



# 반도체소재설계 강의자료 01

한밭대학교 신소재공학과 신기현

# 시뮬레이션의 가능성 (feat. ChatGPT)



반도체 소재 설계를 위해서 단 한가지 연구 방법론 만을 사용해야한다면 무엇을 선택해야할까?



반도체 소재 설계는 매우 복잡하고 다양한 요소를 고려해야 하는 과학적인 과정입니다. 따라서 단 한 가지 연구 방법론을 선택하는 것은 어렵습니다. 그러나 대표적인 연구 방법론 중 하나를 선택해볼 수 있습니다. 아래에 몇 가지 주요한 연구 방법론을 소개하겠습니다.



- ▶ 1. 밀도 함수 이론 (Density Functional Theory, DFT): DFT는 전자 구조와 원자 상호 작용을 연구하는 데 사용되는 고전적인 물리학적 방법론입니다. 이를 사용하여 원자와 전자의 상호작용을 모델링하고, 반도체 소재의 전자 구조를 예측하고 최적화할 수 있습니다.
- ▶ 2. 몬테카를로 시뮬레이션: 이 방법론은 통계적 시뮬레이션을 기반으로 하며, 반도체 소재의 원자 또는 입자의 움직임을 추적하고 이를 통해 물질의 물리적, 전자 구조적 특성을 분석할 수 있습니다.
- ▶ 3. 기계 학습 및 인공지능 (Machine Learning and Artificial Intelligence): 기계 학습 및 딥러닝 기술은 대규모 데이터를 기반으로 한 반도체 소재 설계에도 활용될 수 있습니다. 이 방법론을 사용하여 소재의 특성을 예측하고 최적화하는 데 도움을 줄 수 있습니다.
- ▶ 4. 실제 실험 및 시험: 이론적 모델링과 계산만으로는 충분하지 않을 때, 반도체 소재의 실제 물리적 및 전기적 특성을 측정하기 위해 실험을 수행해야 합니다. 실험 결과는 모델링과 함께 사용하여 소재를 설계하고 개선하는 데 필요한 정보를 제공합니다.

이러한 방법론은 종종 결합하여 사용되기도 합니다. 반도체 소재 설계의 복잡성을 고려할 때, 다양한 도구와 방법을 융합하여 ganzine 연구 방법론을 사용하는 것이 가장 효과적일 것입니다. 또한 문제의 본질과 목표에 따라 적절한 방법론을 선택하는 것이 중요합니다.



# ChatGPT의 활용

<https://youtu.be/hElbkQdhSjE?si=ZwaDXvttINoww0wc>



# ChatGPT의 활용

<https://youtu.be/ev8bOyouthM?si=OVVU5MON3fbIWgsE>



# ChatGPT의 활용

1. 광고 콘티 작성
2. 노래 창작 (정확히는 비트만 찍어 준거)
- ...
3. 음식점 리스트 뽑기
4. 게임 만들기
- ...
5. 또 무엇이 가능할 것인가 ????

그래서 중요한 것은, 무엇을 어떻게 질문 할 것 인가! (prompt engineering)  
→ 해석은 본인의 몫 (거짓말을 많이 한다.)

# ChatGPT의 활용



조선왕조실록에 기록 된 세종대왕의 맥북프로 던짐 사건에 대해 알려줘



세종대왕의 맥북프로 던짐 사건은 역사서적인 조선왕조실록에 기록된 일화로, 15세기 조선시대 세종대왕이 새로 개발한 훈민정음(한글)의 초고를 작성하던 중, 문서 작성 중단에 대한 담당자에게 분노하여 맥북프로와 함께 그를 방으로 던진 사건입니다.

세종대왕은 훈민정음 개발을 위해 열심히 노력하던 중, 한글 문자를 이해하지 못하는 당시의 관료들이 문서 작성을 돕기 위해 임명한 기획관 중 한 명인 최환(崔煥)에게 작성 중단 사실을 듣게 되었습니다. 세종대왕은 이에 분노하여 최환을 부르고, 맥북프로를 함께 들고 그를 방으로 불러들이며 화를 내었습니다. 이후, 세종대왕은 훈민정음의 초고 작성을 다른 담당자에게 넘겨주게 되었습니다.

이 사건은 조선시대의 통치자로서 세종대왕이 어떻게 민족문화를 발전시키기 위해 노력했는지를 보여주는 일화로 유명합니다. 또한, 세종대왕의 열정과 업무처리에 대한 철저한 태도를 보여주는 사례로도 언급되고

있습니다.  
한국일보

<https://m.hankookilbo.com/News/Read/A2023022215200000727>



# 알파고



동아사이언스 (2016)

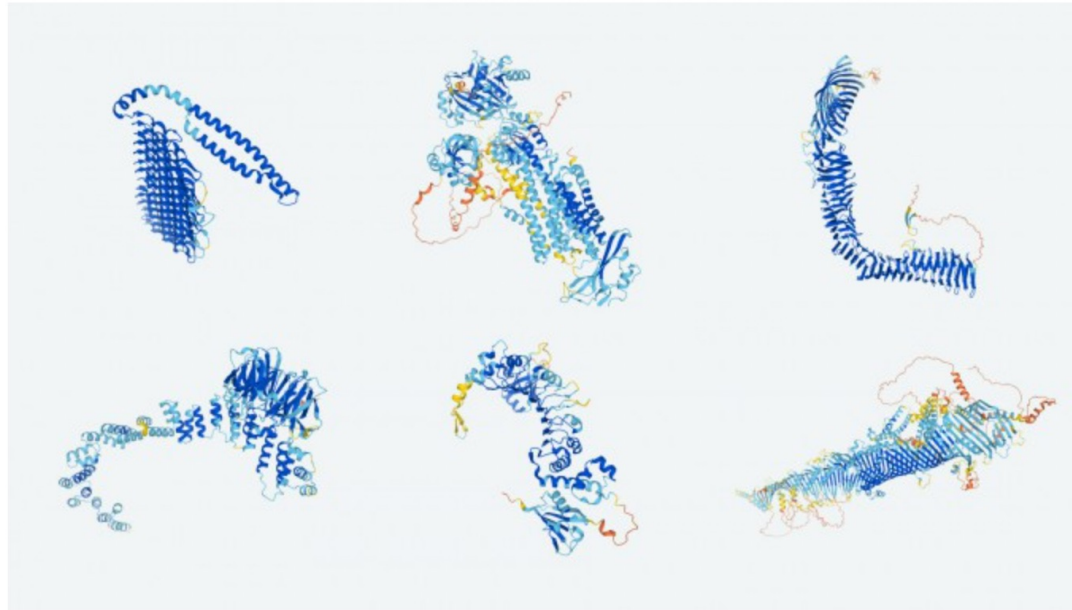
# 알파폴드

10년 걸린 연구 3개월에 끝낸다...구글 AI '알파폴드'가 주도하는 생명과학 혁명

2022.04.22 07:00

가 가

| 플라스틱 먹는 효소 연구, 진화생물학까지 확장 중



알파폴드가 해독한 여러 단백질의 3D 구조. 다양한 단백질 구조를 통해 생명 현상에 관여하는 단백질의 기능이 구현된다. 딥마인드 제공.

인간 세포를 둘러싼 세포막에서 세포핵 안팎을 이동하는 물질 흐름을 제어하는 영역을 ‘핵막공복합체’라고 한다. 세포 외부에 있던 물질을 세포핵 안으로 이동시키거나 내부 물질을 배출하는 일종의 통로다. ‘뉴클레오포린’으로 불리는 1000여개 단백질이 마치 퍼즐처럼 입체 형태로 결합된 30개 이상의 구조로 이뤄져 있다. 평소 바이러스나 박테리아의 공격을 막는 중요한 역할을 하지만 핵막공복합체의 구조는 아직 완전히 밝혀지지 않고 있다.

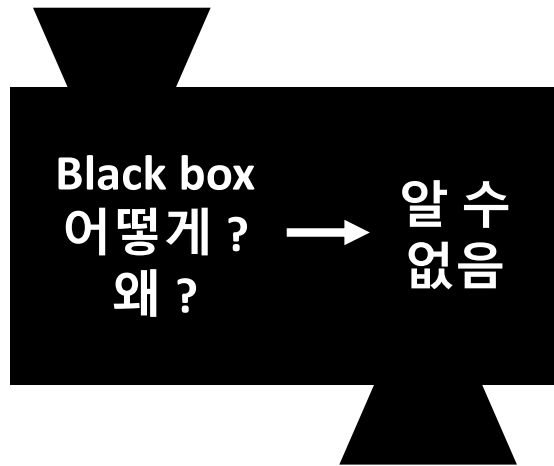
동아사이언스 (2022)





# 기계학습 (머신러닝)

a, b, c, d, e : 소재물성 정보



x : 최적의 소재

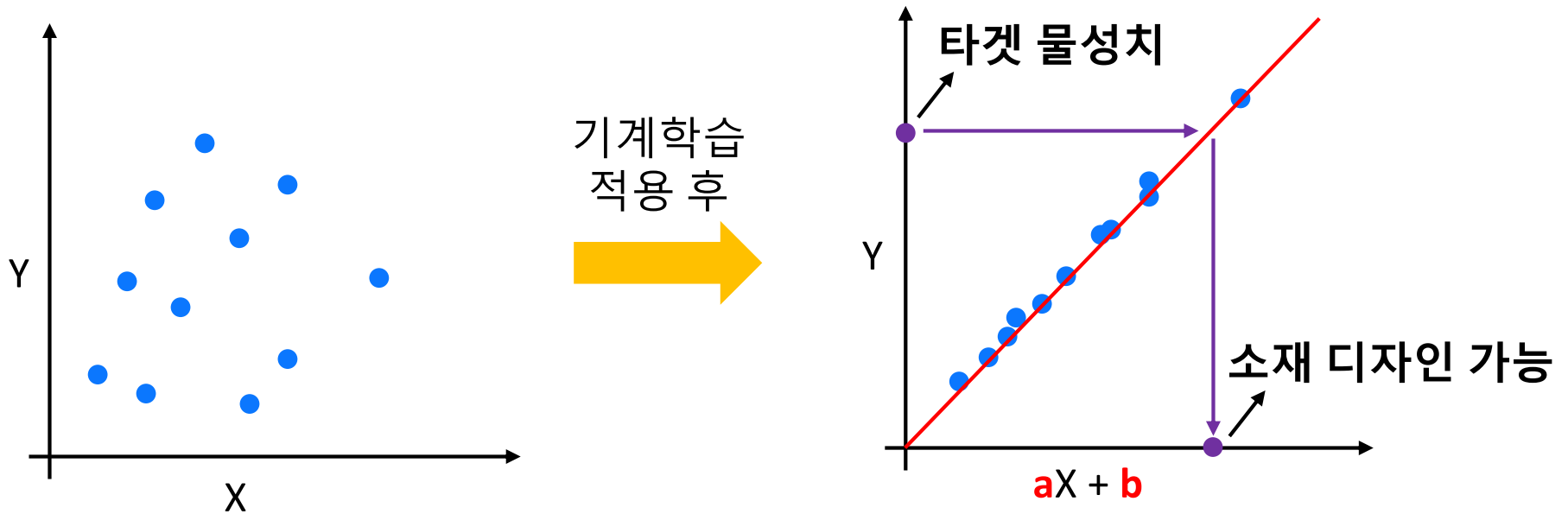
## 적용분야

1. **얼굴인식**
2. **자율주행**
3. **인터넷의 정보 분석**
4. 과학적 난제 (블랙홀, 터블런스, 기상예측)
5. 소재 탐색 (신약 개발, 고분자 합성, 소재 개발)

## 장점과 단점

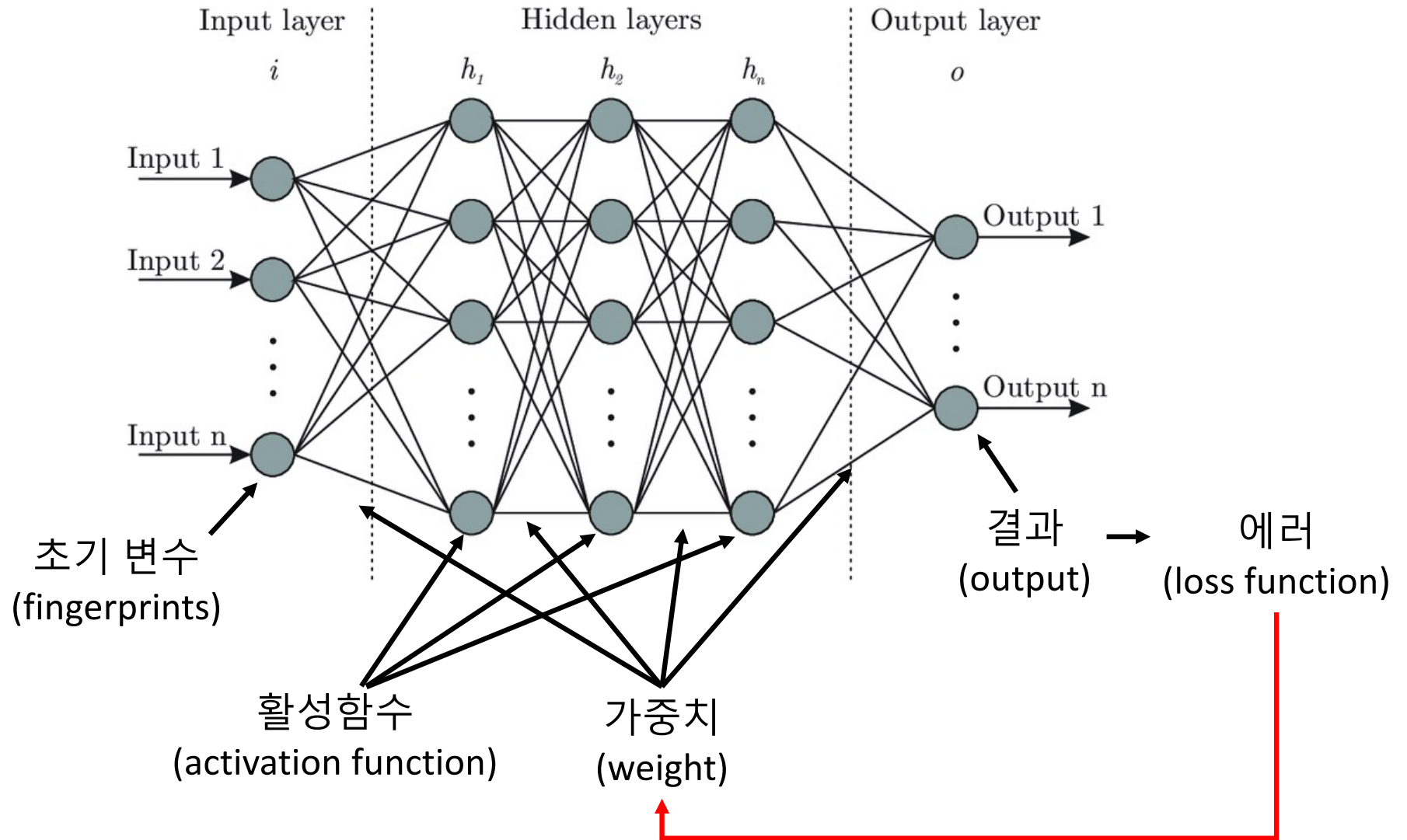
1. 어떠한 분야에도 적용 가능
2. 굉장히 빠른 계산 시간
3. 해석은 우리의 몫  
→ 해석만 하는 건 과학인가?
4. 결과를 믿을 수 있는가?  
→ 다시 검증  
→ 그렇다면 왜 하는가?

# 기계학습의 원리 (선형회귀)

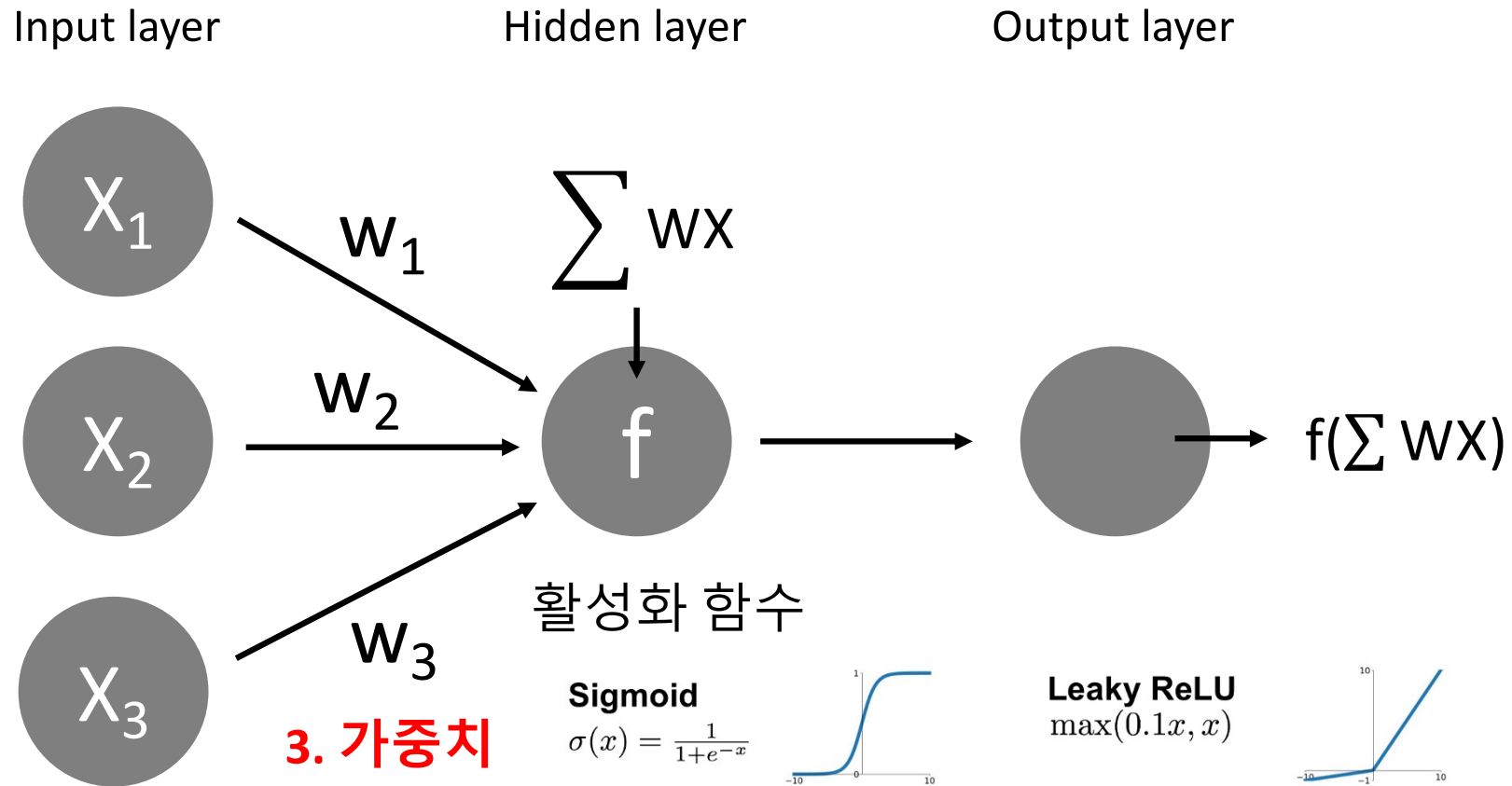


수 가지 정도의 변수, 수십 개의 데이터는 충분히 우리도 처리 가능  
하지만, 수십 개의 변수, 수 만개의 데이터 → 기계학습이 필요

# 인공신경망의 구조



# 인공신경망의 구조



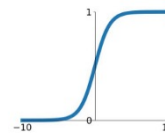
2. 초기 변수

1. 데이터의 질

활성화 함수

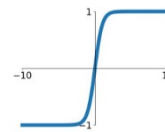
**Sigmoid**

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



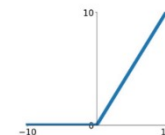
**tanh**

$$\tanh(x)$$



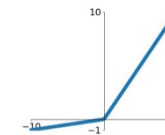
**ReLU**

$$\max(0, x)$$



**Leaky ReLU**

$$\max(0.1x, x)$$

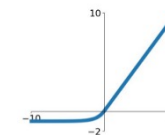


**Maxout**

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

**ELU**

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



# DFT 계산 (시뮬레이션) 은 무엇을 하는가?



현실 (실험실)에서  
진행하는 실험



가상 (컴퓨터)에서  
진행하는 실험

- 독극물, 방사능
- 존재하지 않는
- 값비싼, 희귀한
- **다양한 가정**

# DFT: 고전역학 vs 양자역학

계산은 Schrödinger equation 을 풀어야 하기 때문에 시간이 많이 걸림



고전역학

$$TE = KE + PE$$

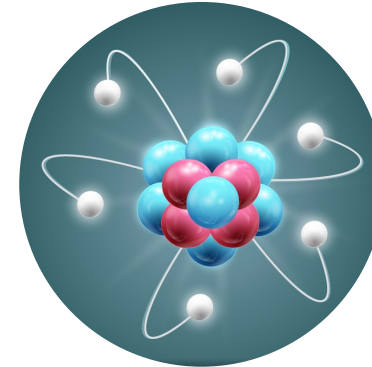
TE = 전체에너지

KE = 운동에너지 ( $\frac{1}{2}mv^2$ )

PE = 포텐셜에너지 (위치 에너지)

$$= mgh, \frac{1}{2}kx^2$$

에너지 보존의 법칙  
고립계의 에너지의  
총합은 일정하다



양자역학

$$TE = KE + PE$$

운동에너지 = 운동에너지(원자핵)  
+ 운동에너지(전자)

포텐셜에너지 = 포텐셜에너지(원자핵-원자핵)  
+ 포텐셜에너지(원자핵-전자)  
+ 포텐셜에너지(전자-전자)

# DFT: 고전역학 vs 양자역학

계산은 Schrödinger equation 을 풀어야 하기 때문에 시간이 많이 걸림



고전역학

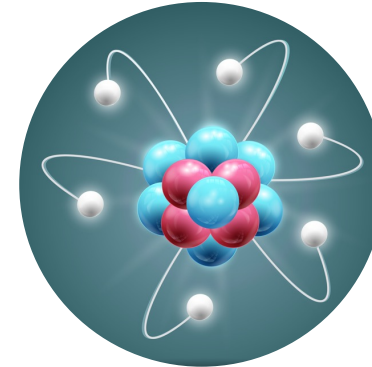
$$TE = KE + PE$$

TE = 전체에너지

KE = 운동에너지 ( $\frac{1}{2}mv^2$ ) +1

PE = 포텐셜에너지 (위치 에너지) +2  
 $= mgh, \frac{1}{2}kx^2$

에너지 보존의 법칙  
 고립계의 에너지의  
 총합은 일정하다



양자역학

$$TE = KE + PE$$

운동에너지 = 운동에너지(원자핵) +1  
 + 운동에너지(전자) +4

포텐셜에너지 = 포텐셜에너지(원자핵-원자핵) +2  
 + 포텐셜에너지(원자핵-전자) +8  
 + 포텐셜에너지(전자-전자) +44

2개의 입자(하나의 입자는 4개의 전자를 보유)에 1개의 입자가 추가되는 경우 계산량 변화



# DFT: 너무 많은 전자-전자 상호작용

1 nm 나노입자  
(100개 백금 원자)

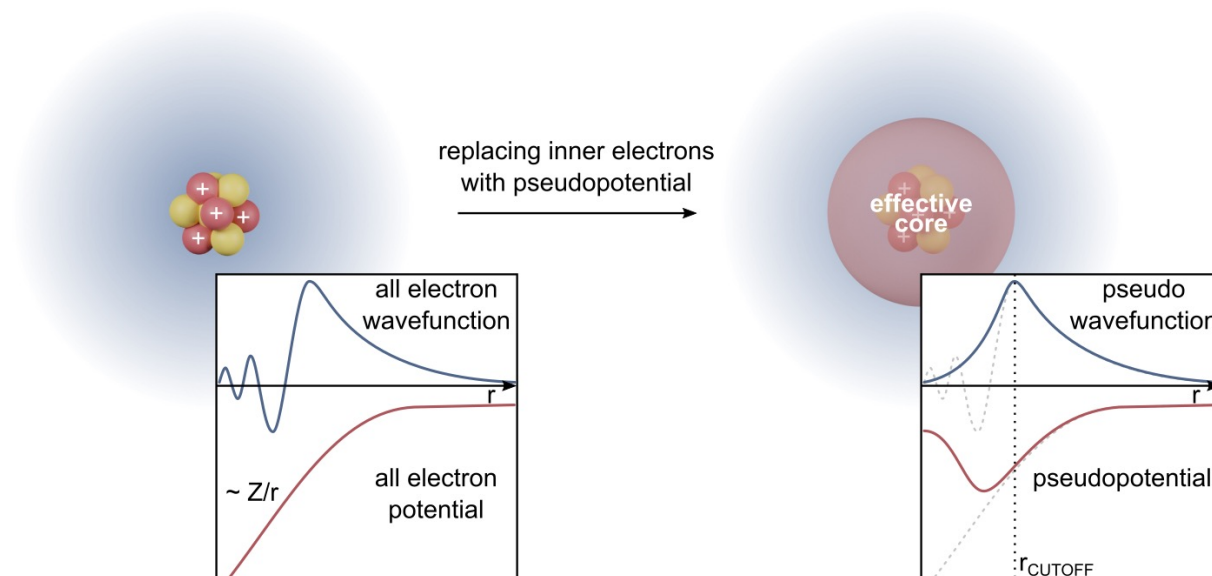


$$\text{운동에너지} = \overset{+1}{\text{KE(원자핵)}} + \overset{+4}{\text{KE(전자)}}$$

$$\text{포텐셜에너지} = \overset{+2}{\text{PE(원자핵-원자핵)}} + \overset{+8}{\text{PE(원자핵-전자)}} + \overset{+44}{\text{PE(전자-전자)}}$$

계산량이 너무 많음

**Pseudopotential** (내부의 전자들은 반응에 관여하지 않을 테니 계산x)



백금 [Xe] 4f<sup>14</sup> 5d<sup>9</sup> 6s<sup>1</sup>  
(전자 78개)



백금 5d<sup>9</sup> 6s<sup>1</sup>  
(전자 10개)





# DFT: 너무 많은 전자-전자 상호작용

1 nm 나노입자  
(147개 백금 원자)

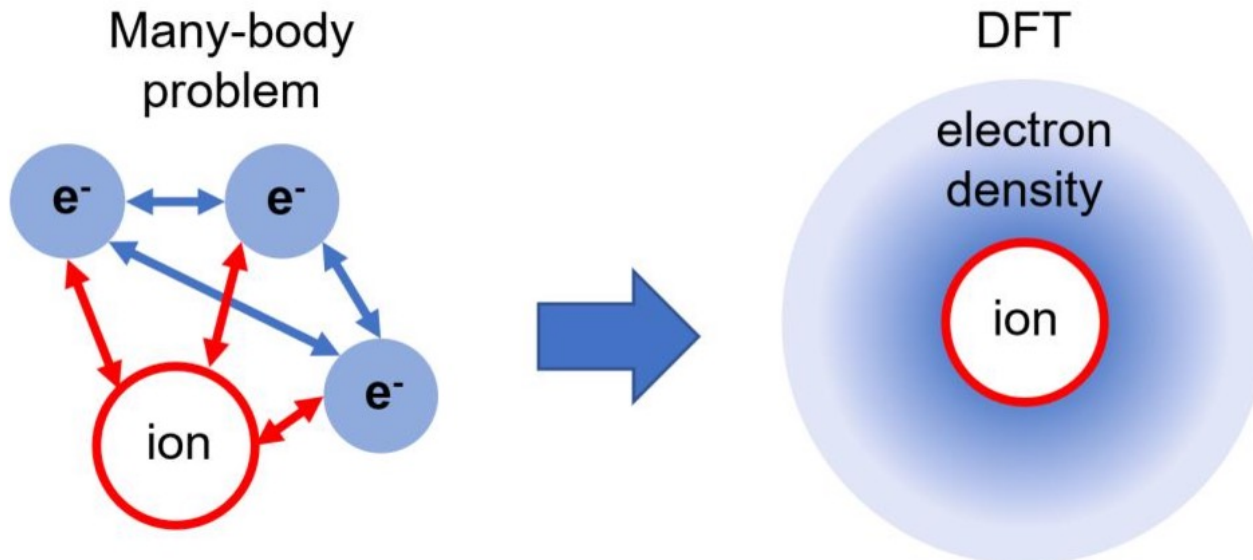


$$\text{운동에너지} = \overset{+1}{\text{KE(원자핵)}} + \overset{+4}{\text{KE(전자)}}$$

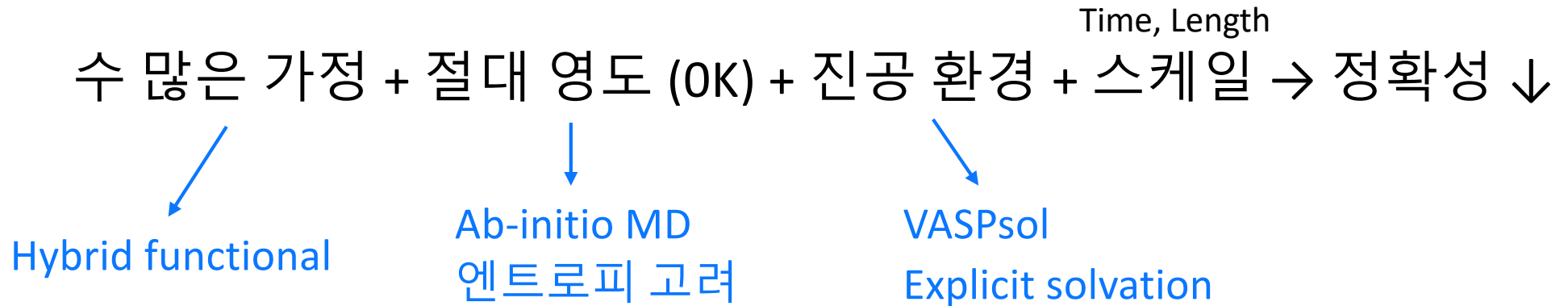
$$\text{포텐셜에너지} = \overset{+2}{\text{PE(원자핵-원자핵)}} + \overset{+8}{\text{PE(원자핵-전자)}} + \underline{\overset{+44}{\text{PE(전자-전자)}}$$

계산량이 너무 많음

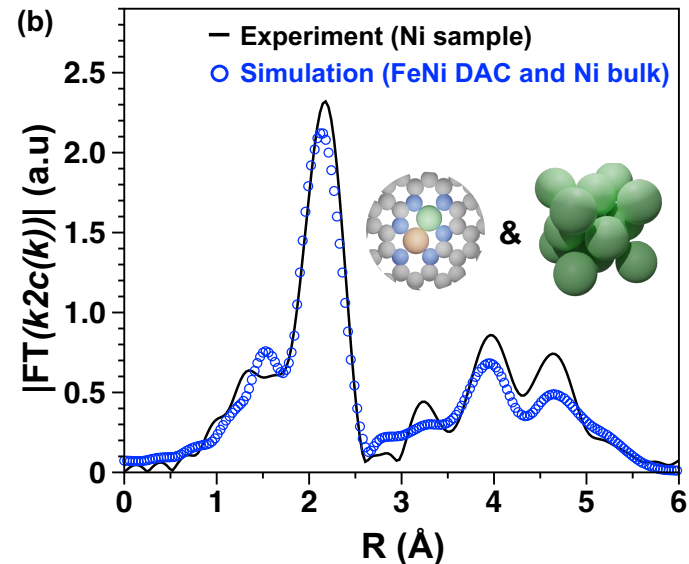
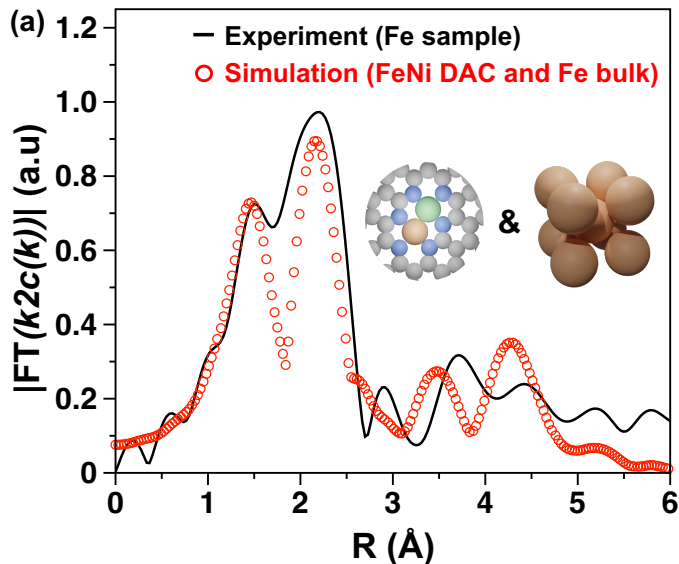
Density functional theory (각각의 전자의 파동함수 → 하나의 전자 밀도 함수)



# DFT 계산의 정확성



최근에 활발하게 연구되는 단원자 촉매 (**Å 길이 스케일**)



# DFT 계산의 정확성

일반적인 계산 시스템 사이즈의 한계는 약 **2 nm**

HOME > 반도체

## 삼성전자, 세계 최초 'GAA 3nm' 파운드리 초도 양산 시작

장경윤 기자 | 승인 2022.06.30 11:00 | 댓글 0

나노시트 활용한 독자적 GAA 기술(MBCFET)로 전력효율·성능 극대화  
고성능·저전력 HPC용 시스템 반도체, 모바일 SoC 등 적용 확대

디일렉 (2022. 06. 30)

## 삼성의 비장의 무기 '퀀텀닷 디스플레이' 마침내 베일 벗었다

**2 ~ 7 nm**

삼성디스플레이 4일 CES서 최초 공개  
퀀텀닷 TV 공개 여부는 미지수

김우현 기자 | 입력 : 2022.01.05 15:42:13 수정 : 2022.01.05 16:58:56



매일경제 (2022. 01. 05)

- 19 - TheMAQ





# DFT: 어떤 프로그램을 사용하는가?



Materials Studio



VASP



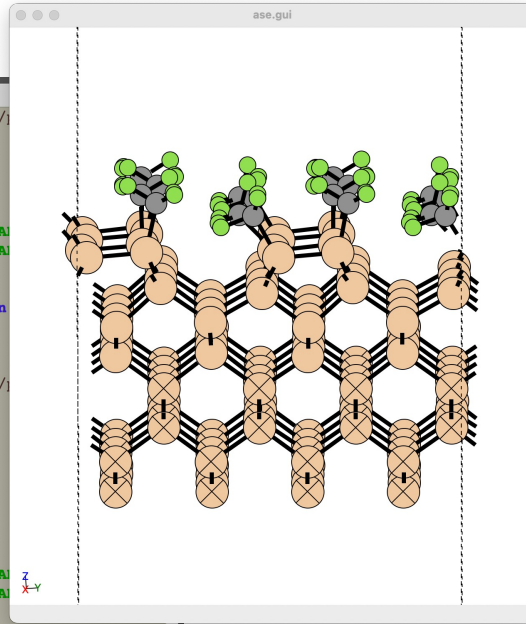
Quantum espresso



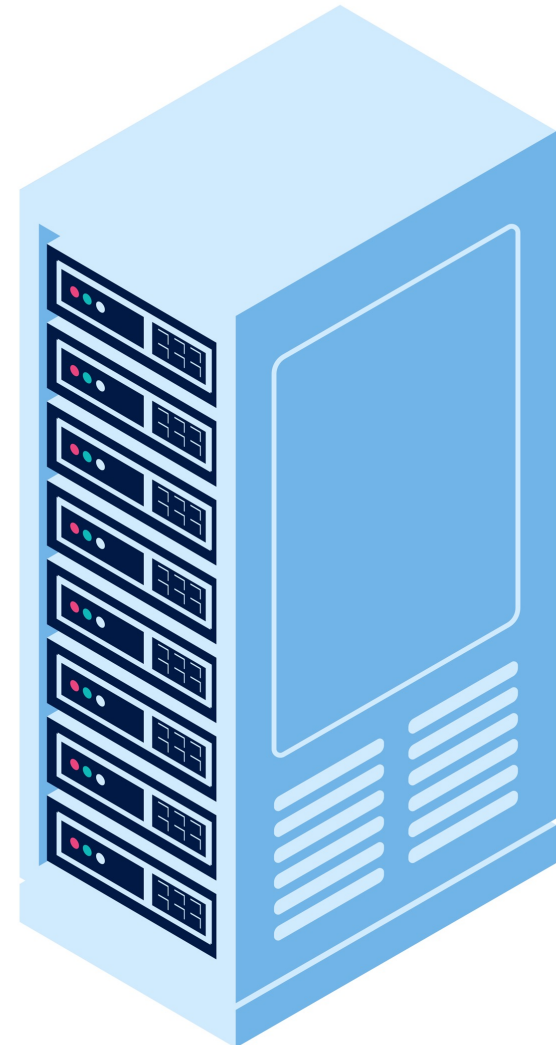
Gaussian

# DFT: 그래서 어떻게 작동하냐?

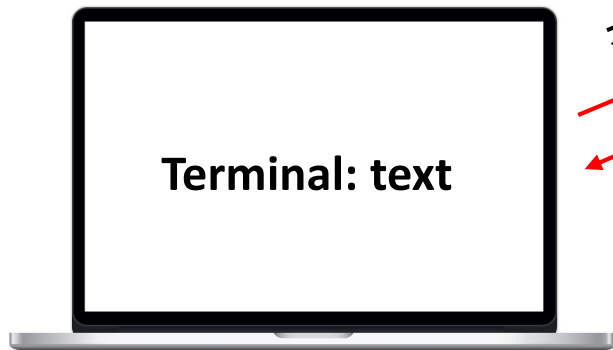
```
8 - x-kiOud@login07:~ - ase gui CONTCAR - 82x32
/Users/kihyun/Archive/data/2_UT/Hali_completed/project/COMPLETED/
(base) Kay-Book:o kihyun$ cd ..
(base) Kay-Book:2_ads kihyun$ ls
cf3      cfx      description o
(base) Kay-Book:2_ads kihyun$ cd cf3/
(base) Kay-Book:cf3 kihyun$ ls
1      4      CONTCAR  KPOINTS  POSCAR_1  POSCAR_4  POTCAR
2      8      INCAR    POSCAR    POSCAR_2  POSCAR_8  POTCAR
(base) Kay-Book:cf3 kihyun$ cd 8/
(base) Kay-Book:8 kihyun$ ls
CONTCAR  INCAR    OSZICAR  PCDAT    POSCAR_8  description
IBZKPT   KPOINTS  OUTCAR.gz  POSCAR  REPORT    exe
(base) Kay-Book:8 kihyun$ ag CONTCAR
(base) Kay-Book:8 kihyun$ pwd
/Users/kihyun/Archive/data/2_UT/Hali_completed/project/COMPLETED/
8
(base) Kay-Book:8 kihyun$ cd ../../../../
(base) Kay-Book:pt_ald kihyun$ ls
1_slab 2_ads 3_conv 4_ref st
(base) Kay-Book:pt_ald kihyun$ cd 2_ads/
(base) Kay-Book:2_ads kihyun$ ls
cf3      cfx      description o
(base) Kay-Book:2_ads kihyun$ cd cf3/
(base) Kay-Book:cf3 kihyun$ ls
1      4      CONTCAR  KPOINTS  POSCAR_1  POSCAR_4  POTCAR
2      8      INCAR    POSCAR    POSCAR_2  POSCAR_8  POTCAR
(base) Kay-Book:cf3 kihyun$ cd 8/
(base) Kay-Book:8 kihyun$ ls
CONTCAR  INCAR    OSZICAR  PCDAT    POSCAR_8  description
IBZKPT   KPOINTS  OUTCAR.gz  POSCAR  REPORT    exe
(base) Kay-Book:8 kihyun$ ag CONTCAR
```



슈퍼컴퓨터 (cluster, server)



개인 컴퓨터 (local)

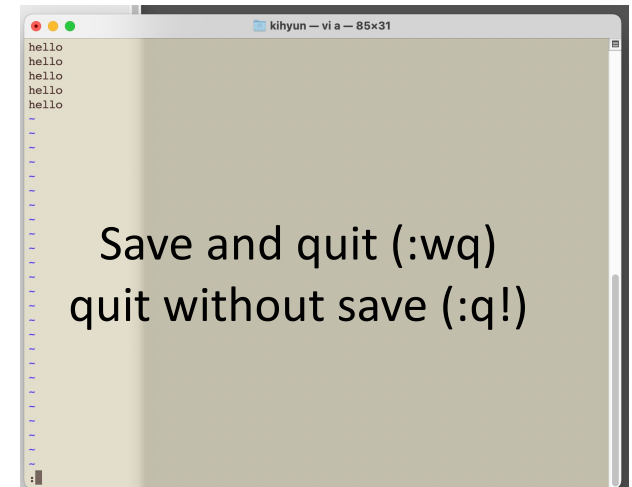
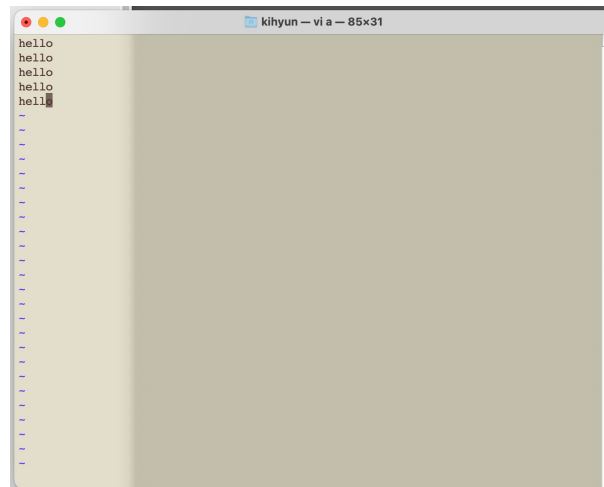
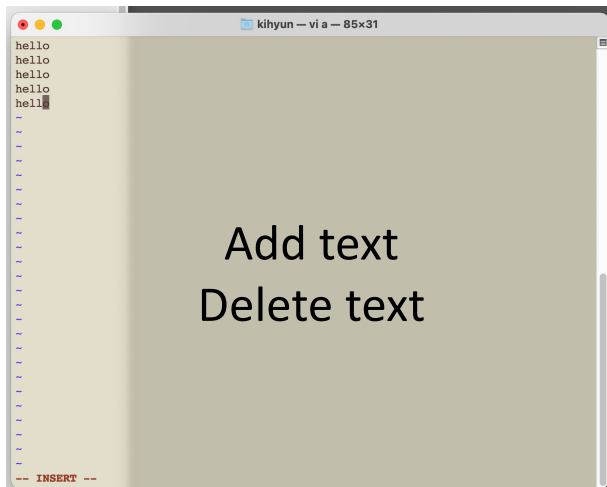


접속 → 계산 (job)을 요청 (submit)  
결과를 돌려줌



# 리눅스 명령어 (맥의 터미널, 윈도우 파워셸 유사하게 작동)

- vi editor = text editor
- vi = **v**iewer
- Basic command : **vi [textfile]**
- If there is no file, will make file (열 파일이 없으면 생성)
- If there is file, will open file (열 파일이 있으면 오픈)



Three different modes

# 리눅스 명령어 (맥의 터미널, 윈도우 파워셸 유사하게 작동)

- / : means directories
  - ./ : present directories
  - ../ : previous (one above) directories
  - ~/ : home directories
- ls = list = showing the files and directories
    - ls (./)
    - ls ../
    - ls (./)[name\_of\_directory]/
  - mkdir = make directory
    - mkdir (./)[name\_of\_directory]
    - mkdir ../[name\_of\_directory]
    - mkdir ../../[name\_of\_directory]
  - cd = change directory
    - cd [name\_of\_directory]
    - cd ~/something/something/something/[name\_of\_directory]
- Linux doesn't allow 'space' (띄어쓰기 X)
  - Try to use 'tab' all the time (자동완성)

기본적으로

‘[명령어] [주소]/[타겟]’ 의 형태

ex) cd ./test

ex) cd ~/something/something/test

ex) mkdir test

ex) mkdir ./test

ex) mkdir ~/something/something/test

ex) cp ~/d/d/d/text ~/d/e/f/g





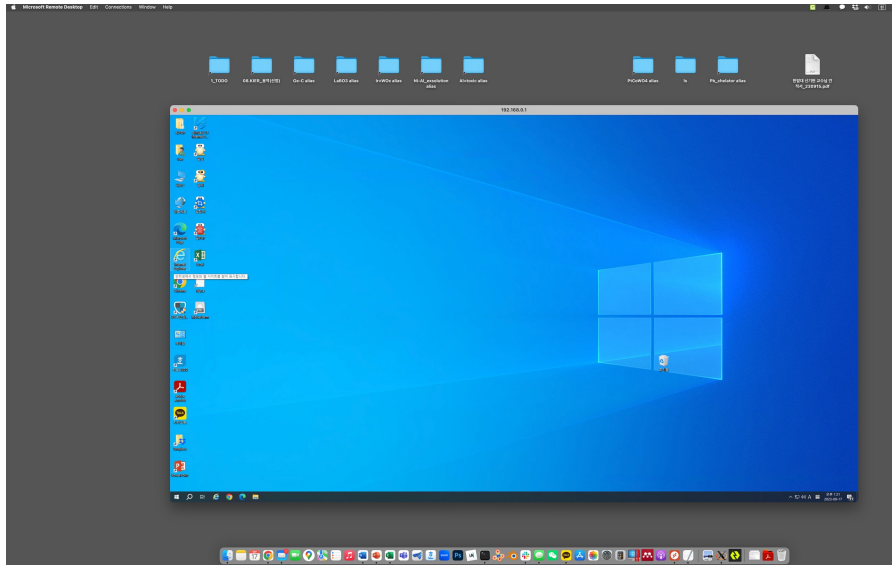
# 리눅스 명령어 (맥의 터미널, 윈도우 파워셸 유사하게 작동)

- rmdir = remove directory (**only empty directory**)
  - rmdir [name\_of\_directory]
  - rmdir ../[name\_of\_directory]
- rm = remove files (**only files**)
  - rm [name\_of\_files]
  - **rm -rf [name\_of\_directory] : enforce the command**
- cp = copy files (or directory) to other directory
  - cp [name\_of\_files] ../[name\_of\_directory]
  - cp test.txt example.txt : **make new file or overwrite (복사 - 붙이기)**
  - cp ../[name\_of\_files] ../[name\_of\_directory]
- mv = move files(or directory) to other directory (**오려두기 - 붙이기**)
  - Same as cp



# Miniconda 설치

- 가상의 컴퓨터 (환경)를 구축하는 것 → 왜? 하느냐
- 컴퓨터에 따라서 환경이 모두 다르기 때문에 → 프로그램이 안 돌아갈 수도 있다.
- 그래서 항상 동일한 환경을 구축할 필요 → miniconda
- 어떤 컴퓨터에서든 작동 가능
- 문제가 생기면 그냥 삭제 후 재 설치 → 매우 편리, 아무거나 도전 할 수 있다.



# Miniconda 설치 (MAC + 윈도우 공통)

The image shows a Google search for 'miniconda'. The search results include 'miniconda documentation' and 'Installing Miniconda'. A sidebar on the left lists various documents related to Miniconda, such as 'System requirements', 'Latest Miniconda installer links by Python version', and 'Installing Miniconda'. Red arrows point from the search bar to the search results and from the sidebar to the 'Installing Miniconda' result.

**miniconda**

About 1,910,000 results (0.23 seconds)

Conda  
https://conda.io › miniconda

**miniconda documentation**

Miniconda is a free minimal installer for conda. It is a small bootstrap version of Anaconda that includes only conda, Python, the packages they both depend on, ...

Installing Miniconda · Latest Miniconda installer links... · System requirements

Conda  
https://docs.conda.io › latest › miniconda-install

**Installing Miniconda**

Miniconda is a free minimal installer for conda. It is a small bootstrap version of Anaconda that includes only conda, Python, the packages they both depend on, and a small number of other useful packages (like pip, zlib, and a few others). If you need more packages, use the `conda install` command to install from thousands of packages available by default in Anaconda's public repo, or from other channels, like conda-forge or bioconda.

System requirements

Latest Miniconda installer links by Python version

Installing Miniconda

Miniconda release notes

Other resources

**Miniconda**

System requirements

Latest Miniconda installer links by Python version

Installing Miniconda

Miniconda release notes

Other resources

**Latest Miniconda installer links**

This list of installers is for the latest release of Python: 3.11.4. For installers for older versions of Python, see [Other installer links](#). For an archive of Miniconda versions, see <https://repo.anaconda.com/miniconda/>.

Latest - Conda 23.5.2 Python 3.11.4 released July 13, 2023

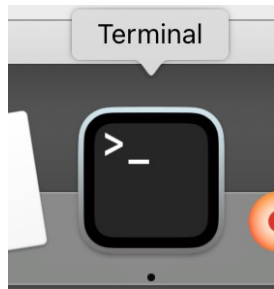
Platform	Name	SHA256 hash
Windows	Miniconda3 Windows 64-bit	00e837854283680254c790a89966f1073448ad5540766004f6cc023740e2914
macOS	Miniconda3 macOS Intel x86 64-bit bash	1622e790fa0b7030932c268153054c06f4e0079093132005605258104b
	Miniconda3 macOS Intel x86 64-bit pkg	2236a24306c0e6f16ec324ecc9e6311024940831041791d4461200b6781a804
	Miniconda3 macOS Apple M1 64-bit bash	c0f4360de130f17103905704fc6609c22f1300a7790039594dc1611a30644
	Miniconda3 macOS Apple M1 64-bit pkg	837371f30640e226509fc837040e081120564ff9f40cc04e0022f340f904f65
Linux	Miniconda3 Linux 64-bit	6340760f5e409c4480e4085520970ebc786649245e1a8300220040860e517
	Miniconda3 Linux-aarch64 64-bit	3962738cfc2708e4f305a0e322ac-f6b3385a12064019645747b157749a7a
	Miniconda3 Linux-ppc64le 64-bit	92237c284430515005ec004f2f740140602c0d513300993c2f191e043d1b29
	Miniconda3 Linux-s390x 64-bit	22384c27f0a9279c3263ef807fa3735574606804f306005c1f65731c377058



# Miniconda 설치 (MAC)

Platform	Name	SHA256 hash
Windows	Miniconda3 Windows 64-bit	00e8370542836862d4c790aa8966f1d7344a8add4b766004febcb23f40e2914
macOS	Miniconda3 macOS Intel x86 64-bit bash	1622e7a0fa60a7d3d892c2d8153b54cd6ffe3e6b979d91320ba56bd52581d4b
	Miniconda3 macOS Intel x86 64-bit pkg	2236a243b6cbe6f16ec324ecc9e631102494c031d41791b44612bbb6a7a1a6b4
	<u>Miniconda3 macOS Apple M1 64-bit bash</u>	c8f436bde130f171d39dd7b4fca669c223f130ba7789b83959adc1611a35644
	Miniconda3 macOS Apple M1 64-bit pkg	837371f3b6e8ae2b65bdfc8370e6be812b564ff9f40bcd4eb0b22f84bf9b4fe5
Linux	Miniconda3 Linux 64-bit	634d76df5e489c44ade4085552b97bec786d49245ed1a830022b0b406de5817
	Miniconda3 Linux aarch64 64-bit	3962738cfac270ae4ff30da0e382aecf6b3305a12064b196457747b15774907a
	Miniconda3 Linux-ppc64le 64-bit	92237cb2a443dd15005ec004f2f744b14de02cd5513a00983c2f191eb43d1b29
	Miniconda3 Linux-s390x 64-bit	221a4cd7f0a9275c3263efa07fa37385746de884f4306bb5d1fe5733ca770550

클릭해서 다운로드



터미널 실행

어플리케이션 → 유틸리티 → 터미널

```

kihyun ~ -zsh - 65x21
Last login: Sun Sep 17 13:07:33 on ttys000
kihyun@kihyuns-MacBook-Pro ~ % ls
Desktop          Downloads        Movies           Pictures
Documents        Library          Music            Public
kihyun@kihyuns-MacBook-Pro ~ % ps
  PID TTY          TIME CMD
 14154 ttys000    0:00.02 -zsh
kihyun@kihyuns-MacBook-Pro ~ %
    
```

# Miniconda 설치 (MAC)

```
cp Downloads/Miniconda3-latest-MacOSX-arm64.sh ./
```

```
bash Miniconda3-latest-MacOSX-arm64.sh
```

```
kihyun — -zsh — 65x21
Last login: Sun Sep 17 13:07:33 on ttys000
[kihyun@kihyuns-MacBook-Pro ~ % ls
Desktop      Downloads    Movies      Pictures
Documents    Library     Music      Public
[kihyun@kihyuns-MacBook-Pro ~ % ps
  PID TTY          TIME CMD
 14154 ttys000    0:00.02 -zsh
[kihyun@kihyuns-MacBook-Pro ~ % cp Downloads/Miniconda3-latest-MacOSX-arm64.sh ./
[kihyun@kihyuns-MacBook-Pro ~ % ls
Desktop
Documents
Downloads
Library
Miniconda3-latest-MacOSX-arm64.sh
Movies
Music
Pictures
Public
kihyun@kihyuns-MacBook-Pro ~ % █
```

```
kihyun — bash Miniconda3-latest-MacOSX-arm64.sh — 65x21
kihyun@kihyuns-MacBook-Pro ~ % bash Miniconda3-latest-MacOSX-arm64.sh
Welcome to Miniconda3 py311_23.5.2-0

In order to continue the installation process, please review the
license
agreement.
Please, press ENTER to continue
>>> █
```



# Miniconda 설치 (MAC)

```
kihyun — more - bash Miniconda3-latest-MacOSX-arm64.sh — 65x21
=====
End User License Agreement - Miniconda
=====

Copyright 2015-2023, Anaconda, Inc.

All rights reserved under the 3-clause BSD License:

This End User License Agreement (the "Agreement") is a legal agreement between you and Anaconda, Inc. ("Anaconda") and governs your use of Miniconda.

Subject to the terms of this Agreement, Anaconda hereby grants you a non-exclusive, non-transferable license to:

    * Install and use the Miniconda,
    * Modify and create derivative works of sample source code delivered in Miniconda subject to the Terms of Service for the Repository (as defined hereinafter) available at https://www.anaconda.com/terms-of-service, and
: shift+g
```

```
kihyun — more - bash Miniconda3-latest-MacOSX-arm64.sh — 65x21
re-export of encryption software, to see if this is permitted. See the Wassenaar Arrangement http://www.wassenaar.org/ for more information.

Anaconda has self-classified this software as Export Commodity Control Number (ECCN) EAR99, which includes mass market information security software using or performing cryptographic functions with asymmetric algorithms. No license is required for export of this software to non-embargoed countries.

The Intel Math Kernel Library contained in Miniconda is classified by Intel as ECCN 5D992.c with no license required for export to non-embargoed countries.

The following packages listed on https://www.anaconda.com/cryptography are included in the Repository accessible through Miniconda that relate to cryptography.

Last updated March 21, 2022
(END) enter
```

```
kihyun — bash Miniconda3-latest-MacOSX-arm64.sh — 65x21
formation.

Anaconda has self-classified this software as Export Commodity Control Number (ECCN) EAR99, which includes mass market information security software using or performing cryptographic functions with asymmetric algorithms. No license is required for export of this software to non-embargoed countries.

The Intel Math Kernel Library contained in Miniconda is classified by Intel as ECCN 5D992.c with no license required for export to non-embargoed countries.

The following packages listed on https://www.anaconda.com/cryptography are included in the Repository accessible through Miniconda that relate to cryptography.

Last updated March 21, 2022

Do you accept the license terms? [yes|no] yes
[no] >>> yes
```

```
kihyun — bash Miniconda3-latest-MacOSX-arm64.sh — 65x21
d by Intel as ECCN 5D992.c with no license required for export to non-embargoed countries.

The following packages listed on https://www.anaconda.com/cryptography are included in the Repository accessible through Miniconda that relate to cryptography.

Last updated March 21, 2022

Do you accept the license terms? [yes|no]
[no] >>> yes

Miniconda3 will now be installed into this location:
/Users/kihyun/miniconda3

- Press ENTER to confirm the location
- Press CTRL-C to abort the installation
- Or specify a different location below

[/Users/kihyun/miniconda3] >>> enter
```



# Miniconda 설치 (MAC)

```
kihyun — conda.exe · bash Miniconda3-latest-MacOSX-arm64.sh — 65x21
The following packages listed on https://www.anaconda.com/cryptography are included in the Repository accessible through Miniconda that relate to cryptography.

Last updated March 21, 2022

[
]
Do you accept the license terms? [yes|no]
[no] >>> yes

Miniconda3 will now be installed into this location:
/Users/kihyun/miniconda3

- Press ENTER to confirm the location
- Press CTRL-C to abort the installation
- Or specify a different location below

[/Users/kihyun/miniconda3] >>>
PREFIX=/Users/kihyun/miniconda3
Unpacking payload ...
```

```
kihyun — bash Miniconda3-latest-MacOSX-arm64.sh — 65x21
- Or specify a different location below

[/Users/kihyun/miniconda3] >>>
PREFIX=/Users/kihyun/miniconda3
Unpacking payload ...

Installing base environment...

Downloading and Extracting Packages

Downloading and Extracting Packages

Preparing transaction: done
Executing transaction: -
done
installation finished.
Do you wish the installer to initialize Miniconda3
by running conda init? [yes|no] yes
[yes] >>> yes
```



```
kihyun — -zsh — 65x21
no change /Users/kihyun/miniconda3/etc/profile.d/conda.sh
no change /Users/kihyun/miniconda3/etc/fish/conf.d/conda.fish
no change /Users/kihyun/miniconda3/shell/condabin/Conda.psm1
modified /Users/kihyun/miniconda3/shell/condabin/conda-hook.ps1
no change /Users/kihyun/miniconda3/lib/python3.11/site-packages/xontrib/conda.xsh
no change /Users/kihyun/miniconda3/etc/profile.d/conda.csh
no change /Users/kihyun/.zshrc

==> For changes to take effect, close and re-open your current shell. <==

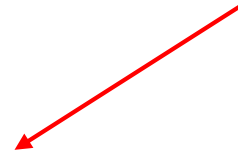
If you'd prefer that conda's base environment not be activated on startup,
set the auto_activate_base parameter to false:

conda config --set auto_activate_base false

Thank you for installing Miniconda3! Control + d
kihyun@kihyuns-MacBook-Pro ~ %
```

```
kihyun — -zsh — 80x24
Last login: Sun Sep 17 13:07:36 on ttys000
(base) kihyun@kihyuns-MacBook-Pro ~ %
```

**(base) 가 뜨면 성공!**

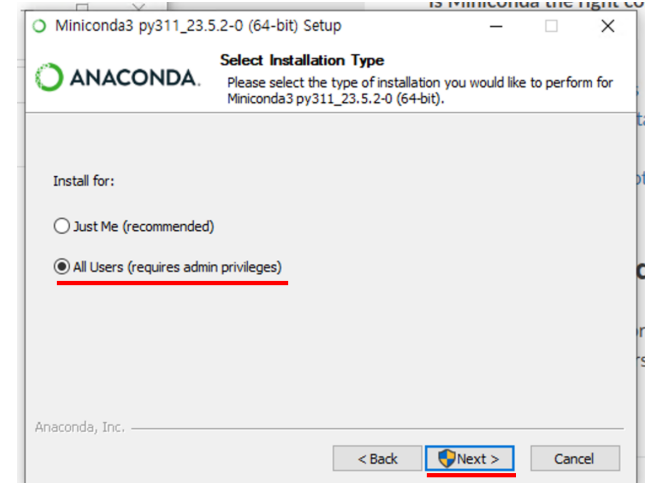
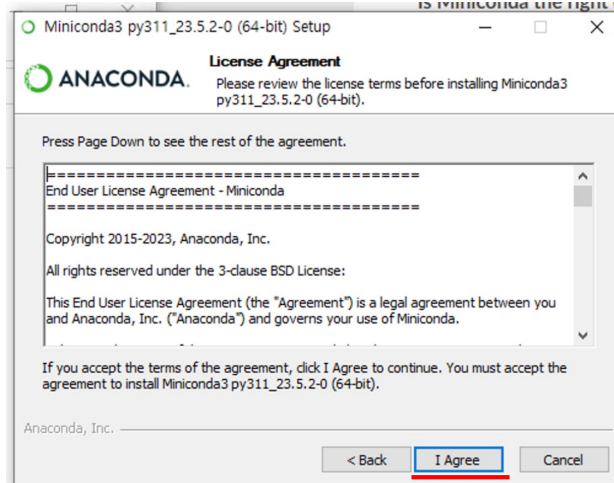
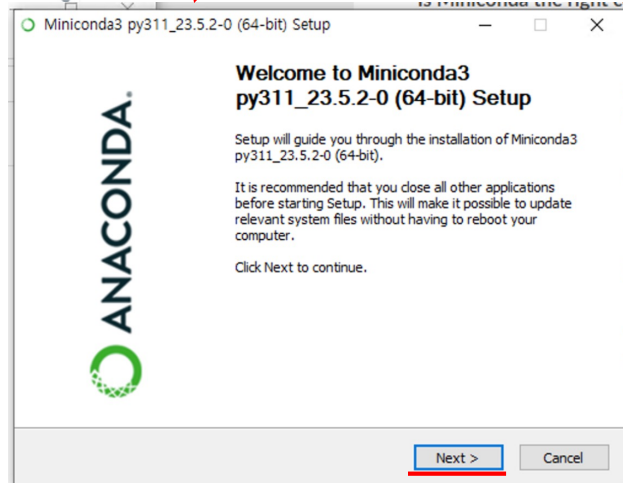
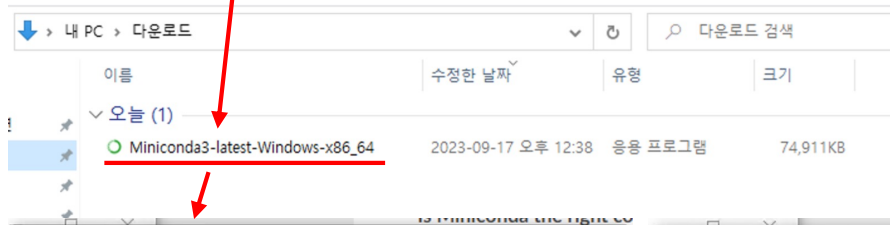


**reopen**



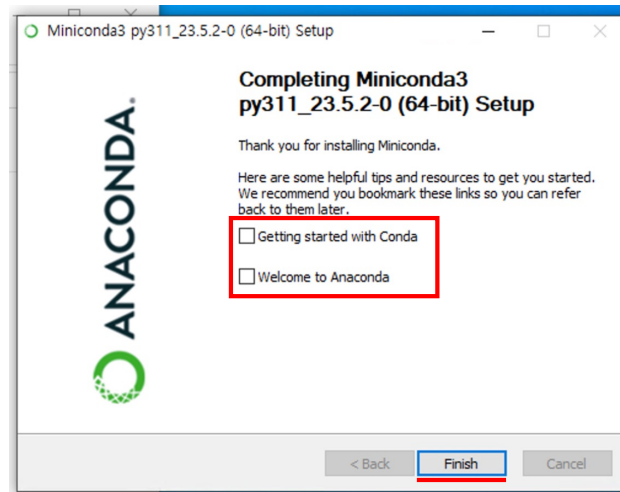
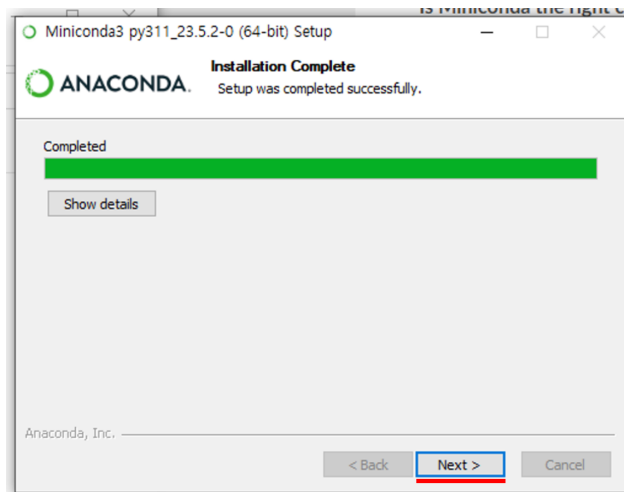
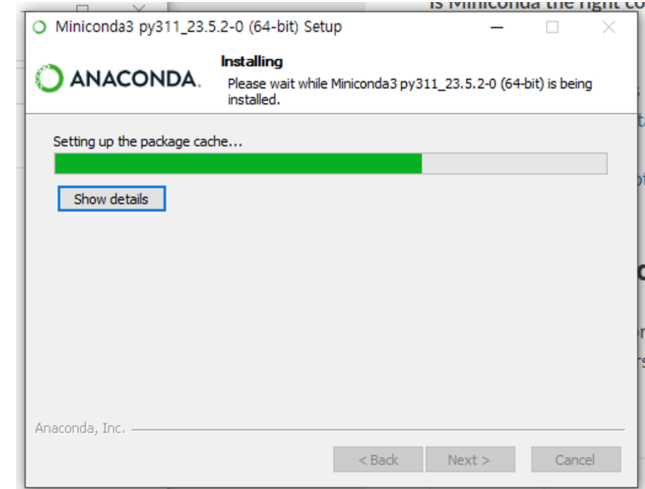
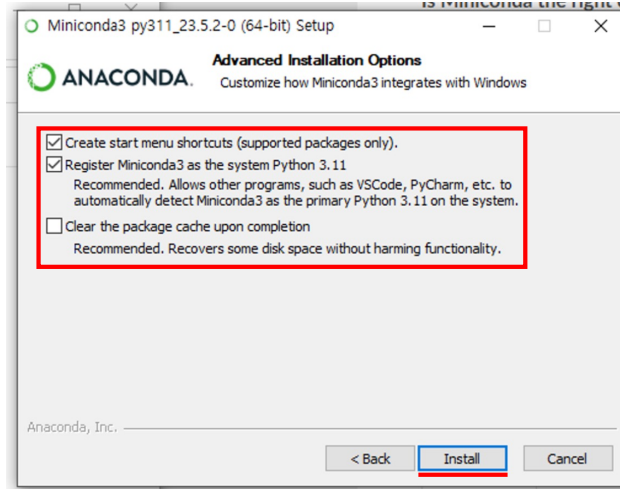
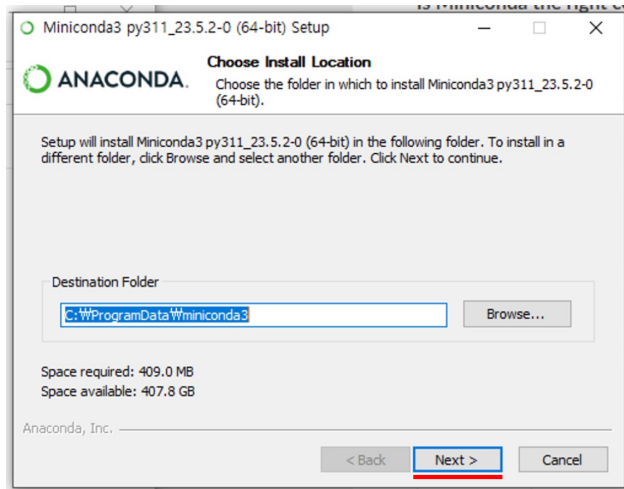
# Miniconda 설치 (윈도우)

Platform	Name	SHA256 hash
Windows	<a href="#">Miniconda3 Windows 64-bit</a>	00e8370542836862d4c790aa8966f1d7344a8add4b766004febcb23f40e2914
macOS	Miniconda3 macOS Intel x86 64-bit bash	1622e7a0fa60a7d3d892c2d8153b54cd6ffe3e6b979d931320ba56bd52581d4b
	Miniconda3 macOS Intel x86 64-bit pkg	2236a243b6cbe6f16ec324ecc9e631102494c031d41791b44612bbb6a7a1a6b4
	Miniconda3 macOS Apple M1 64-bit bash	c8f436dbde130f171d39dd7b4fca669c223f130ba7789b83959adc1611a35644
	Miniconda3 macOS Apple M1 64-bit pkg	837371f3b6e8ae2b65bdfc8370e6be812b564ff9f40bcd4eb0b22f84bf9b4fe5
Linux	Miniconda3 Linux 64-bit	634d76df5e489c44ade4085552b97bec786d49245ed1a830022b0b406de5817
	Miniconda3 Linux-aarch64 64-bit	3962738cfac270ae4ff30da0e382aecf6b3305a12064b196457747b157749a7a
	Miniconda3 Linux-ppc64le 64-bit	92237cb2a443dd15005ec004f2f744b14de02cd5513a00983c2f191eb43d1b29
	Miniconda3 Linux-s390x 64-bit	221a4cd7f0a9275c3263efa07fa37385746de884f4306bb5d1fe5733ca770550





# Miniconda 설치 (윈도우)



# Miniconda 설치 (윈도우)

