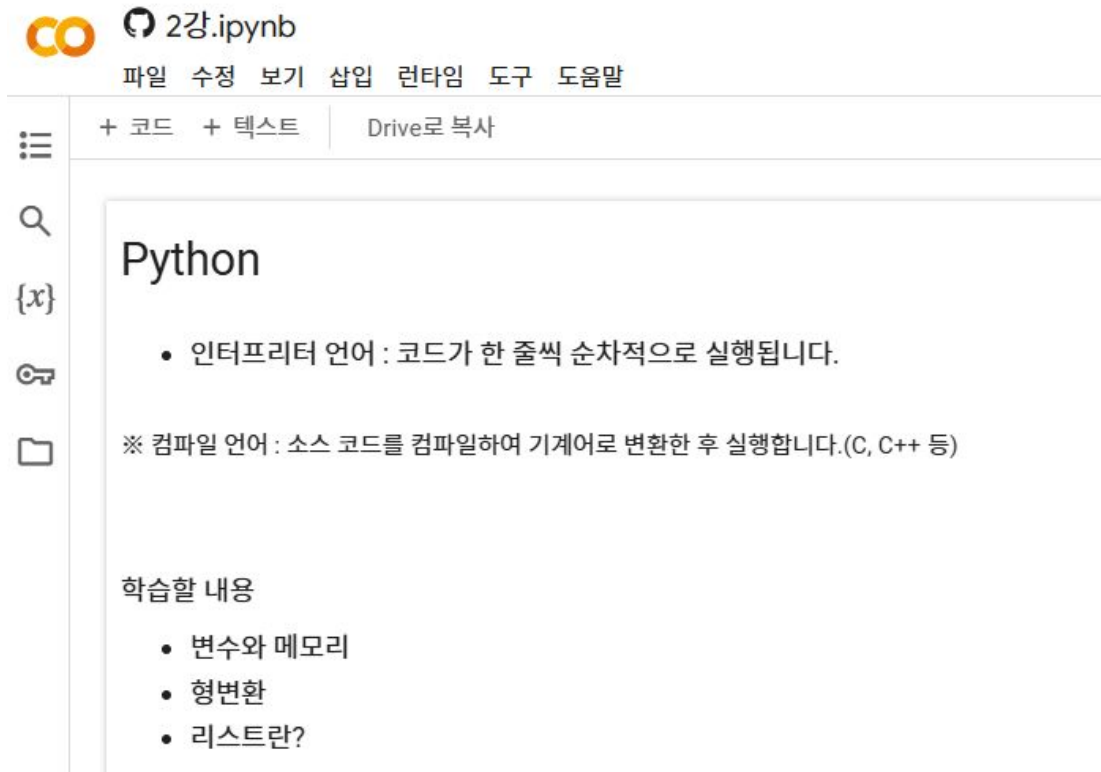
2강

- [LectureNote](#)
- [Colab](#)

3강

- [Colab](#)

✓ Colab 파일 확인



CO 2강.ipynb

파일 수정 보기 삽입 런타임 도구 도움말

+ 코드 + 텍스트 | Drive로 복사

Python

- 인터프리터 언어 : 코드가 한 줄씩 순차적으로 실행됩니다.

※ 컴파일 언어 : 소스 코드를 컴파일하여 기계어로 변환한 후 실행합니다.(C, C++ 등)

학습할 내용

- 변수와 메모리
- 형변환
- 리스트란?

✓ 파일 > Drive에 사본 저장 클릭



Q & A