

1. conda 命令工具

(1) 常用功能：-h 或 --help 用于显示帮助信息；-v 或 --verbose 可多次使用，用于控制输出详细程度；--no-plugins 用于禁用非内置插件；-V 或 --version 用于显示 conda 版本号。

(2) 主要命令：涵盖环境管理（如 activate 激活环境、deactivate 停用环境、create 创建环境）、程序管理（如 install 安装、remove 卸载、update 更新）、配置管理（config 修改 .condarc 中的配置值）、信息查看（info 显示安装信息、list 列出已安装包）等多种功能。

```
MINGW64~/c/Users/PC/repo x + v
usage: conda-script.py [-h] [-v] [--no-plugins] [-V] COMMAND ...

conda is a tool for managing and deploying applications, environments and packages.

options:
  -h, --help            Show this help message and exit.
  -v, --verbose          Can be used multiple times. Once for detailed output, twice for INFO logging, thrice for DEBUG
                        logging, four times for TRACE logging.
  --no-plugins          Disable all plugins that are not built into conda.
  -V, --version          Show the conda version number and exit.

commands:
  The following built-in and plugins subcommands are available.

COMMAND
activate      Activate a conda environment.
build         Build conda packages from a conda recipe.
clean         Remove unused packages and caches.
commands      List all available conda subcommands (including those from plugins). Generally only used by
              tab-completion.
compare       Compare packages between conda environments.
config        Modify configuration values in .condarc.
content-trust Signing and verification tools for Conda
convert       Convert pure Python packages to other platforms (a.k.a., subdirs).
create        Create a new conda environment from a list of specified packages.
deactivate    Deactivate the current active conda environment.
debug         Debug the build or test phases of conda recipes.
develop       Install a Python package in 'development mode'. Similar to 'pip install --editable'.
doctor        Display a health report for your environment.
export        Export a given environment
index         Update package index metadata files.
info          Display information about current conda install.
init          Initialize conda for shell interaction.
inspect       Tools for inspecting conda packages.
install       Install a list of packages into a specified conda environment.
list          List installed packages in a conda environment.
metapackage   Specialty tool for generating conda metapackage.
notices       Retrieve latest channel notifications.
pack          See 'conda pack --help'.
package       Create low-level conda packages. (EXPERIMENTAL)
remove (uninstall) Remove a list of packages from a specified conda environment.
rename        Rename an existing environment.
render        Expand a conda recipe into a platform-specific recipe.
repo          See 'conda repo --help'.
repoquery     Advanced search for repodata.
run           Run an executable in a conda environment.
search        Search for packages and display associated information using the MatchSpec format.
server        See 'conda server --help'.
skeleton      Generate boilerplate conda recipes.
token         See 'conda token --help'.
update (upgrade) Update conda packages to the latest compatible version.
```

2. python 与 python 哲学

```
MINGW64~/Users/PC/repo x + v
bash: conda1: command not found

PCDESKTOP-326PLNC MINGW64 ~
$ which conda
/c/Users/PC/anaconda3/Scripts/conda

PCDESKTOP-326PLNC MINGW64 ~
$ echo $PATH
/c/Users/PC/bin:/mingw64/bin:/usr/local/bin:/usr/bin:/bin:/mingw64/bin:/usr/bin:/c:/Users/PC/bin:/c:/Windows/system32:/c:/Windows:/c:/Windows/System32/Wbem:/c:/Windows/System32/WindowsPowerShell/v1.0:/c:/Windows/System32/OpenSSH:/c:/Program Files/NVIDIA Corporation/NVIDIA app/NVIDIA.LSR:/c:/Program Files (x86)/NVIDIA Corporation/PhysX/Common:/cmd:/mingw64/bin:/usr/bin:/c:/Users/PC/anaconda3:/c:/Users/PC/anaconda3/Library/mingw-w64/bin:/c:/Users/PC/anaconda3/Library/usr/bin:/c:/Users/PC/anaconda3/Library/bin:/c:/Users/PC/anaconda3/Scripts:/c:/Users/PC/AppData/Local/Microsoft/WindowsApps:/c:/Users/PC/AppData/Local/Programs/Microsoft VS Code/bin:/usr/bin/vendor_perl:/usr/bin/cpan_perl

PCDESKTOP-326PLNC MINGW64 ~
$ python
Python 3.12.7 | packaged by Anaconda, Inc. | (main, Oct 4 2024, 13:17:27) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> quit()

PCDESKTOP-326PLNC MINGW64 ~
$ which python
/c:/Users/PC/anaconda3/python

PCDESKTOP-326PLNC MINGW64 ~
$ python3
Python 3.12.7 | packaged by Anaconda, Inc. | (main, Oct 4 2024, 13:17:27) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import this
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
```

3. Python 代码的使用演示

```
MINGW64~/Users/PC/repo x + v
PCDESKTOP-326PLNC MINGW64 ~
$ ipython
Python 3.12.7 | packaged by Anaconda, Inc. | (main, Oct 4 2024, 13:17:27) [MSC v.1929 64 bit (AMD64)]
Type "copyright", "credits" or "license()" for more information.
IPython 8.27.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: print('hello')
hello

In [2]:
Do you really want to exit (Y/n)? y

PCDESKTOP-326PLNC MINGW64 ~
$ python
Python 3.12.7 | packaged by Anaconda, Inc. | (main, Oct 4 2024, 13:17:27) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print('hello')
hello
>>> quit()

PCDESKTOP-326PLNC MINGW64 ~
$ pwd
/c:/Users/PC

PCDESKTOP-326PLNC MINGW64 ~
$ mkdir repo

PCDESKTOP-326PLNC MINGW64 ~
$ ls
[开始]  菜单@      ntuser.dat.LOG2
'IO Objects'  NTUSER.DAT{ca6dd9f81-f9fb-11ef-ba89-7c10c9231f89}.tm.blf
'anaconda3'  NTUSER.DAT{ca6dd9f81-f9fb-11ef-ba89-7c10c9231f89}.TMCContainer00000000000000000001.regtrans-ms
AppData/     NTUSER.DAT{ca6dd9f81-f9fb-11ef-ba89-7c10c9231f89}.TMCContainer00000000000000000002.regtrans-ms
'Application Data'@  ntuser.ini
Contacts/       Onedrive/
Cookies@        Pictures/
Desktop/        PrintHood@
Documents/      Recent@
Downloads/      repo/
Favorites/      'Saved Games'/
Links/          Searches/
'Local Settings'@ SendTo@
Music/          Templates@
'My Documents'@  Videos/
Network@        'OneDrive cloud Files'/
NTUSER.DAT      WPSDrive/
ntuser.dat.LOG1

PCDESKTOP-326PLNC MINGW64 ~
$ cd repo

PCDESKTOP-326PLNC MINGW64 ~/repo
```

4. SSH 密钥生成与 Git 操作

(1) 生成 SSH 密钥：在 MinGW64 终端执行 `ssh-keygen -t ed25519 -C "15116912830@163.com"` 命令生成 ed25519 类型的密钥对，指定了保存路径，设置了密码（可空），并成功生成了私钥 `id_ed25519` 和公钥 `id_ed25519.pub`。

```
MINGW64~/Users/PC/repo x + -
PCDESKTOP-326PLMC MINGW64 ~/repo
$ ssh-keygen -t ed25519 -C "1511691283@163.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/c:/Users/PC/.ssh/id_ed25519):
Created directory /c:/Users/PC/.ssh/.
Enter passphrase for /c:/Users/PC/.ssh/id_ed25519 (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c:/Users/PC/.ssh/id_ed25519
Your public key has been saved in /c:/Users/PC/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:USkQM6x8c56Gs83AC8FouhUPazuJ7P0d0JHqk3YJhk 1511691283@163.com
The key's randomart image is:
+--[ED25519 256]--+
|
|o..
|o o
|+ + o
|+ E o + o
|o + + .
| . * o.
|o.X.o.
|==
|+--[SHA256]--+

PCDESKTOP-326PLMC MINGW64 ~/repo
$ cd
PCDESKTOP-326PLMC MINGW64 ~
$ pwd
/c:/Users/PC
PCDESKTOP-326PLMC MINGW64 ~
$ cd .ssh
PCDESKTOP-326PLMC MINGW64 ~/.ssh
$ ls
id_ed25519 id_ed25519.pub
PCDESKTOP-326PLMC MINGW64 ~/.ssh
$ cat id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZD1NTESAAAAIAK9Qzrly87VmaX0alsZtNmRP1FAu8sKieGzF4QIQ2 1511691283@163.com
PCDESKTOP-326PLMC MINGW64 ~/.ssh
$ ssh -T git@gitcode.com
The authenticity of host 'gitcode.com (116.205.2.91)' can't be established.
RSA key fingerprint is SHA256:aflsy+4a2RC7mWyy5eKlQk0tk8yv7Jd+XG04EXj1Y.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'gitcode.com' (RSA) to the list of known hosts.
remote: Welcome to GitCode, GJ2025
PCDESKTOP-326PLMC MINGW64 ~/.ssh
```

(2) 配置Git 账号信息：使用 `git config --global user.name GJ2025` 和 `git config --global user.email GJ2025@noreply.gitcode.com` 命令配置全局用户名和邮箱，之后通过 `git config --list --global` 命令查看配置结果。

```
MINGW64~/Users/PC/repo x + -
PCDESKTOP-326PLMC MINGW64 ~/.ssh
$ ssh -T git@gitcode.com
remote: Welcome to GitCode, GJ2025
PCDESKTOP-326PLMC MINGW64 ~/.ssh
$ git config --global user.name GJ2025
git config --global user.email GJ2025@noreply.gitcode.com
PCDESKTOP-326PLMC MINGW64 ~/.ssh
$ git config --list --global
core.editor=C:\Users\PC\AppData\Local\Programs\Microsoft VS Code\bin\code" --wait
user.name=GJ2025
user.email=GJ2025@noreply.gitcode.com
PCDESKTOP-326PLMC MINGW64 ~/.ssh
$ pwd
/c:/Users/PC/.ssh
PCDESKTOP-326PLMC MINGW64 ~/.ssh
$ cd
PCDESKTOP-326PLMC MINGW64 ~
$ pwd
/c:/Users/PC
PCDESKTOP-326PLMC MINGW64 ~
$ cd repo
PCDESKTOP-326PLMC MINGW64 ~/repo
$ pwd
/c:/Users/PC/repo
PCDESKTOP-326PLMC MINGW64 ~/repo
$ git clone git@gitcode.com:GJ2025/week01.git
Cloning into 'week01'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 7 (delta 0), reused 7 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (7/7), 8.75 KiB | 8.75 MiB/s, done.
PCDESKTOP-326PLMC MINGW64 ~/repo
$ ll
total 0
-rw-r--r-- 1 PC 197121 0 3月 15 17:48 script1.py
drwxr-xr-x 1 PC 197121 0 3月 15 18:25 week01/
PCDESKTOP-326PLMC MINGW64 ~/repo
$ code week01/
PCDESKTOP-326PLMC MINGW64 ~/repo
$
```

(3) 克隆Git 仓库：先切换到指定目录，然后执行 `git clone git@gitcode.com:GJ2025/week01.git` 命令，成功将远程仓库克隆到本地，克隆完成后在本地目录中查看相关文件，并使用 `code week01/` 命令在VS Code 中打开克隆的项目。

