

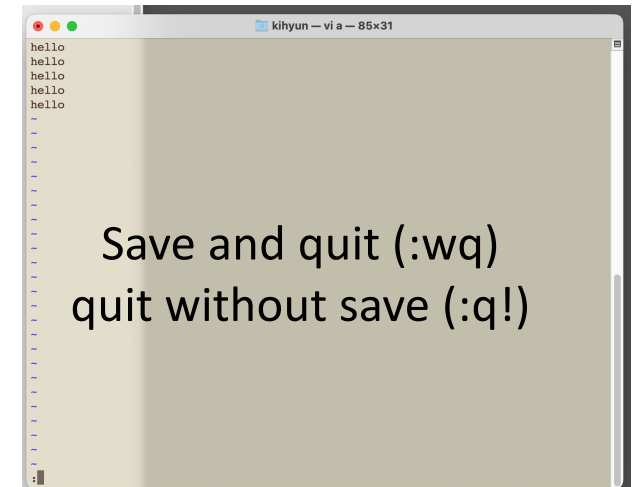
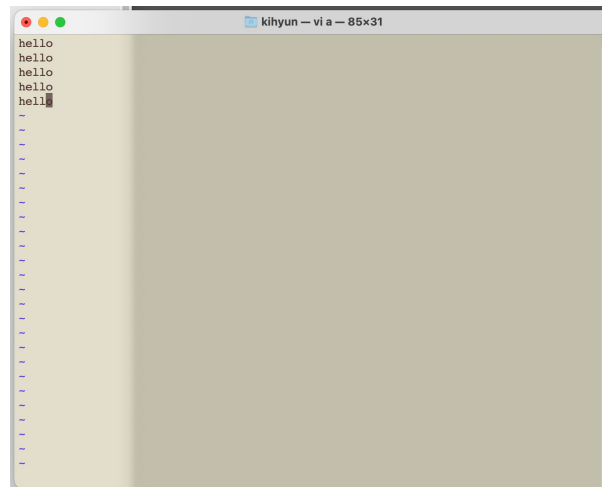
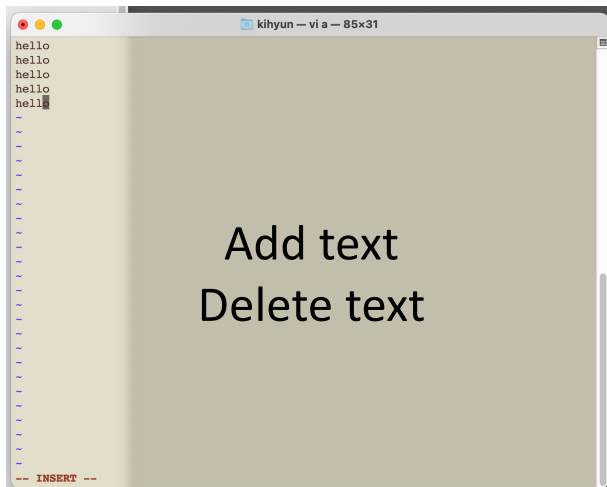


반도체소재설계 강의자료 02

한밭대학교 신소재공학과 신기현

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

- Linux doesn't allow 'space' (띄어쓰기 x)
- Try to use 'tab' all the time (자동완성)

- 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]

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



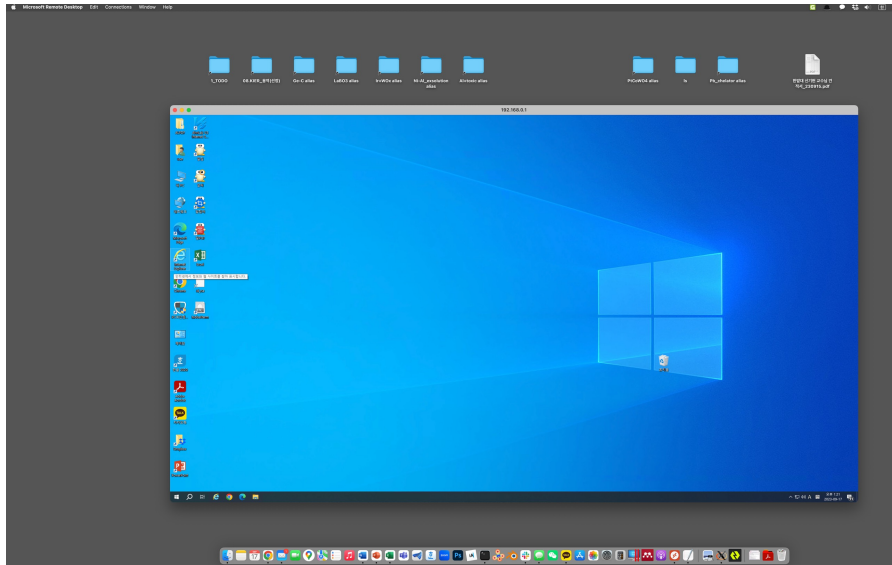
리눅스 명령어 (맥의 터미널, 윈도우 파워셸 유사하게 작동)

- 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)

Google

miniconda

miniconda - Google 검색

miniconda3

miniconda vs anaconda

miniconda uninstall

miniconda windows

miniconda mac

miniconda install ubuntu

miniconda jupyter notebook

miniconda ubuntu

miniconda path

miniconda latest

Search docs

System requirements

Latest Miniconda installer links by Python version

Installing Miniconda

Miniconda release notes

Other resources

Google

miniconda

Videos Images Download Tutorial Vs Anaconda Install Windows Ubuntu

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

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.

Is Miniconda the right conda install for you? The [Anaconda](#) or [Miniconda](#) page lists some reasons why you might want one installation over the other.

- 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	00e037854283680254c7989a9366f1d73448ad5d40766004f6cc0237f40e2914
macOS	Miniconda3 macOS Intel x86 64-bit bash	1622e790f9a0a703d093c2d8153054cd6f6e0d979d931320b95d05258104b
	Miniconda3 macOS Intel x86 64-bit pkg	2236a24306c0e6f36ec324eccc9e631180494c831041791d4461200b697818d04
	Miniconda3 macOS Apple M1 64-bit bash	c0f436d0e130f17103905704fc6609c223f130b7790d33959adc1611835644
	Miniconda3 macOS Apple M1 64-bit pkg	837371f306a8e2d65d0fc8379e0b08120564ff9f40bc04e002f34d9d4f65
Linux	Miniconda3 Linux 64-bit	6340760f5e089c448de4085552097d0c786d49245ed1a8300220b0406de5037
	Miniconda3 Linux-aarch64 64-bit	3962738cfac270ae4f30d0a8328accf6d3385a120640196457747b15774967a
	Miniconda3 Linux-ppc64le 64-bit	92237c284436015005ec004f2f7440140602c0513900993c2f1914043d1d29
	Miniconda3 Linux-s390x 64-bit	22184c07f9a9279c3263ef07fa37357460684f430d05d1f6573c780550

네이버 블로그

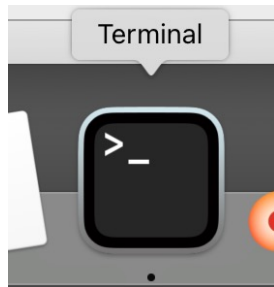
는 패키지들 (dependent package),

의 ...

Miniconda 설치 (MAC)

Platform	Name	SHA256 hash
Windows	Miniconda3 Windows 64-bit	00e8370542836862d4c790aa8966f1d7344a8add4b766004febcb23f40e2914
macOS	Miniconda3 macOS Intel x86 64-bit bash	1622e7a0fa60a7d3d892c2d8153b54cd6ffe3e6b979d931320ba56bd52581d4b
	Miniconda3 macOS Intel x86 64-bit pkg	2236a243b6cbe6f16ec324ecc9e631102494c031d41791b44612bbb6a7a1a6b4
	<u>Miniconda3 macOS Apple M1 64-bit bash</u>	c8f436dbde130f171d39dd7b4fca669c223f130ba7789b83959adc1611a35644
	Miniconda3 macOS Apple M1 64-bit pkg	837371f3b6e8ae2b5bdfc8370e6be812b564ff9f40bcd4eb0b22f84bf9b4fe5
Linux	Miniconda3 Linux 64-bit	634d76df5e489c44ade4085552b97bec786d49245ed1a830022b0b406de5817
	Miniconda3 Linux-aarch64 64-bit	3962738cfac270ae4ff30da0e382aecf6b3305a12064b196457747b157749a7a
	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

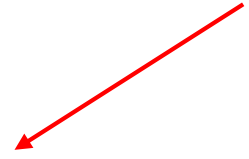
==> For changes to take effect, close and re-open your current shell. <==

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!
kihyun@kihyuns-MacBook-Pro ~ %
```

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



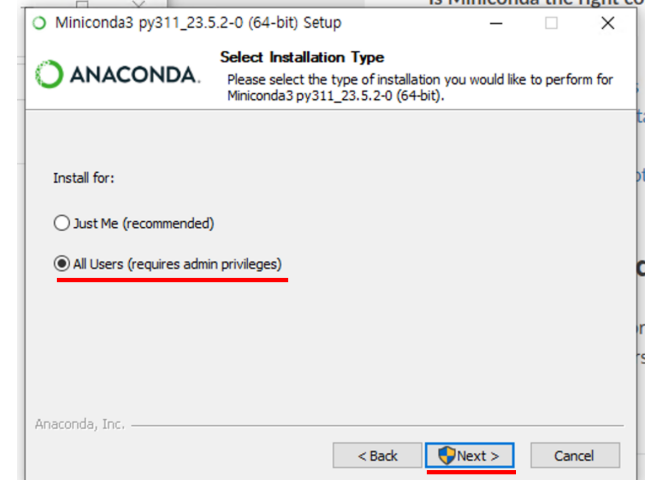
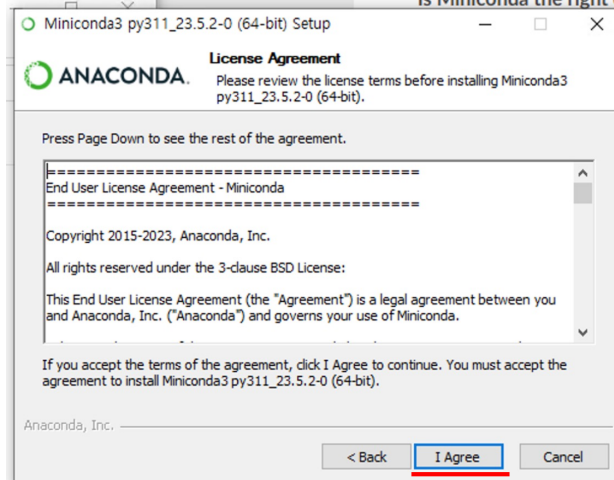
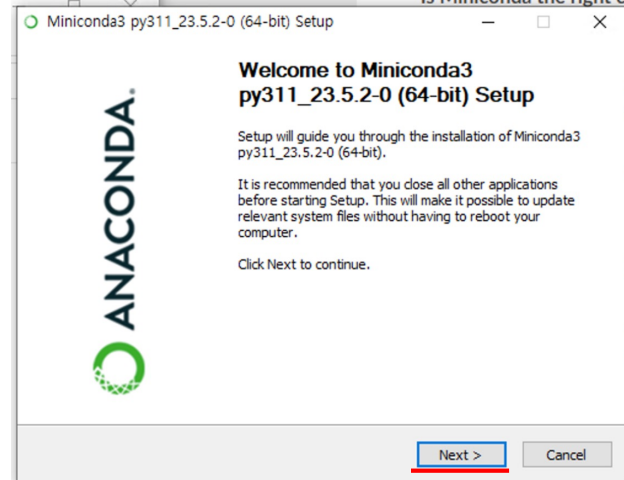
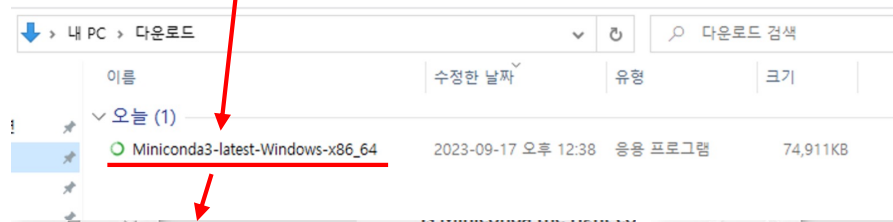
Control + d

(base) 가 뜨면 성공!

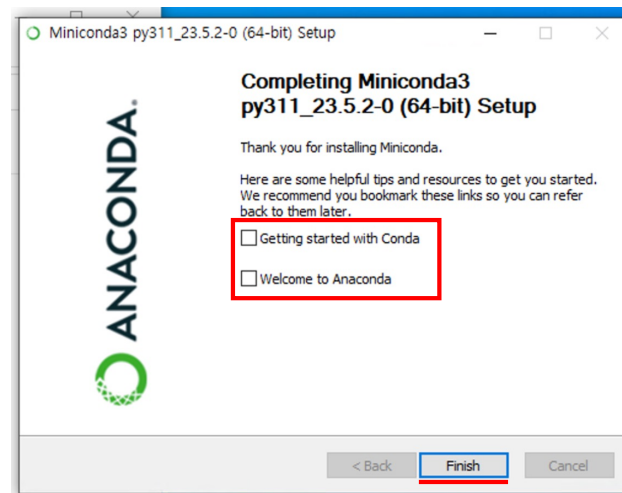
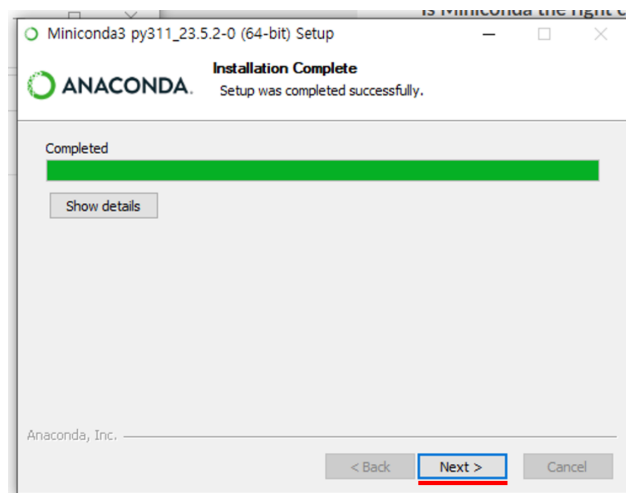
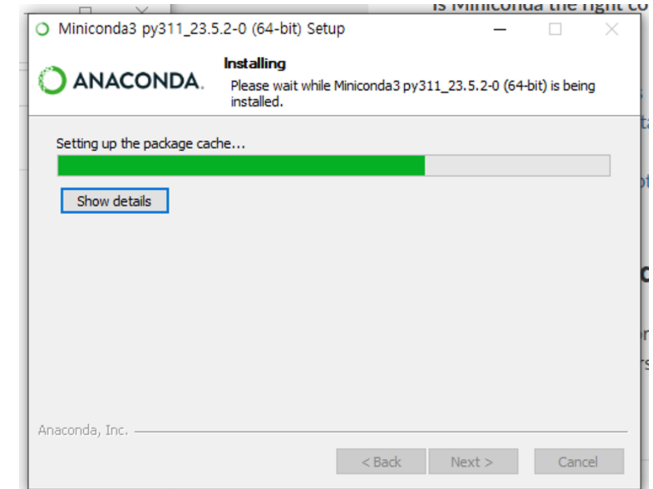
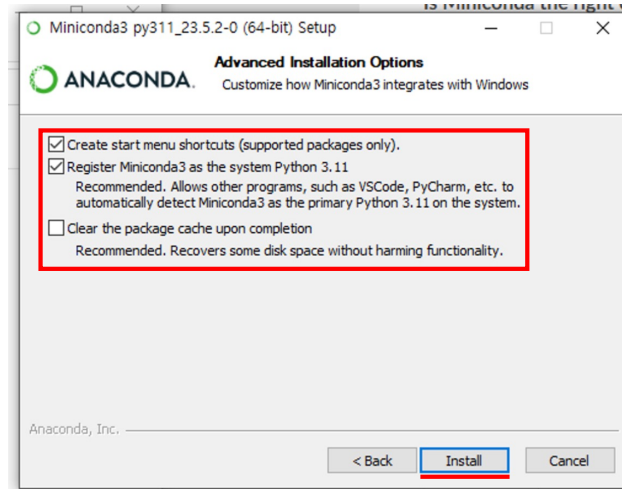
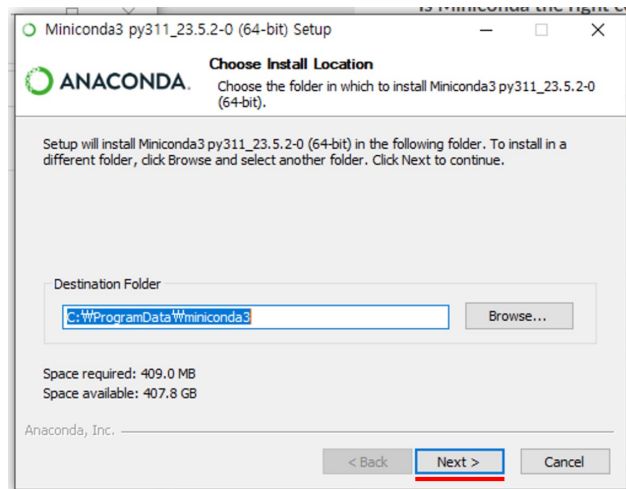


Miniconda 설치 (윈도우)

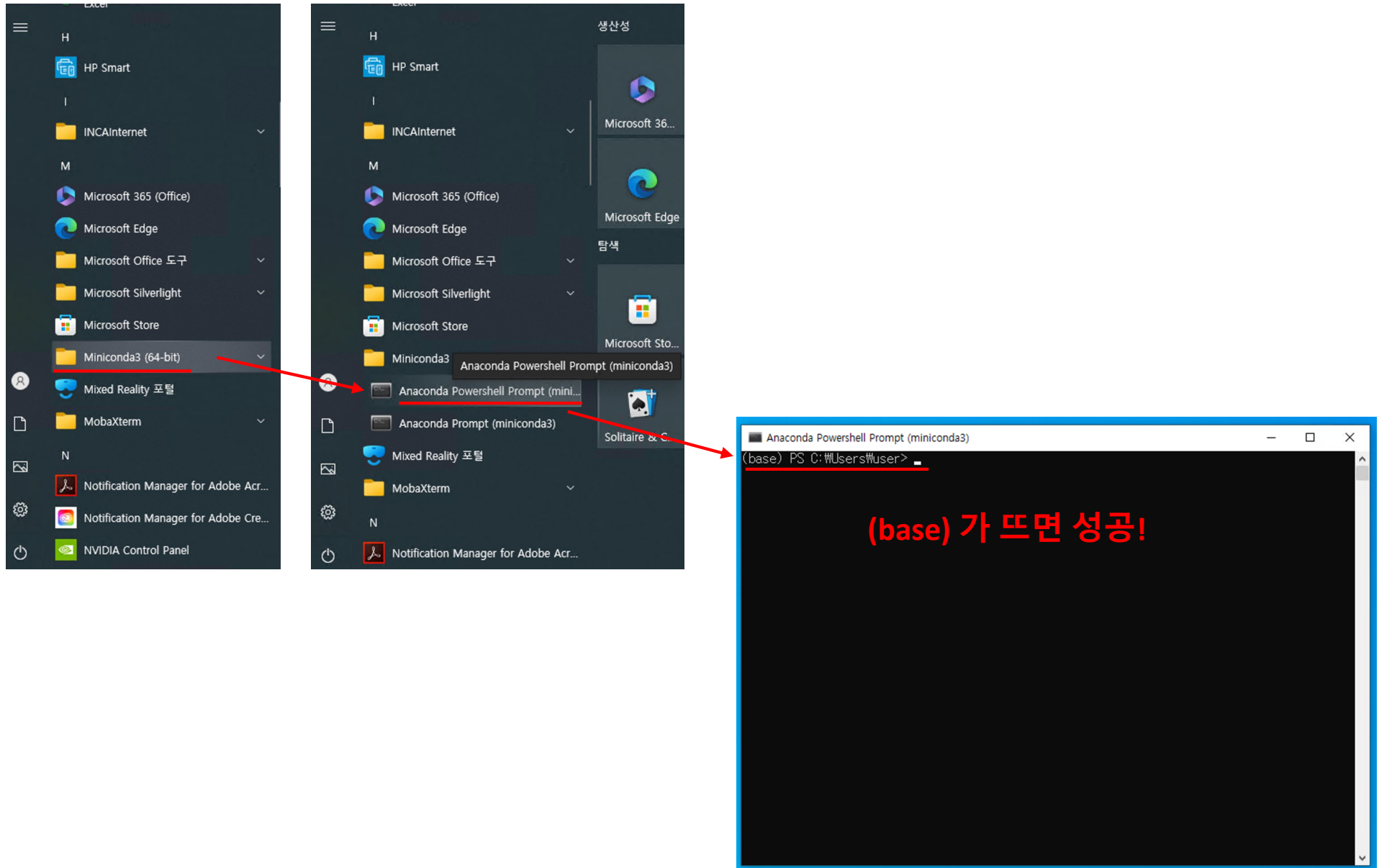
Platform	Name	SHA256 hash
Windows	<u>Miniconda3 Windows 64-bit</u>	00e8370542836862d4c790aa8966f1d7344a8add4b766004febcb23f40e2914
macOS	Miniconda3 macOS Intel x86 64-bit bash	1622e7a0fa60a7d3d892c2d8153b54cd6ffe3e6b979d931320ba56bd52581d4b
	Miniconda3 macOS Intel x86 64-bit pkg	2236a243b6cbe6f16ec324ecc9e631102494c031d41791b44612bbb6a7a1a6b4
	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)



대부분의 명령어

```
Anaconda Powershell Prompt
(base) PS C:\Users\user> ls

디렉터리 : C:\Users\user

Mode                LastWriteTime         Length Name
----                -
d-----          2023-07-10 오후 2:06             .ms-ad
d-r-----        2021-01-27 오전 10:22             3D Objects
d-r-----        2024-03-06 오후 6:16             Contacts
d-r-----        2024-10-09 오후 5:35             Creative Cloud Files
d-r-----        2024-10-08 오후 7:47             Desktop
d-r-----        2024-10-14 오후 8:45             Documents
d-r-----        2024-10-10 오후 2:20             Downloads
d-r-----        2024-10-10 오후 2:21             Dropbox
d-r-----        2024-03-06 오후 6:16             Favorites
d-----        2021-11-18 오후 3:14             Intel
d-r-----        2024-03-06 오후 6:16             Links
d-----        2023-10-17 오후 7:55             miniconda3
d-r-----        2024-03-06 오후 6:16             Music
d-r-----        2021-01-27 오전 10:24             OneDrive
d-r-----        2024-03-06 오후 6:16             Pictures
d-r-----        2024-03-06 오후 6:16             Saved Games
d-r-----        2024-03-06 오후 6:16             Searches
d-r-----        2024-03-06 오후 6:16             Videos
d-----        2024-10-08 오후 8:04             vimfiles
-a-----        2024-10-08 오후 8:04             946 .test.un~
-a-----        2024-10-08 오후 8:04             920 _viminfo

(base) PS C:\Users\user>
```

```
(base) PS C:\Users\user> cd .\Desktop\
(base) PS C:\Users\user\Desktop> cd ../
(base) PS C:\Users\user> cd .\Desktop\
(base) PS C:\Users\user\Desktop> mkdir test
```

디렉터리 : C:\Users\user\Desktop

Mode	LastWriteTime	Length	Name
d-----	2024-10-14 오후 9:13		test

```
(base) PS C:\Users\user\Desktop> rmdir test
```

```
(base) PS C:\Users\user\Desktop\test> mv .\a\ d
(base) PS C:\Users\user\Desktop\test> ls
```

디렉터리 : C:\Users\user\Desktop\test

Mode	LastWriteTime	Length	Name
d-----	2024-10-14 오후 9:14		b
d-----	2024-10-14 오후 9:14		c
d-----	2024-10-14 오후 9:14		d

```
(base) PS C:\Users\user\Desktop\test> cp .\d\ f
(base) PS C:\Users\user\Desktop\test> ls
```

디렉터리 : C:\Users\user\Desktop\test

Mode	LastWriteTime	Length	Name
d-----	2024-10-14 오후 9:14		b
d-----	2024-10-14 오후 9:14		c
d-----	2024-10-14 오후 9:14		d
d-----	2024-10-14 오후 9:14		f

대부분의 명령어가 무리없이 작동한다.



Vi editor 작동 X

```
Anaconda Powershell Prompt
(base) PS C:\Users\user\Desktop\test> vi
vi : 'vi' 용어가 cmdlet, 함수, 스크립트 파일 또는 실행할 수 있는 프로그램 이름으로 인식되지 않습니다. 이름이 정확한지
확인하고 경로가 포함된 경우 경로가 올바른지 검증한 다음 다시 시도하십시오.
위치 줄:1 문자:1
+ vi
+ ~~~
+ CategoryInfo          : ObjectNotFound: (vi:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

(base) PS C:\Users\user\Desktop\test> vi test
vi : 'vi' 용어가 cmdlet, 함수, 스크립트 파일 또는 실행할 수 있는 프로그램 이름으로 인식되지 않습니다. 이름이 정확한지
확인하고 경로가 포함된 경우 경로가 올바른지 검증한 다음 다시 시도하십시오.
위치 줄:1 문자:1
+ vi test
+ ~~~
+ CategoryInfo          : ObjectNotFound: (vi:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

(base) PS C:\Users\user\Desktop\test> |
```

Vi editor 가 작동하지 않는다.

Vim text editor 설치

The image shows a Google search for 'vim' and the resulting Vim website. The website has a 'Download' button highlighted with a red box. A red arrow points from this button to the 'gvim 9.1.0 x86 signed.exe' link in the 'MS-Windows' section of the 'Downloading Vim' page.

Google Search Results:

- Search: vim
- Results: Vim
- URL: <https://www.vim.org>
- Buttons: [Download](#), [About Vim](#), [Documentation](#)

Vim.org Download Page:

- SPONSOR VOTE: Vim development for features
- not logged in ([login](#))
- ENHANCED BY Google
- Search
- Home
- Advanced search
- About Vim
- Community
- News
- Sponsoring
- Trivia
- Documentation
- Download
- Vim from GitHub
- Vim from Mercurial
- List of Mirrors
- Sources
- Patches
- Development
- Runtime files
- Script links
- Translations

Downloading Vim

Vim is available for many different systems and there are several versions

Most popular:

Recent and signed MS-Windows files are available on the [win32](#) page. The current stable version is [gvim 9.1.0000_x64.exe](#) (64bit installer). A zip package (32bit and 64bit) is also available: [gvim 9.1.0000_x86.zip](#) and [gvim 9.1.0000_x64.zip](#).

Signed MS-Windows builds are available from the [win32](#) page.

MS-Windows:

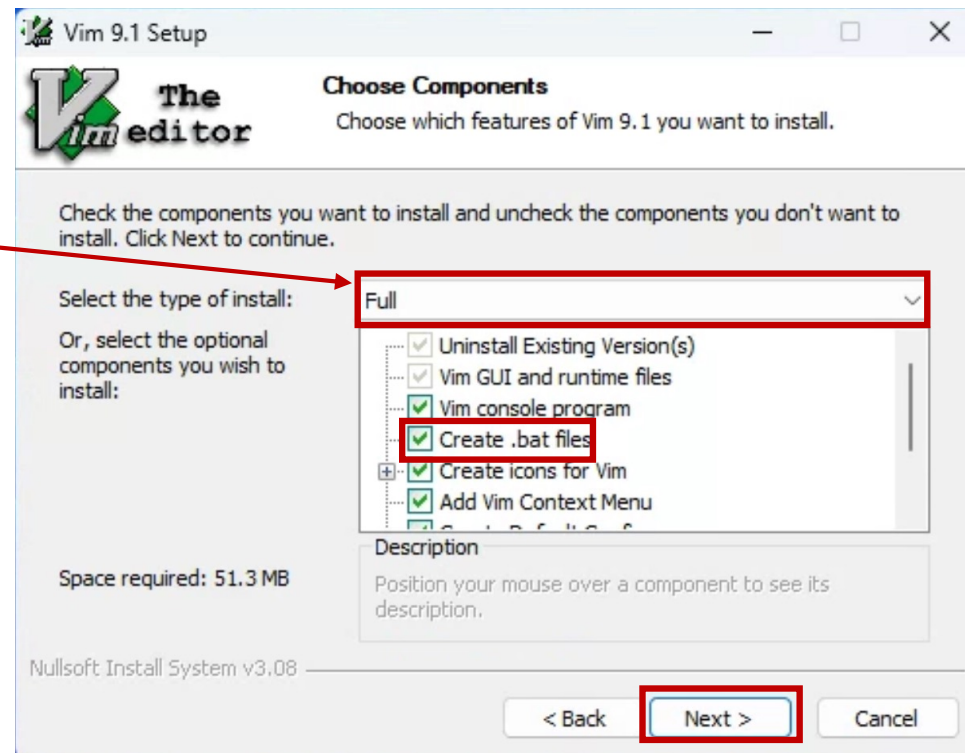
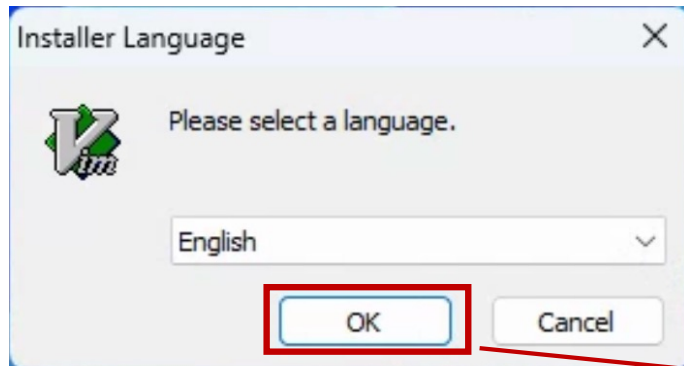
- [gvim 9.1.0 x86 signed.exe](#) (32bit installer)
- [gvim 9.1.0 x64 signed.exe](#) (64bit installer)
- [gvim 9.1.0 x86 signed.zip](#) (32bit zip package)
- [gvim 9.1.0 x64 signed.zip](#) (64bit zip package)

Winget packages are also available: [vim.vim \(stable\)](#) and [vim.vim \(nightly\)](#).

Unix: See the [GitHub](#) page, or [Mercurial](#), if you prefer that. The [MacVim](#) project for a GUI version and [Homebrew](#) for a command-line version.

Mac: See the [MacVim](#) project for a GUI version and [Homebrew](#) for a command-line version.

Vim text editor 설치



```
Anaconda Powershell Prompt
(base) PS C:\Users\user> vim test
```

vim editor 가 성공적으로 작동

참고

The image shows a Windows File Explorer window on the left and an Anaconda PowerShell Prompt window on the right.

File Explorer: The address bar shows the path `test`. The left sidebar shows the navigation pane with `test` selected. The main pane displays a list of folders:

이름	수정된 날짜	유형	크기
b	2024-10-14 오후 9:14	파일 폴더	
c	2024-10-14 오후 9:14	파일 폴더	
d	2024-10-14 오후 9:14	파일 폴더	
f	2024-10-14 오후 9:14	파일 폴더	

Anaconda PowerShell Prompt: The prompt shows the following commands and output:

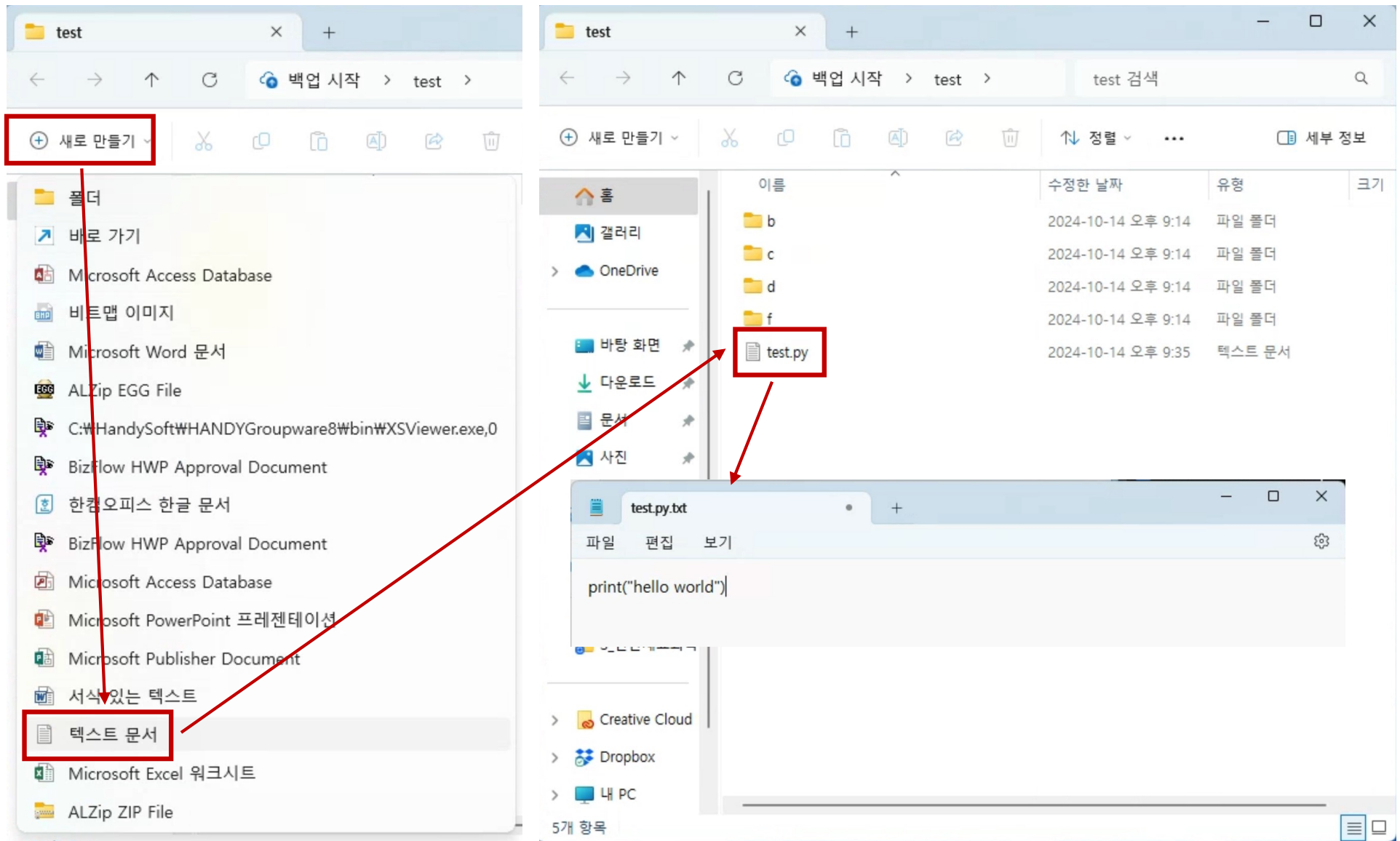
```
(base) PS C:\Users\user> cd .\Desktop\  
(base) PS C:\Users\user\Desktop> cd .\test\  
(base) PS C:\Users\user\Desktop\test> ls
```

The output shows the current directory: `C:\Users\user\Desktop\test`. Below this, a table lists the files in the directory:

Mode	LastWriteTime	Length	Name
d-----	2024-10-14 오후 9:14		b
d-----	2024-10-14 오후 9:14		c
d-----	2024-10-14 오후 9:14		d
d-----	2024-10-14 오후 9:14		f

The prompt then shows the command `(base) PS C:\Users\user\Desktop\test>`.

참고



참고

The screenshot displays two windows side-by-side. The left window is a Windows File Explorer titled 'test', showing a directory with the following contents:

이름	수정한 날짜	유형	크기
b	2024-10-14 오후 9:14	파일 폴더	
c	2024-10-14 오후 9:14	파일 폴더	
d	2024-10-14 오후 9:14	파일 폴더	
f	2024-10-14 오후 9:14	파일 폴더	
test.py	2024-10-14 오후 9:39	텍스트 문서	

The right window is an Anaconda PowerShell Prompt titled 'Anaconda Powershell Prompt'. It shows the following commands and output:

```
(base) PS C:\Users\user\Desktop\test> ls

디렉터리: C:\Users\user\Desktop\test

Mode                LastWriteTime         Length Name
----                -
d-----          2024-10-14 오후 9:14             b
d-----          2024-10-14 오후 9:14             c
d-----          2024-10-14 오후 9:14             d
d-----          2024-10-14 오후 9:14             f
-a-----          2024-10-14 오후 9:39            20 test.py.txt

(base) PS C:\Users\user\Desktop\test> python .\test.py.txt
hello world
(base) PS C:\Users\user\Desktop\test> |
```

ChatGPT: Prompt Engineering

1. 페르소나 패턴

- 특정 인물이나 역할을 상상하여 질문에 답하는 것.
- 답변의 패턴을 특정 영역으로 제한 가능 (**파이썬 전문가**)

2. 레시피 패턴

- 단계별 지시 사항을 따르는 방식으로 답변하는 것.
- 각각의 요소를 알고 있는 경우에 큰 도움 (**특정 라이브러리**)

3. 리플렉션 패턴

- 모든 답변에 대한 이유를 설명하도록 GPT에게 요청하는 방식.
- 답변을 이해하는데 도움 (**디버깅**) + 가짜를 구별하기 쉽다

4. 거부 차단기 패턴

- GPT가 어떻게든 답변을 하게끔 유도하는 방법
- 지식제한, 안전제한에 대한 부분을 회피 가능 (**계속 에러가 나는 경우**)

5. 뒤집힌 상호작용 패턴

- 원하는 것을 달성할 때까지 ChatGPT가 우리에게 역으로 질문하게끔 하는 방식
- 원하는 것은 알지만, 중간단계를 모를 때 도움 (**어떻게 접근 해야 할지 모를 때**)



ChatGPT: Prompt Engineering

0. 일반 질문

K

벌크구조 POSCAR를 만드는 코드를 짜줘



POSCAR 파일은 VASP(Vienna Ab initio Simulation Package) 프로그램에서 사용하는 구조 파일입니다. VASP를 사용하여 계산을 수행하려는 구조의 정보를 담고 있는 파일입니다. 아래에 간단한 Python 코드를 제공하여 벌크 구조의 POSCAR 파일을 만드는 방법을 안내해 드리겠습니다. 이 코드는 구조를 직접 지정하는 간단한 예제입니다. 자세한 구조를 입력하려면 구조의 원자 좌표와 기타 정보를 수정해야 할 수 있습니다.

<https://chat.openai.com/share/88fb4665-2830-4db9-aeec-a55049500f16>

ChatGPT: Prompt Engineering

1. 페르소나 패턴

K

본인이 VASP 과 Python 을 오랫동안 다뤄온 전문가라고 가정하고 답변해줘. 효율성과 범용성에 주의를 기울여, 저문가의 입장에서 벌크 물질의 POSCAR를 만드는 코딩을 만들어주세요



VASP (Vienna Ab initio Simulation Package)를 사용하여 벌크 물질의 POSCAR 파일을 만드는 Python 코드를 만들어보겠습니다. VASP의 POSCAR 파일 형식은 특정한 구조 정보를 담고 있으며, 이를 생성하기 위해 물질의 구조 정보가 필요합니다. 예를 들어, 결정 구조의 구획 크기, 원자좌표 등이 필요한 정보입니다.

아래는 간단한 Python 코드 예제입니다. 이 코드는 간단한 큐빅 구조 (face-centered cubic, FCC)의 물질을 예로 들고 있습니다. 여기서는 알루미늄 (Al)을 예로 들겠습니다.

<https://chat.openai.com/share/13dbe0a5-ff5e-4f01-8ed9-28747163d8a2>

ChatGPT: Prompt Engineering

2. 레시피 패턴

K

벌크 구조의 POSCAR를 출력하는 파이썬 코드는 만들고자 합니다. 원하는 물질을 입력받아야하고, 관련된 python library가 있는지 검색하고, 물질의 lattice constant 정보를 알아야하고, 원자의 위치를 알아야 한다고 알고 있습니다. 이를 위한 전체 단계를 알려주고, 누락된 부분이 있다면 그 부분을 채워주고, 불필요한 단계가 있다면 알려주세요. 또 다른 새로운 제안이 있다면 제안해 주세요.



벌크 구조의 POSCAR를 출력하는 파이썬 코드를 작성하는 것은 몇 가지 단계로 나눌 수 있습니다. 아래는 전체적인 단계와 필요한 라이브러리, 그리고 각 단계에서의 구체적인 설명입니다.

1. 필요한 라이브러리 가져오기:

- 파이썬에서는 Materials Project, pymatgen 또는 ASE(Atomic Simulation Environment)와 같은 라이브러리를 사용하여 구조 정보를 다룰 수 있습니다. 이 예시에서는 pymatgen을 사용하겠습니다.

bash

Copy code

```
pip install pymatgen
```

2. 사용자로부터 물질 정보 입력 받기:

- 사용자로부터 물질의 화학식 또는 구조 정보를 입력 받습니다.

<https://chat.openai.com/share/0740c740-b7fe-4d7c-b53a-3fc6061959ff>

ChatGPT: Prompt Engineering

3. 리플렉션 패턴

K

벌크 구조의 POSCAR를 출력하는 파이썬 코드는 만들고자 합니다. 원하는 물질을 입력받아야하고, 관련된 python library가 있는지 검색하고, 물질의 lattice constant 정보를 알아야하고, 원자의 위치를 알아야 한다고 알고 있습니다. 이를 위한 전체 단계를 알려주고, 누락된 부분이 있다면 그 부분을 채워주고, 불필요한 단계가 있다면 알려주세요. 또 다른 새로운 제안이 있다면 제안해 주세요. 모든 답변에 대해서 이유를 설명해 주세요



벌크 구조의 POSCAR 파일을 생성하는 파이썬 코드를 만들려면 다음 단계를 따를 수 있습니다. 이 작업은 두 가지 주요 부분으로 나뉩니다. 먼저, 원하는 물질의 결정 구조 정보를 얻어오고, 그런 다음 POSCAR 파일을 생성하는 코드를 작성해야 합니다.

1. 원하는 물질 결정 구조 정보 수집:

- 먼저, 원하는 물질의 결정 구조 정보를 얻어야 합니다. 이 정보는 결정 구조, 격자 상수, 원자 위치 등을 포함합니다.
- 이 정보를 얻는 방법은 여러 가지가 있을 수 있으며, 대표적인 방법으로는 물질의 결정 구조 데이터베이스에서 정보를 검색하거나, 다른 소프트웨어나 라이브러리를 사용하여 계산 결과를 얻는 것이 있습니다.

2. POSCAR 파일 생성:

- 결정 구조 정보를 기반으로 POSCAR 파일을 생성해야 합니다. POSCAR 파일은 VASP (Vienna Ab initio Simulation Package)와 같은 전자 구조 계산 소프트웨어에서 사용되는 파일 형식입니다.
- POSCAR 파일 형식은 다음과 같습니다:

<https://chat.openai.com/share/86a585c9-7ee2-4aa5-8fde-2fcaecd2a262>



ChatGPT: Prompt Engineering

5. 뒤집힌 상호작용 패턴

K

벌크 구조의 POSCAR를 만들기 위한 python 코딩을 만드려고해. 성공적으로 작동하기 위해서 필요한 모든 질문들을 나에게 해주세요. 필요한 모든 정보를 얻으면 전체 python 코드를 출력해줘



벌크 구조의 POSCAR 파일을 생성하는 Python 코드를 작성하려면 다음과 같은 정보와 단계가 필요합니다. 아래의 질문에 답하시면, 전체 Python 코드를 제공해 드릴 수 있을 것입니다.

1. 어떤 결정 구조(큐빅, 바싸, 페이스센터 큐빅, 등)를 가진 물질을 다루고 있나요?
2. 결정 구조의 단위 격자 상자 크기(a, b, c)를 알려주세요.
3. 격자 상자 각 변의 각도 (alpha, beta, gamma)를 알려주세요.
4. 단위 격자 상자 내에서 원자의 종류 및 수를 제공하세요.
5. 각 원자의 종류와 단위 격자 상자 내 좌표를 알려주세요.
6. 선택적으로, 기타 정보를 POSCAR 파일에 추가하고 싶은 경우, 그 정보도 알려주세요 (예: 선택적 태그, 질량, 전기적 또는 자기 모멘트 등).

위의 정보를 제공하면 필요한 Python 코드를 생성하는 데 도움이 될 것입니다.

<https://chat.openai.com/share/c5f61359-c59e-4a57-b952-7440493733df>

ChatGPT - 실습

1. 사용자가 문자열을 입력 받아 $a \rightarrow w, b \rightarrow x, c \rightarrow y, d \rightarrow z$ 로 변환하는 프로그램을 만드시오.
예시: 입력 : abcdeftz → 출력 : wxyzftz
2. 다음 주어진 문자열에서 모든 대문자를 소문자로 변환하고, 문자 ',' 와 '.'을 제거한 후 각 단어를 순서대로 출력 하는 프로그램을 만드시오. 단 중복되는 단어는 한번만 출력 되어야 하고 단어가 문자열에서 발생한 횟수도 함께 출력한다. 사전을 이용: 단어-key, 발생횟수-value s = 'We propose to start by making it possible to teach programming in Python, an existing scripting language, and to focus on creating a new development environment and teaching materials for it.'
예시 : 입력 받는게 없고, 출력 : We 1번, ... It x번
3. 자연수를 입력 받아 소수 인지, 합성수인지 구분할 수 있는 프로그램을 만드시오.
4. 자연수를 입력 받아 원하는 진수 (2 ~ 16진수)로 변환하는 프로그램을 만드시오.
5. 카카오 api 를 활용하여, '유성온천역' 근처 1km 내의 레스토랑 정보를 모두 리스팅 하는 프로그램을 만드시오.
6. 삼성전자의 주가를 예측하는 프로그램을 만드시오 (2020년의 데이터만을 활용하여, 2021년의 주가를 예측하고, 실제와 얼마나 다른지 비교)

