

Course preparation

by Zheng Hao, Li Zhe, wwy

Operating System

Windows, MacOS, Linux are all OK. You can use Powershell/ Terminal/ Command Window or etc. in your OS.

Connect to remote server

In this course, we maybe use remote server later. SSH is short for Secure Shell, created by Network Working Group of IETF. SSH is a reliable protocol that provides security for remote login sessions and other network services. There are many different ways to achieve connection to remote server, **choose anyone as you like**. Here we only show some examples:

Using terminal

1. Start your terminal

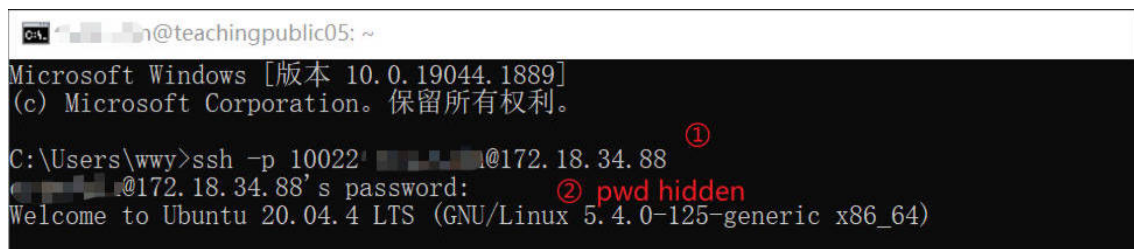
- WIN+R & `cmd.exe`
- Click the Launchpad in the Dock, type `Terminal` in the search field, then click Terminal.

2. SSH to server

```
ssh username@host -p PORT
```

- Temporary server: (users would be deleted after this lab)

- host: 172.18.34.88(for example)
- PORT: 10022
- username:
- password: `_cv`



The screenshot shows a Windows terminal window with the following text:

```
Microsoft Windows [版本 10.0.19044.1889]  
(c) Microsoft Corporation。保留所有权利。  
  
C:\Users\wwy>ssh -p 10022' [redacted]@172.18.34.88 ①  
[redacted]@172.18.34.88's password: ② pwd hidden  
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-125-generic x86_64)
```

Annotations: ① points to the IP address in the command, and ② points to the password prompt.

- SSH command line options:

- `-C`: Compresses all data (including stdin, stdout, stderr, and data for forwarded X11 and TCP connections) for a faster transfer of data.
- `-q`: Suppresses all errors and warnings
- `-v`: Verbose mode. It echoes everything it is doing while establishing a connection. It is very useful in the debugging of connection failures

* Check more command line options in web link <https://www.ssh.com/academy/ssh/command> . If you do not know the meaning of a ssh command, you can check [here](#) .

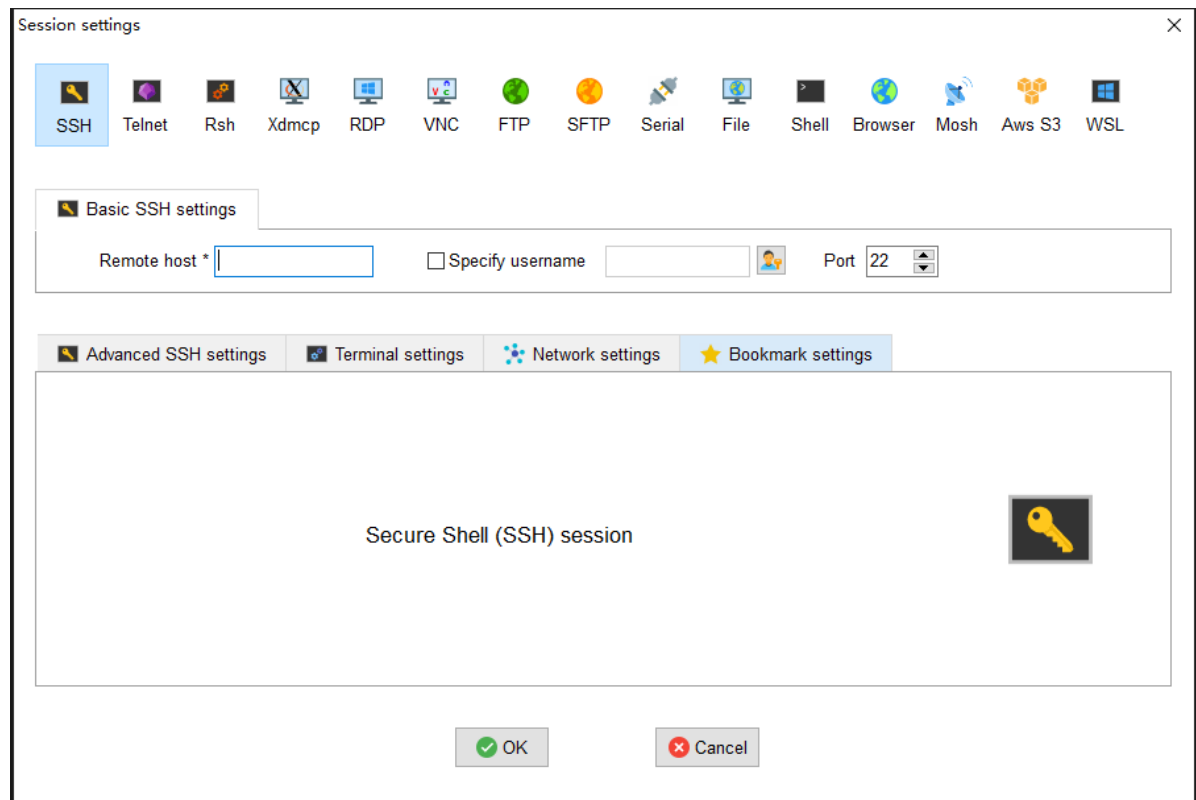
Using Xshell

Xshell is a useful tool for ssh connection. You can apply for the free version.

<https://www.xshell.com/zh/free-for-home-school/>

Using MobaXterm

MobaXterm. Enhanced terminal for Windows with X11 server, tabbed SSH client, network tools and much more.



Necessary tools

Linux basic command

- **pwd**: Use the pwd command to find out the path of the current working directory (folder) you're in.
- **cd**: To navigate through the Linux files and directories, use the cd command. It requires either the full path or the name of the directory (You can use tab to complete the file name), depending on the current working directory that you're in. E.g. **cd ..** (with two dots) to move one directory up.
- **ls**: The ls command is used to view the contents of a directory. By default, this command will display the contents of your current working directory. E.g., **ls -R** will list all the files in the sub-directories as well. **ls -a** will show the hidden files. **ls -al** will list the files and directories with detailed information like the permissions, size, owner, etc.
- **cat**: (short for concatenate) is one of the most frequently used commands in Linux. It is used to list the contents of a file on the standard output (stdout). E.g., **cat filename1 filename2 > filename3** joins two files (1 and 2) and stores the output of them in a new file (3).

- **cp**: Use the cp command to copy files from the current directory to a different directory. E.g., **cp -r dir1/ dir2/** copy the directory dir1 to the dir2 directory.
- **mv**: The primary use of the mv command is to move files, although it can also be used to rename files. E.g., **mv source destination** rename or move file(s) or directories.
- **mkdir**: Use mkdir command to make a new directory.
- **rm**: The rm command is used to delete directories and the contents within them. If you only want to delete the directory — as an alternative to **rmdir** — use **rm -r**. Additionally, **rm -rf *** can remove any file, **but it is very dangerous!***
- **grep**: It lets you search through all the text in a given file. E.g., **grep blue notepad.txt** will search for the word blue in the notepad file. Lines that contain the searched word will be displayed fully. **ps -ef | grep python** pipe the ps query result to grep to find the process that contains a specific string(e.g. python). Then you can use **kill -s 9 PID**(corresponding second string above) to kill program that have not been closed normally

* More command listed in Appendix A

Anaconda or Miniconda

They are both management tool for Python. Anaconda is a Python distribution preloaded with a large number of scientific computing and data analysis related packages, including conda, Python, and more than 150 scientific packages and their dependencies. Miniconda offers more flexibility as a more lightweight option that only includes Python and Conda, with no other libraries pre-installed. The choice between Anaconda and Miniconda depends on the user's specific needs and resources. Here we take Miniconda for example:

1. Download package

- Windows: https://mirrors.sustech.edu.cn/anaconda/miniconda/Miniconda3-py310_23.11.0-2-Windows-x86_64.exe
- Mac:
 - Non-M series CPU: https://mirrors.sustech.edu.cn/anaconda/miniconda/Miniconda3-py310_23.11.0-2-MacOSX-x86_64.sh
 - M series CPU: https://mirrors.sustech.edu.cn/anaconda/miniconda/Miniconda3-py310_23.11.0-2-MacOSX-arm64.sh
- Linux: https://mirrors.sustech.edu.cn/anaconda/miniconda/Miniconda3-py310_23.11.0-2-Linux-x86_64.sh

2. Install

- For Mac: Check [official document](#) for details

The image shows the Anaconda Documentation website on the left and a macOS installation guide on the right. The documentation sidebar on the left has 'Miniconda' selected, with 'Installing Miniconda' highlighted by a red arrow. The macOS guide on the right has tabs for 'Windows Command Prompt', 'Windows PowerShell', 'macOS' (selected), and 'Linux'. It lists four commands to download and install Miniconda. Below the commands, there are tabs for 'Apple Silicon' and 'Intel', with a code block showing the installation commands for Apple Silicon. A red arrow points to the 'macOS' tab, and another points to the 'Installing Miniconda' link in the sidebar.

Anaconda Documentation

Getting Started

- Getting started with Anaconda
- Anaconda Distribution
- Miniconda**
 - System requirements
 - Installing Miniconda**
 - Miniconda release notes
 - Uninstalling Miniconda
- Anaconda Learning

Tools

- Working with conda
- Anaconda Navigator
- Anaconda Notebooks

macOS

1. Run the following four commands to download and install the latest macOS installer for your chosen chip architecture. Line by line, these commands:

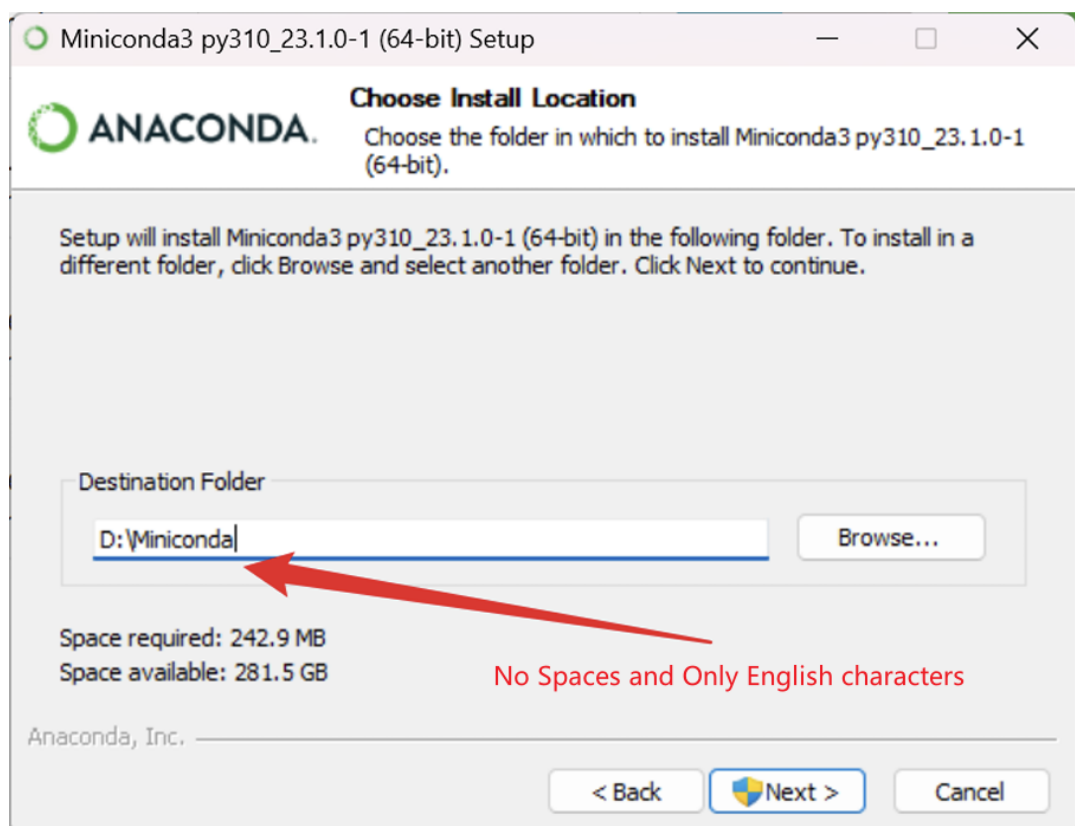
- create a new directory named "miniconda3" in your home directory.
- download the macOS Miniconda installation script for your chosen chip architecture and save the script as `miniconda.sh` in the miniconda3 directory.
- run the `miniconda.sh` installation script in silent mode using bash.
- remove the `miniconda.sh` installation script file after installation is complete.

Apple Silicon **Intel**

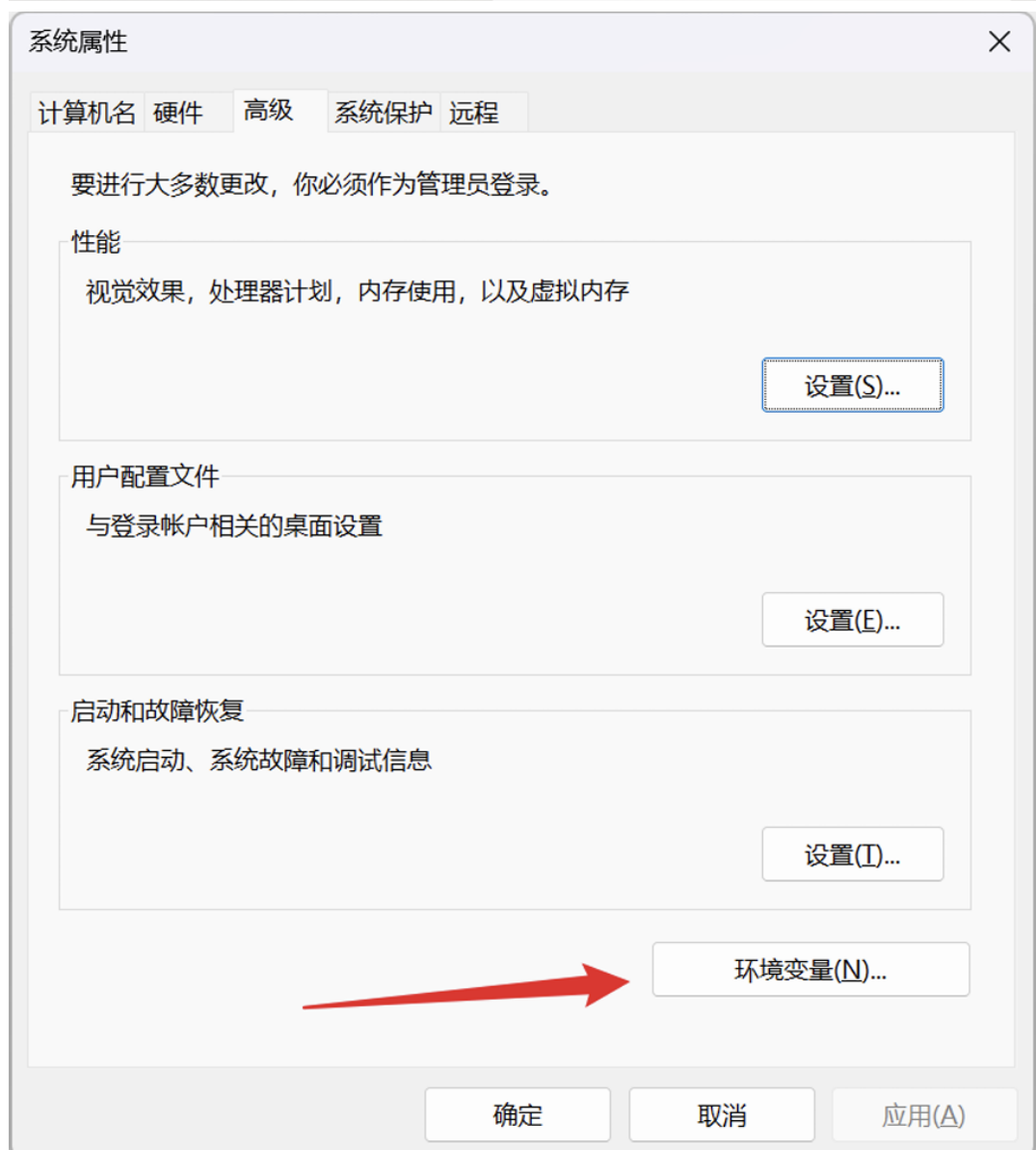
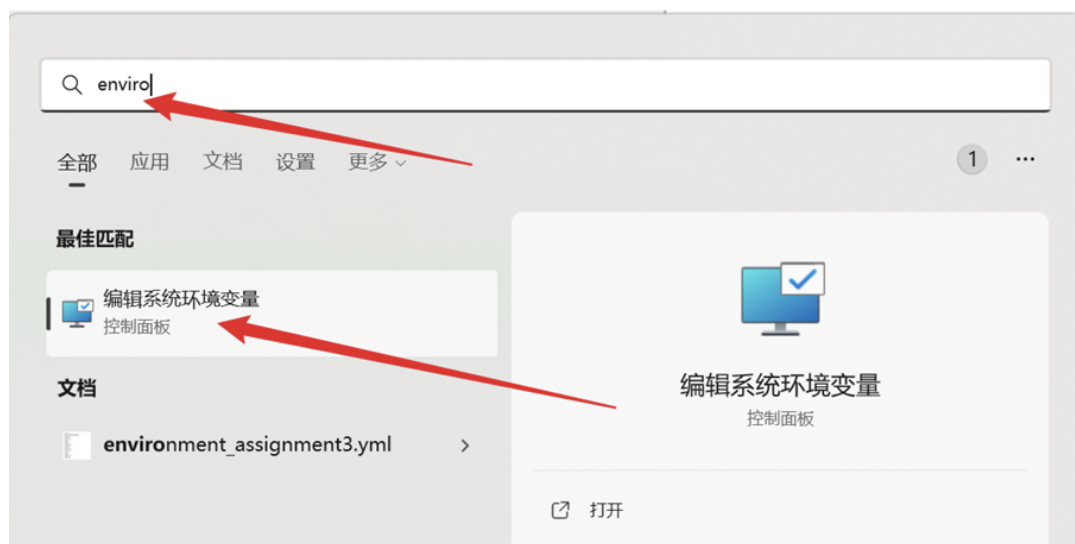
```
mkdir -p ~/miniconda3
curl https://repo.anaconda.com/miniconda/Miniconda3-latest-MacOSX-arm64.pkg -o ~/miniconda3/miniconda.pkg
bash ~/miniconda3/miniconda.sh -b -u -p ~/miniconda3
rm ~/miniconda3/miniconda.pkg
```

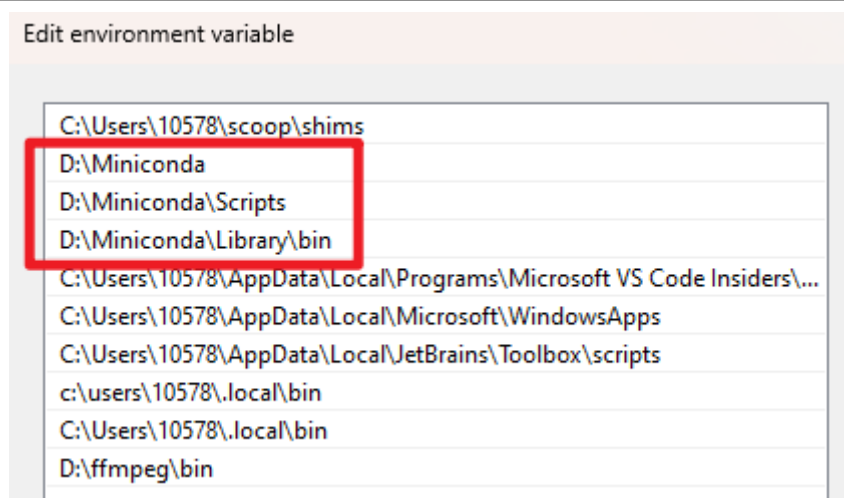
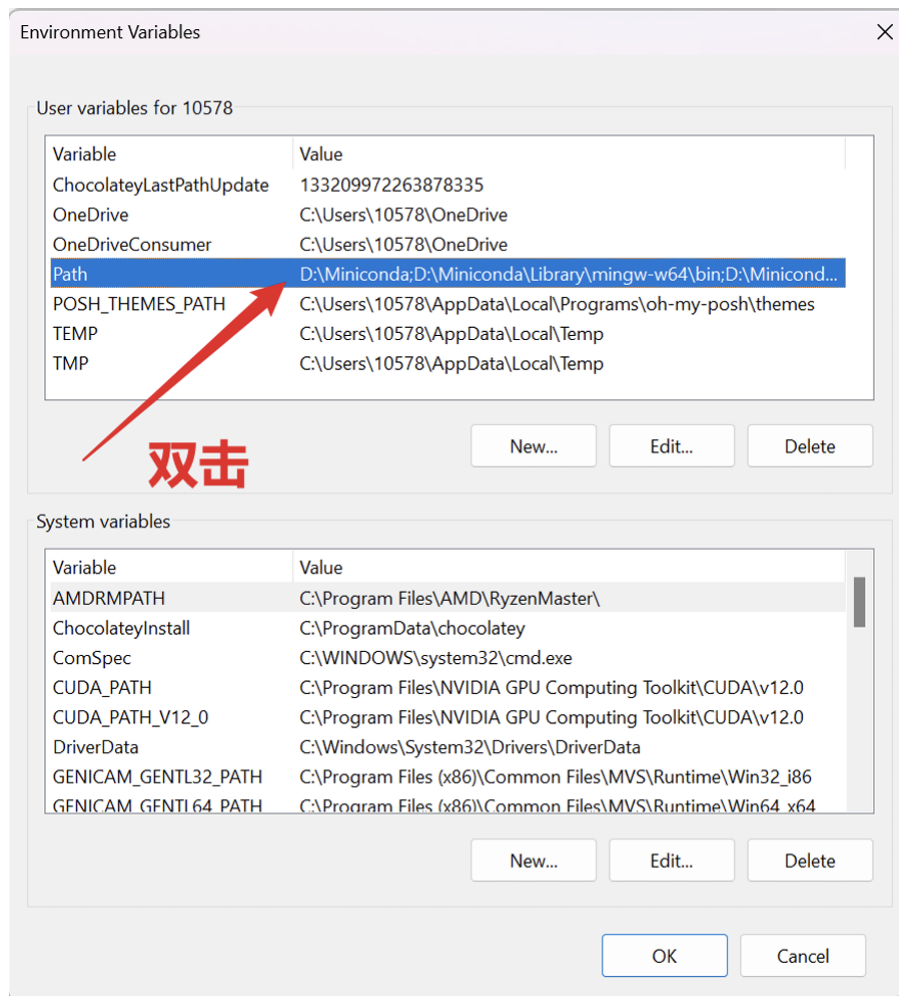
↑ Back to top

- For Linux: Same with Mac customer, check [official document](#) for details.
- For Win:
 - a. Double click the exe download in step 1
 - b. Follow the guide until finish. Notice the following steps



- c. Add Environment variables.





d. Check the conda, open the Powershell/command window

```
Windows PowerShell X amax9 X user@baoder: ~ X + v
(teaching) PS C:\Users\10578> conda info

      active environment : teaching
      active env location : D:\Miniconda\envs\teaching
      shell level         : 2
      user config file    : C:\Users\10578\.condarc
      populated config files : C:\Users\10578\.condarc
      conda version       : 23.11.0
      conda-build version : not installed
      python version      : 3.11.5.final.0
      solver              : libmamba (default)
      virtual packages    : __archspec=1=x86_64
                          __conda=23.11.0=0
                          __cuda=12.4=0
                          __win=0=0
      base environment    : D:\Miniconda (writable)
      conda av data dir   : D:\Miniconda\etc\conda
      conda av metadata url : None
      channel URLs        : https://repo.anaconda.com/pkgs/main/win-64
                          https://repo.anaconda.com/pkgs/main/noarch
                          https://repo.anaconda.com/pkgs/r/win-64
                          https://repo.anaconda.com/pkgs/r/noarch
                          https://repo.anaconda.com/pkgs/msys2/win-64
                          https://repo.anaconda.com/pkgs/msys2/noarch
      package cache       : D:\Miniconda\pkgs
                          C:\Users\10578\.conda\pkgs
                          C:\Users\10578\AppData\Local\conda\conda\pkgs
      envs directories    : D:\Miniconda\envs
                          C:\Users\10578\.conda\envs
                          C:\Users\10578\AppData\Local\conda\conda\envs
      platform            : win-64
      user-agent          : conda/23.11.0 requests/2.31.0 CPython/3.11.5 Windows/10 Windows/10.0.22635 solver/libmamba con
da-libmamba-solver/23.12.0 libmambapy/1.5.3
      administrator      : False
      netrc file         : None
      offline mode       : False

(teaching) PS C:\Users\10578> |
```

3. Change conda's default source to Sustech source

Run command in your terminal

```
conda config --add channels
https://mirrors.sustech.edu.cn/anaconda/pkgs/free/
conda config --add channels
https://mirrors.sustech.edu.cn/anaconda/pkgs/main/
conda config --set show_channel_urls yes
```

Open the configuration file(this path only for Mac/Linux) use vim(detail in Appendix A)

```
vim ~/.condarc
```

Paste the following lines to the file

```
channels:
- defaults
show channel urls : true
default channels :
- https://mirrors.sustech.edu.cn/anaconda/pkgs/main
- https://mirrors.sustech.edu.cn/anaconda/pkgs/free
- https://mirrors.sustech.edu.cn/anaconda/pkgs/r
- https://mirrors.sustech.edu.cn/anaconda/pkgs/pro
- https://mirrors.sustech.edu.cn/anaconda/pkgs/msys2
custom channels:
conda-forge: https://mirrors.sustech.edu.cn/anaconda/cloud
msys2: https://mirrors.sustech.edu.cn/anaconda/cloud
bioconda: https://mirrors.sustech.edu.cn/anaconda/cloud
menpo: https://mirrors.sustech.edu.cn/anaconda/cloud
pytorch: https://mirrors.sustech.edu.cn/anaconda/cloud
simpleitk : https://mirrors.sustech.edu.cn/anaconda/cloud
nvidia : https://mirrors.sustech.edu.cn/anaconda-extra/cloud
```

* **Reference link:** <https://mirrors.sustech.edu.cn/help/anaconda.html>

** **Different path for OS**

- **Linux:** \$HOME/.condarc
 - **MacOS:** \$HOME/.condarc
 - **Windows:** C:\Users<YourUserName>.condarc
4. Pip mirror setting, also change to Sustech source

```
pip install --upgrade pip --index-url
https://mirrors.sustech.edu.cn/pypi/simple
pip config set global.index-url https://mirrors.sustech.edu.cn/pypi/simple
```

Jupyter

1. Create your environment

```
conda create -n myenv
```

```
C:\Users\10578>conda create -n myenv
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 4.13.0
  latest version: 22.9.0

Please update conda by running

  $ conda update -n base -c defaults conda

## Package Plan ##

  environment location: D:\Anaconda3\envs\myenv

Proceed ([y]/n)? | enter y here
```

Activate this environment `myenv`

```
conda activate myenv
```

2. Install jupyterlab(remember to turn off your VPN)

```
pip install jupyterlab
```

3. Launch jupyter-lab

You can use command window or just App icon to launch.

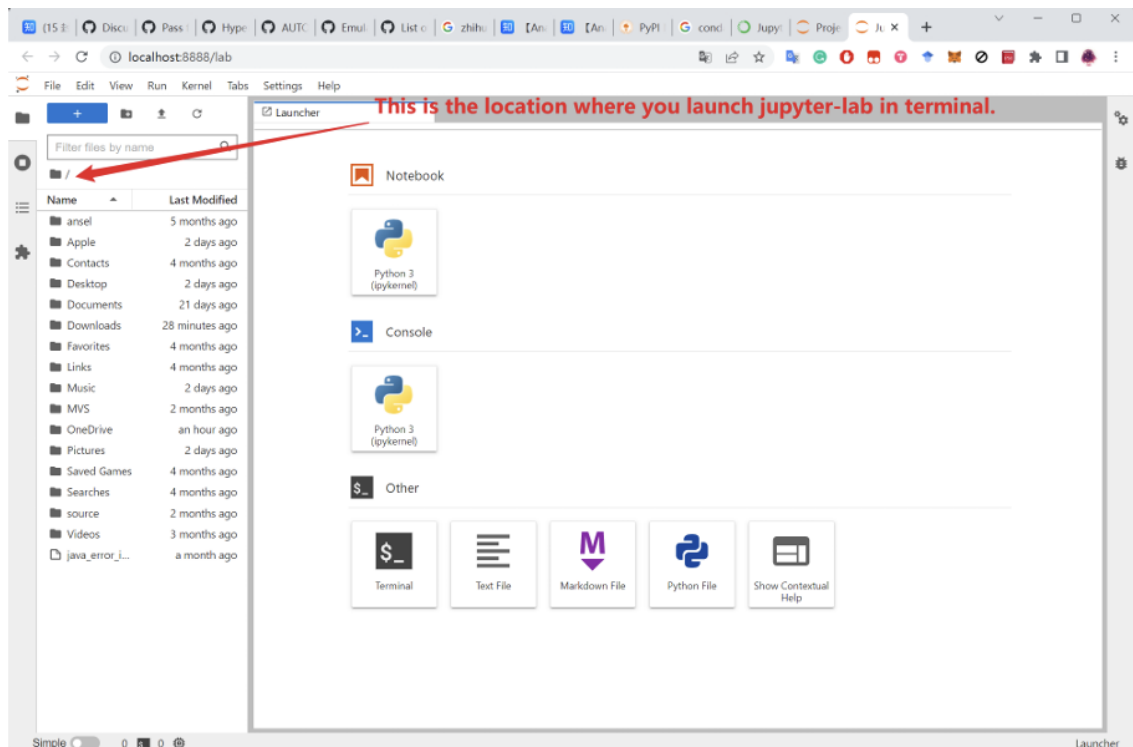

```

C:\Users\10578>jupyterlab
[I 2022-10-26 18:06:38.496 ServerApp] jupyterlab | extension was successfully linked.
[I 2022-10-26 18:06:38.509 ServerApp] Writing Jupyter server cookie secret to C:\Users\10578\AppData\Roaming\jupyter\runtime\jupyter_cookie_secret
[I 2022-10-26 18:06:39.031 ServerApp] nbclassic | extension was successfully linked.
[I 2022-10-26 18:06:39.031 ServerApp] panel.io.jupyter_server_extension | extension was successfully linked.
[I 2022-10-26 18:06:39.101 ServerApp] nbclassic | extension was successfully loaded.
[I 2022-10-26 18:06:39.102 LabApp] JupyterLab extension loaded from D:\Anaconda3\lib\site-packages\jupyterlab
[I 2022-10-26 18:06:39.102 LabApp] JupyterLab application directory is D:\Anaconda3\share\jupyter\lab
[I 2022-10-26 18:06:39.105 ServerApp] jupyterlab | extension was successfully loaded.
/panel-preview/ /panel-preview
[I 2022-10-26 18:06:39.106 ServerApp] panel.io.jupyter_server_extension | extension was successfully loaded.
[I 2022-10-26 18:06:39.108 ServerApp] Serving notebooks from local directory: C:\Users\10578
[I 2022-10-26 18:06:39.108 ServerApp] Jupyter Server 1.13.5 is running at:
[I 2022-10-26 18:06:39.108 ServerApp] http://localhost:8888/lab?token=408f491788b852e603b10b761d38bd0aee4bfeac6fff8348
[I 2022-10-26 18:06:39.108 ServerApp] or http://127.0.0.1:8888/lab?token=408f491788b852e603b10b761d38bd0aee4bfeac6fff8348
[I 2022-10-26 18:06:39.108 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 2022-10-26 18:06:39.161 ServerApp]

To access the server, open this file in a browser:
file:///C:/Users/10578/AppData/Roaming/jupyter/runtime/jpserver-12468-open.html
Or copy and paste one of these URLs:
http://localhost:8888/lab?token=408f491788b852e603b10b761d38bd0aee4bfeac6fff8348
or http://127.0.0.1:8888/lab?token=408f491788b852e603b10b761d38bd0aee4bfeac6fff8348
[W 2022-10-26 18:06:44.005 LabApp] Could not determine jupyterlab build status without nodejs

```

It will shown in your web page.



Pytorch

1. Go to [Pytorch](#)

- **pwd**: Use the pwd command to find out the path of the current working directory (folder) you're in.
- **cd**: To navigate through the Linux files and directories, use the cd command. It requires either the full path or the name of the directory (You can use tab to complete the file name), depending on the current working directory that you're in. E.g. **cd ..** (with two dots) to move one directory up.
- **ls**: The ls command is used to view the contents of a directory. By default, this command will display the contents of your current working directory. E.g., **ls -R** will list all the files in the sub-directories as well. **ls -a** will show the hidden files. **ls -al** will list the files and directories with detailed information like the permissions, size, owner, etc.
- **cat**: (short for concatenate) is one of the most frequently used commands in Linux. It is used to list the contents of a file on the standard output (stdout). E.g., **cat filename1 filename2 > filename3** joins two files (1 and 2) and stores the output of them in a new file (3).
- **cp**: Use the cp command to copy files from the current directory to a different directory. E.g., **cp -r dir1/ dir2/** copy the directory dir1 to the dir2 directory.
- **mv**: The primary use of the mv command is to move files, although it can also be used to rename files. E.g., **mv source destination** rename or move file(s) or directories.
- **mkdir**: Use mkdir command to make a new directory.
- **rm**: The rm command is used to delete directories and the contents within them. If you only want to delete the directory — as an alternative to **rmdir** — use **rm -r**. Additionally, **rm -rf *** can remove any file, **but it is very dangerous!***
- **grep**: It lets you search through all the text in a given file. E.g., **grep blue notepad.txt** will search for the word blue in the notepad file. Lines that contain the searched word will be displayed fully. **ps -ef | grep python** pipe the ps query result to grep to find the process that contains a specific string (e.g. python). Then you can use **kill -s 9 PID** (corresponding second string above) to kill program that have not been closed normally
- **df**: Use df command to get a report on the system's disk space usage, shown in percentage and KBs. If you want to see the report in human readable format, type **df -h**.
- **du**: If you want to check how much space a file or a directory takes, the du (Disk Usage) command is the answer. E.g., **du -h -max-depth=1 2>/dev/null** to check the size of each folder in the current path, and ignore warning messages such as permission denied.
- **tar**: The tar command is the most used command to archive multiple files into a tarball — a common Linux file format that is similar to zip format, with compression being optional. E.g., **tar -zcvf filename.tar.gz /folder** gzip compression and **tar -zxvf filename.tar.gz** unzip files in current directory
- **chmod**: chmod is another Linux command, used to change the read, write, and execute permissions of files and directories. As this command is rather complicated, you can read the full tutorial in order to execute it properly.
- **wget**: The Linux command line is super useful — you can even download files from the internet with the help of the wget command. To do so, simply type wget followed by the download link.
- **top**: As a terminal equivalent to Task Manager in Windows, the top command will display a list of running processes and how much CPU each process uses. **htop** is similar to the top command and can perform related operations (killing, renicing) on this process without inputting its PID. **htop** is an interactive process viewer in Linux system.
- **man**: Confused about the function of certain Linux commands? Don't worry, you can easily learn how to use them right from Linux's shell by using the man command. For instance, entering **man tail** will show the manual instruction of the tail command.
- **echo**: This command is used to move some data into a file. For example, if you want to add the text, "Hello, my name is John" into a file called name.txt, you would type **echo Hello, my name is John >> name.txt**

- **history:** Gives a list of all past commands typed in the current terminal session. You can use the arrow keys up and down to display historical commands.
- **clear:** Clear a command line screen/window for a fresh start.

screen command in Linux provides the ability to launch and use multiple shell sessions from a single ssh session. Sometimes we need to run some tasks that take a long time to complete, such as system backup, ftp transfer, and so on. During this time, you cannot close the window or disconnect, otherwise the task will be killed. Screen can help us solve the above problem.

```
- **screen -ls** List all current sessions.
- **screen -S** your_session_name Create a new session called your_session_name.
- **ctrl** + **a** + **d** Detach, leave the current session temporarily, throw
the current screen session (may contain multiple windows) to the background for
execution, and return to the state when it has not entered the screen.
- **screen -r** your_session_name Resume offline screen jobs.
- **screen -S** your_session_name **-X quit** kill a screen or you can also enter
the corresponding session and then **exit**.
```

Vim is a text editor developed from vi. Vim has two mode: Insert mode(Where you can just type like normal text editor. Press **i** for insert mode), Command mode(Where you give commands to the editor to get things done . Press **ESC** for command mode). Most of them below are in command mode

- a - inserts text after the cursor.
- x - to delete the unwanted character.
- :wq - to save and exit
- :q! - to trash all changes
- u - to undo the last the command and U to undo the whole line.
- ctrl + r - to redo
- 0 - to move to the start of the line.
- \$ - to the end of line
- G - to move you to the bottom of the file
- gg - to move you to the start of the file
- kG - 50G jump to line 50. k G jump to line k.
- h,j,k,l - left, down, up, right. 20j move down 20 lines. kj move down k lines.
- dw - move the cursor to the beginning of the word to delete that word and d2w deletes 2 words. Number can be changed for deleting the number of consecutive words like d3w
- dd - to delete the line and kdd to delete to k lines.
- yy - to copy the line and kyy to copy k lines.
- p - puts the previously deleted text after the cursor.
- / - backward search n to find the next occurrence and N to search in opposite direction.
- ? - forward search.
- v - starts visual mode for selecting the lines and you can perform operation on that like d delete.

Appendix B: Explanation Of Packages

conda-forge: <https://anaconda.org/conda-forge>

Msys2: Software Distribution and Building Platform for Windows

<https://www.msys2.org/>

bioconda: **Bioconda** lets you install thousands of software packages related to biomedical research using the [conda](#) package manager.

<https://bioconda.github.io/>

menpo: The Menpo Project is a set of BSD licensed Python frameworks and associated tooling that provide end-to-end solutions for 2D and 3D deformable modeling.

<https://www.menpo.org/> <https://github.com/menpo/menpo>

pytorch: An open source machine learning framework that accelerates the path from research prototyping to production deployment.

<https://pytorch.org/>

jupyter: Free software, open standards, and web services for interactive computing across all programming languages

<https://jupyter.org/>

simpleitk: A simplified path to Insight. Open-source multi-dimensional image analysis in Python, R, Java, C#, Lua, Ruby, TCL and C++. Developed by the Insight Toolkit community for the biomedical sciences and beyond.

<https://simpleitk.org/>

nvidia: The inventor of the GPU(graphics processing unit)

Appendix C: Extra

1. 推荐IDE: PyCharm (PyCharm: the Python IDE for Professional Developers by JetBrains) edu邮箱免费使用
2. 推荐Windows命令行: Windows Terminal (microsoft/terminal: The new Windows Terminal and the original Windows console host, all in the same place! (github.com))
3. 推荐使用Windows子系统: WSL2 (Install WSL | Microsoft Learn)