**Bài tập thực hành 01:**

Cài đặt và làm quen với Anaconda-Jupyter

# **Mục tiêu cần đạt**:

Anaconda, jupyter là môi trường hỗ trợ quan trọng cho các bạn làm quen với lập trình python, trong buổi này các bạn cần

* Biết cách cài đặt Anaconda
* Biết cách cài đặt Jupyter
* Có thể triển khai chạy Jupyter trên colab
* Thực hiện trả lời các câu hỏi từ đó giúp các bạn nắm được các kỹ thuật thực hành kết hợp lý thuyết

Vì các phần cài đặt có thể thay đổi theo thời gian nên trong hướng dẫn sẽ trình bày các link để các bạn có thể truy cập nguồn tài liệu mới nhất

**Các link tham chiếu/gợi ý**

## Cách cài đặt Anaconda: các bạn tham khảo ở: <https://docs.anaconda.com/anaconda/install/index.html>

## Tạo biến môi trường conda/anaconda để cmd/terminal nhận dạng được lệnh conda

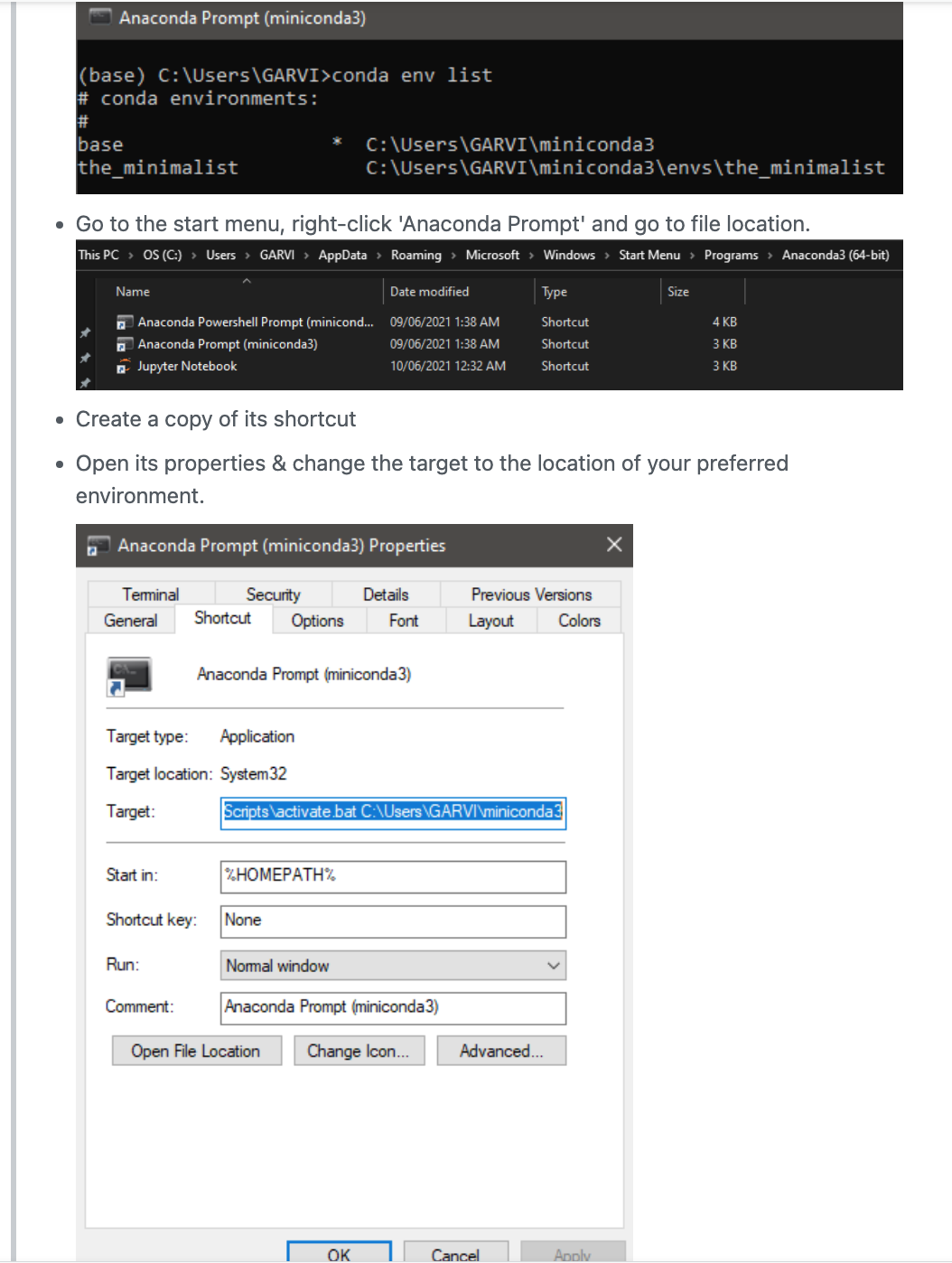
<https://www.youtube.com/watch?v=YZZg_w8atx0>

## Cách tạo/quản lý/kích hoạt môi trường trong Anaconda: <https://conda.io/docs/user-guide/tasks/manage-environments.html>

<https://www.youtube.com/watch?v=ErXtLCxsknE>

## Kích hoạt môi trường khi mở cmd/terminal:

<https://stackoverflow.com/questions/43081098/activate-anaconda-python-environment-persistently-in-windows>



## Cách cài đặt các thư viện trong môi trường anaconda:

conda install <tên thư viện>

## Cách sử dụng colab:

<https://colab.research.google.com/>

Tài liệu chi tiết Jupyter Notebook

<https://jupyter-notebook.readthedocs.io/en/stable/examples/Notebook/examples_index.html>

# Yêu cầu bài thực hành:

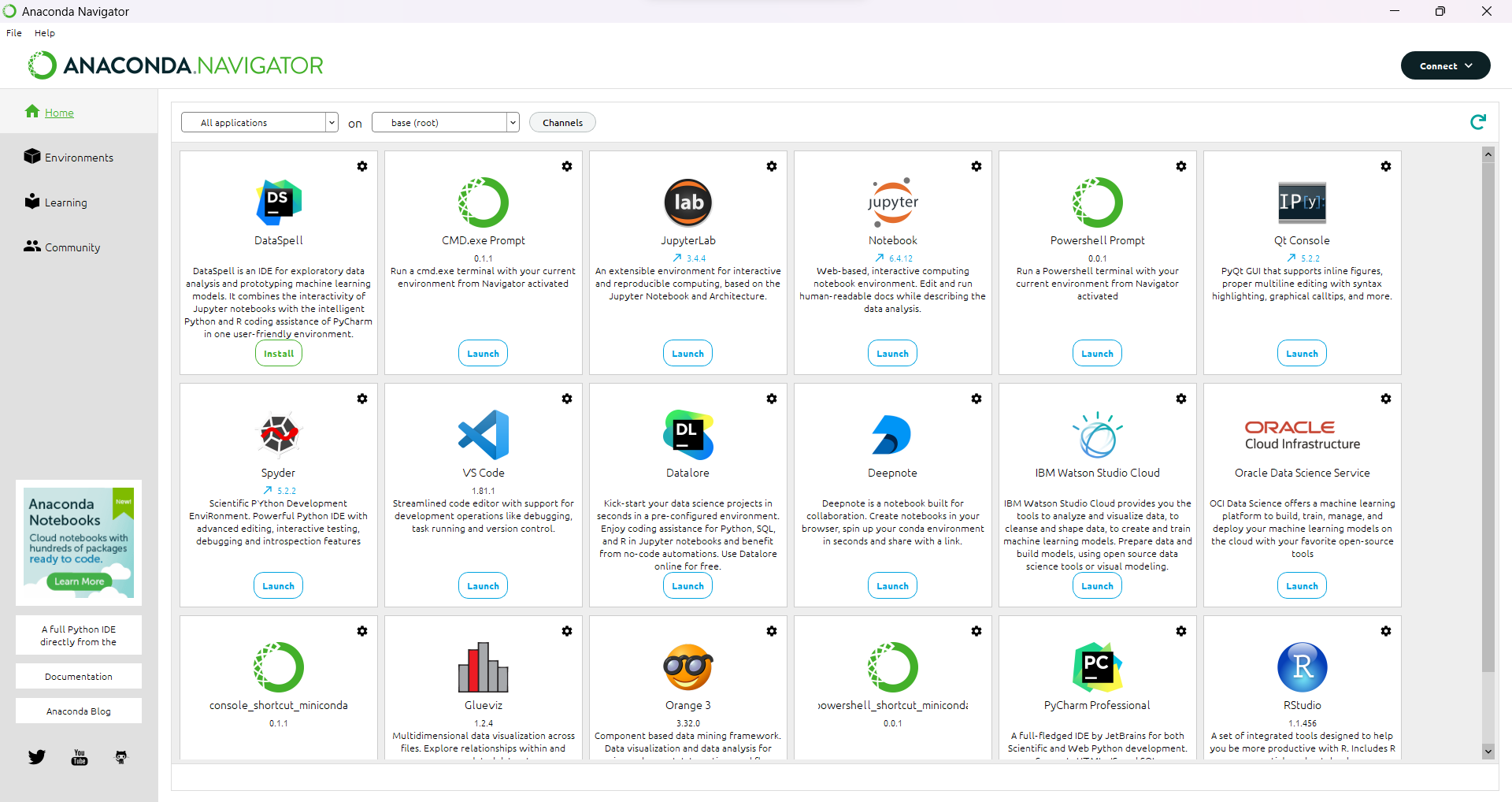
* *Với file word: Với các câu hỏi về lệnh các bạn trả lời bằng text: copy lệnh (dạng text) vào bài word trả lời và kèm hình ảnh minh họa. Tất cả câu trả lời phải có hình minh họa minh chứng kết quả*

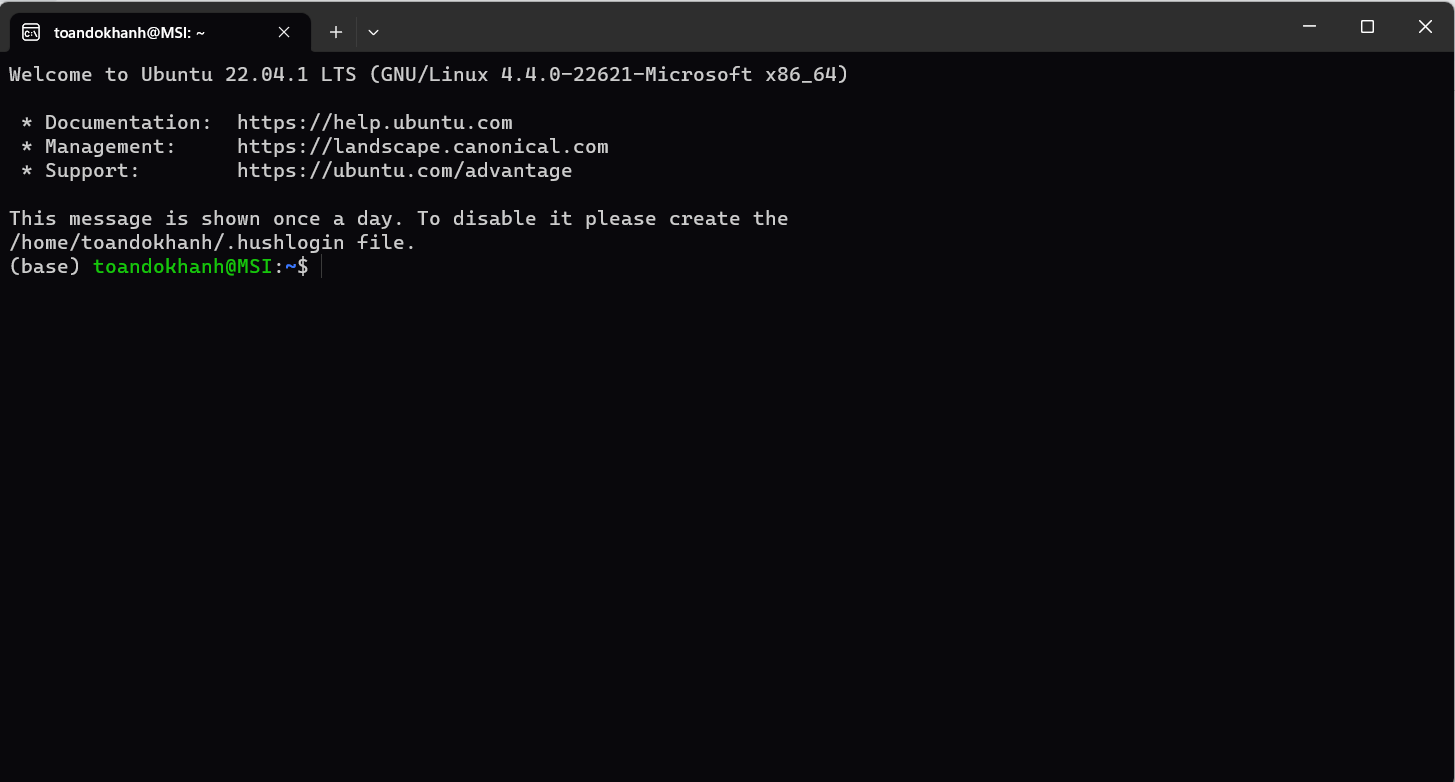
1. Anaconda là gì? Cho biết lịch sử ra đời, và phiên bản tính đến hiện tại.

- Anaconda là nền tảng mã nguồn mở về Khoa học dữ liệu trên Python thông dụng nhất hiện nay. Anaconda Với hơn 11 triệu người dùng, Anaconda là cách nhanh nhất và dễ nhất để học Khoa học dữ liệu với Python hoặc R trên Windows, Linux và Mac OS X.

- Phiên bản mới nhất tính đến hiện tại là: 23.2.0

1. Các bạn hãy cài đặt Anaconda và chụp lại màn hình quá trình cài đặt.



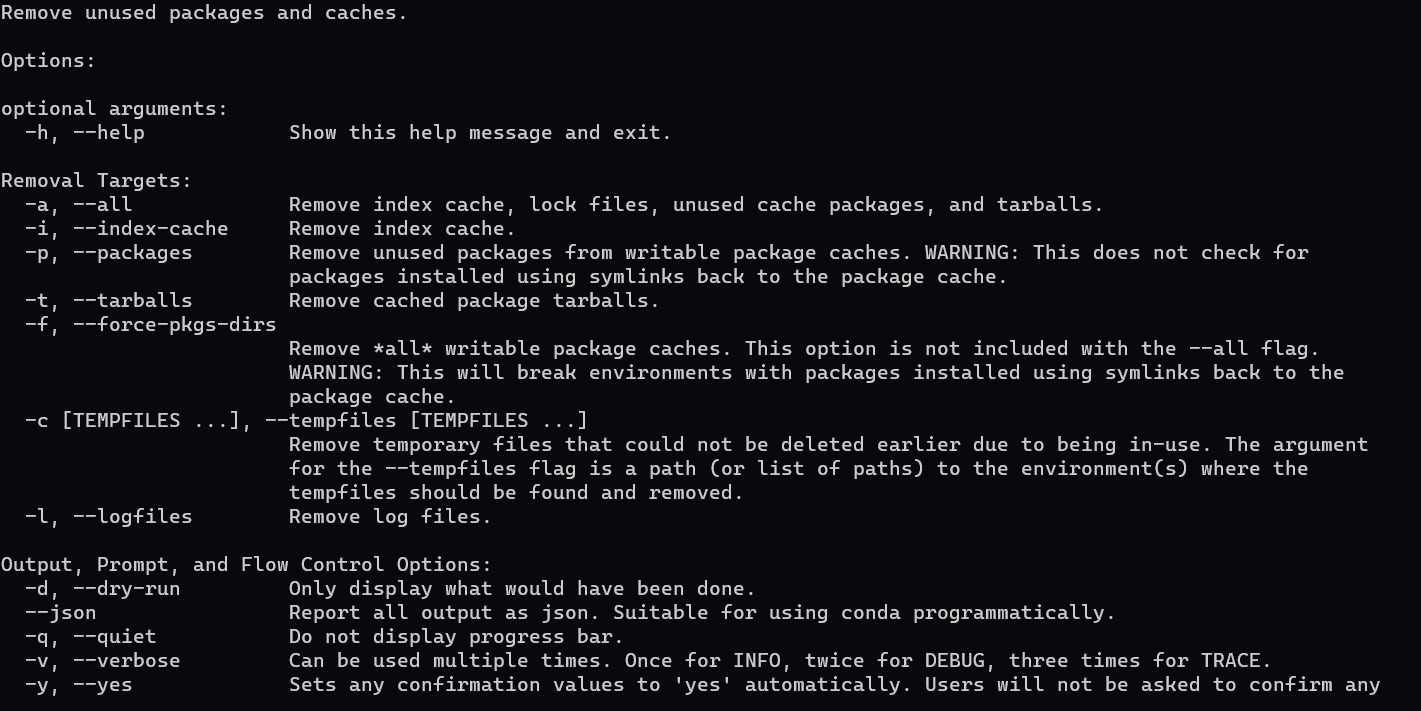


(Phần này em đã cài ở học kỳ rồi nên mong Thầy cho em bỏ qua quá trình cài đặt)

1. Cho biết mục đích/cách sử dụng các lệnh dưới đây. Mỗi lệnh các bạn chạy minh họa 1 ví dụ và chụp lại màn hình thực hiện



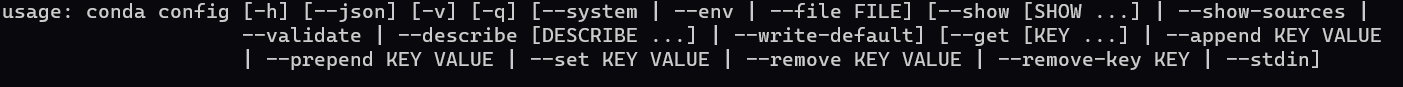
**Conda clean :** Xóa các gói và bộ đệm không sử dụng.



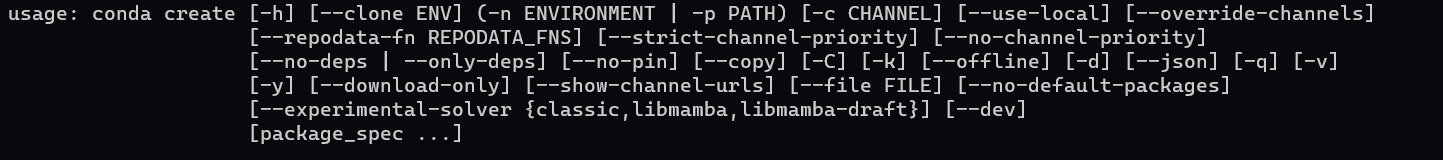
**Conda compare :** so sánh các gói giữa các môi trường conda



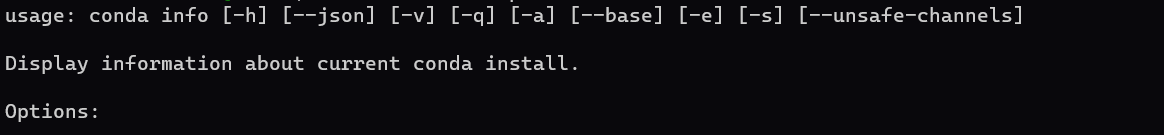
**Conda config:** Sửa đổi giá trị cấu hình trong .condarc. Điều này được mô hình hóa sau lệnh git config. Ghi vào tệp .condarc của người dùng (/home/docs/.condarc) theo mặc định. Sử dụng cờ --show-sources để hiển thị tất cả các vị trí cấu hình đã xác định trên máy tính của bạn.



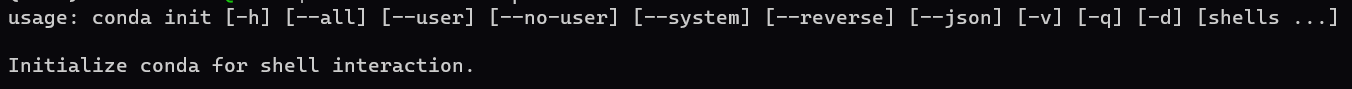
**Conda create**: Tạo môi trường conda mới từ danh sách các gói được chỉ định. Để sử dụng môi trường mới tạo, hãy sử dụng 'conda activate envname'. Lệnh này yêu cầu tùy chọn -n NAME hoặc -p PREFIX.



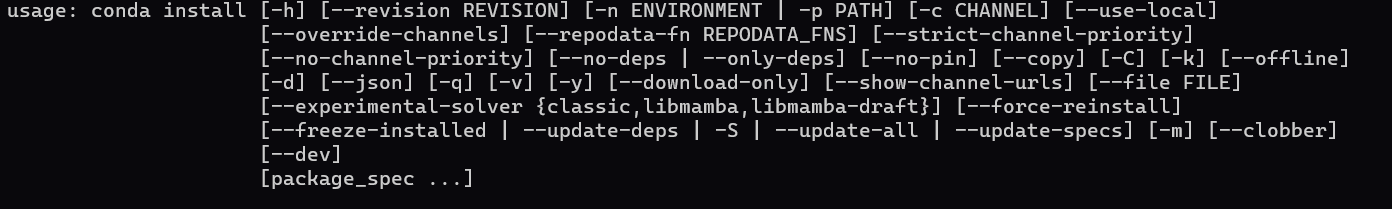
- **Conda info:** Hiển thị thông tin về cài đặt conda hiện tại



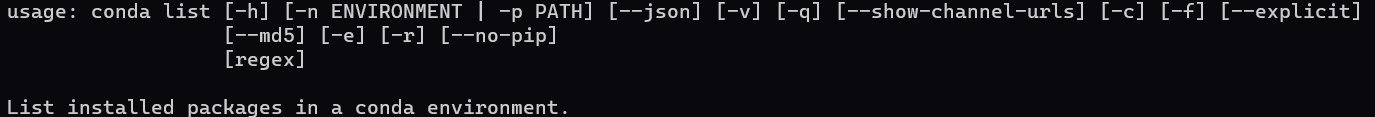
- **Conda init:** Khởi tạo conda cho tương tác shell.



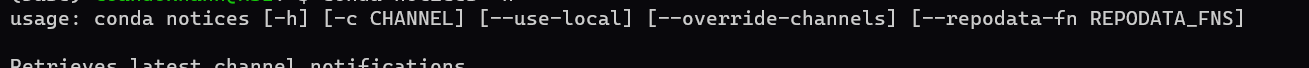
* **Conda install:** Cài đặt danh sách các gói vào một môi trường conda được chỉ định.



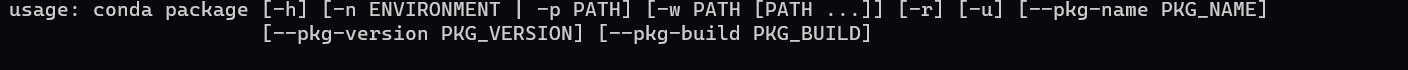
- **Conda list**: Liệt kê các gói đã cài đặt trong môi trường conda.



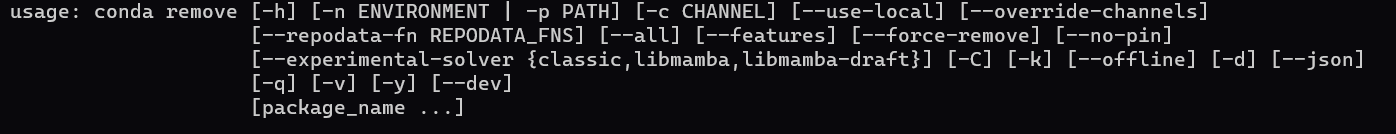
- **Conda notices**: Truy xuất thông báo kênh mới nhất.



- **Conda package**: cài đặt packages vào một môi trường có sẵn giúp dễ dàng quản lý, kiểm soát các packages.



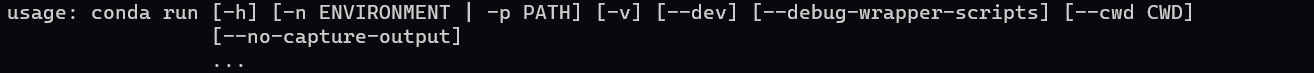
- **Conda remove:** Xóa danh sách các gói khỏi môi trường điều kiện đã chỉ định.



-  **Conda rename:** Đổi tên một môi trường hiện có.



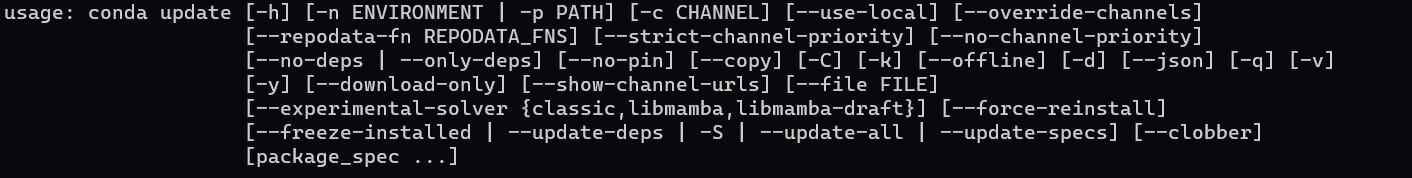
**- Conda run:** Chạy một tệp thực thi trong môi trường conda



- **Conda search**: Tìm kiếm các gói và hiển thị thông tin liên quan. Đầu vào là MatchSpec, ngôn ngữ truy vấn cho các gói conda.



- **Conda update**: Cập nhật các gói conda lên phiên bản tương thích mới nhất.



Link tham khảo:<https://docs.conda.io/>

1. Môi trường Anaconda[[1]](#footnote-1) là gì? Tại sao nên tạo môi trường trong Anaconda?

- Môi trường Anaconda là nơi bạn có thể tạo, xuất, liệt kê, xóa và cập nhật các môi trường có các phiên bản Python và/hoặc gói khác nhau được cài đặt trong đó. Chuyển đổi hoặc di chuyển giữa các môi trường được gọi là kích hoạt môi trường. Bạn cũng có thể chia sẻ tệp môi trường.

- Nên tạo môi trường trong Anaconda vì:

- một trong những phụ thuộc cốt lõi của bạn vừa phát hành một phiên bản mới (cập nhật số phiên bản phụ thuộc).

- bạn cần một gói bổ sung để phân tích dữ liệu (thêm một phụ thuộc mới).

- bạn đã tìm thấy gói tốt hơn và không cần gói cũ hơn nữa (thêm phần phụ thuộc mới và xóa phần phụ thuộc cũ).

1. Môi trường base trong anaconda là gì? Trong môi trường base có chứa sẵn những module nào. Hãy tìm hiểu tùy chỉnh cấu hình để khi bật terminal hoặc cmd thì tự động bật sẵn môi trường base[[2]](#footnote-2), ví dụ:

Trong Anaconda, "base" là một environment mặc định và cơ sở của hệ thống. Nó bao gồm một số phần mềm cơ bản và thư viện Python để giúp bạn bắt đầu dễ dàng. Bạn có thể tạo ra nhiều môi trường riêng biệt với các phiên bản Python hoặc các thư viện khác nhau để chạy các dự án của mình một cách độc lập. Điều này giúp bạn tránh xung đột giữa các dự án và giải quyết vấn đề phiên bản.

- Môi trường base trong Anaconda chứa một số phần mềm và thư viện Python cơ bản như sau:

- Python: Là một ngôn ngữ lập trình chính được sử dụng trong môi trường base.

- NumPy: Là một thư viện tính toán số học cho Python.

- Matplotlib: Là một thư viện vẽ đồ thị cho Python.

- Pandas: Là một thư viện xử lý dữ liệu cho Python.

- Jupyter: Là một nền tảng cho việc tạo và chia sẻ tài liệu kết hợp văn bản, mã và đồ thị.

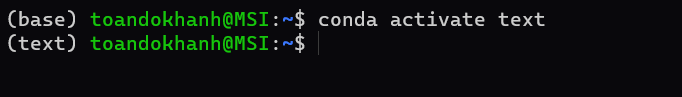
- IPython: Là một shell Python nâng cao với các tính năng tự động hoàn thành, gợi ý lệnh và tự động từ điển.

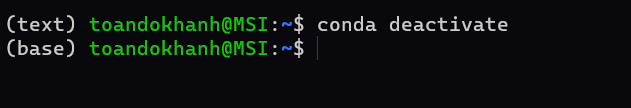
- SciPy: Là một thư viện tính toán khoa học cho Python.

- Scikit-learn: Là một thư viện machine learning cho Python.

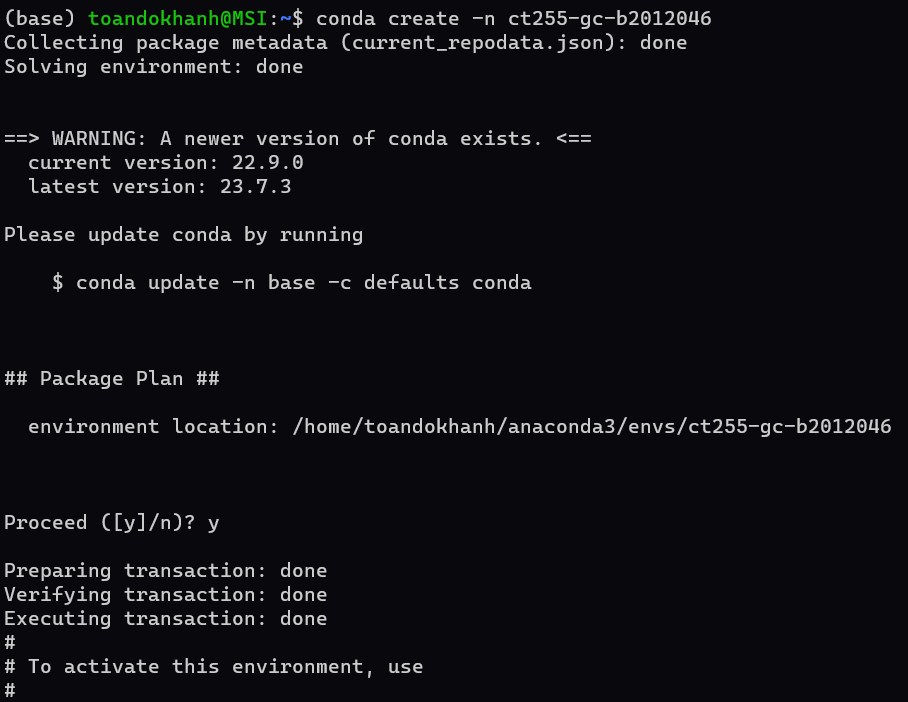


Các lệnh nào để activate và deactive (deactivate) môi trường trong anaconda, chạy minh họa.

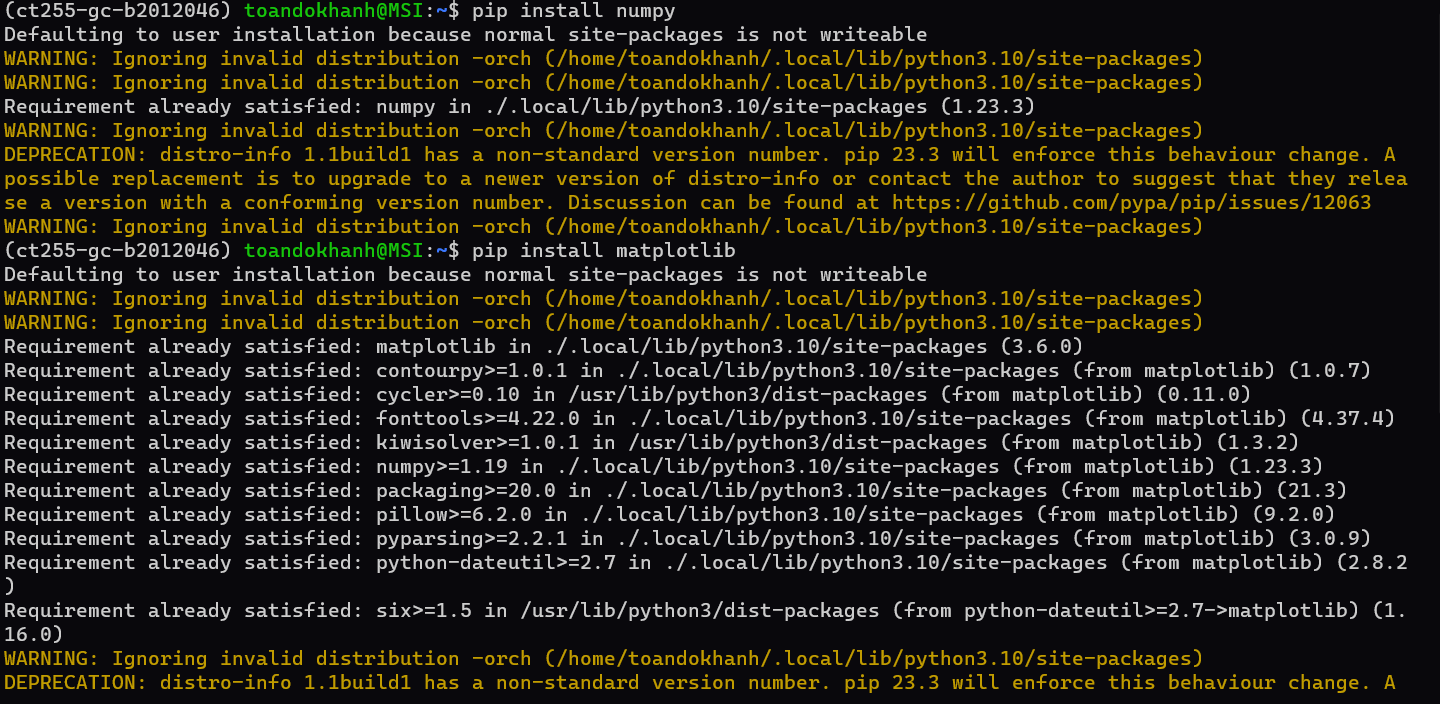




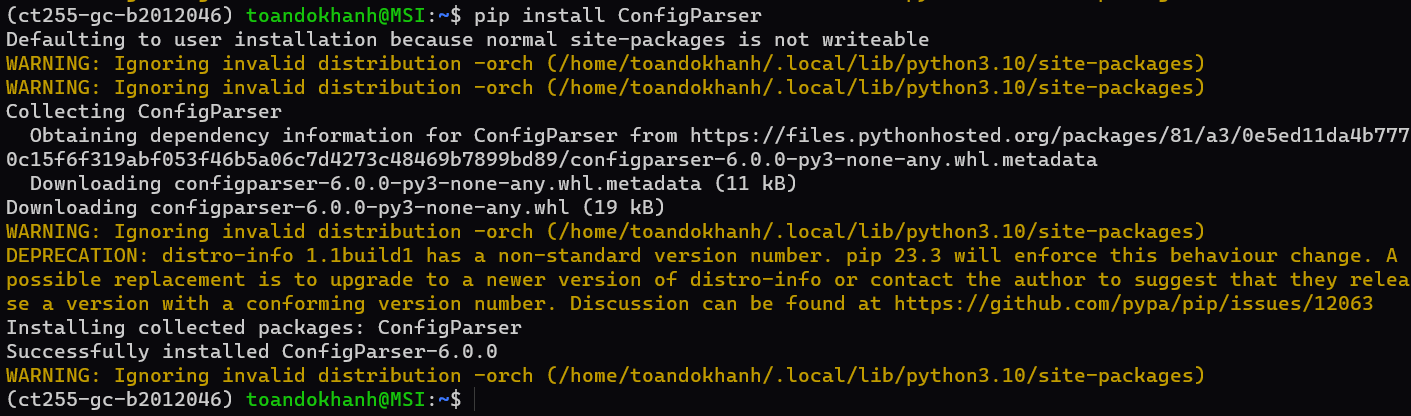
1. Hãy tạo một môi trường tên là ct255-btnhom-<mssv của bạn>. Cài đặt các gói sau vào môi trường đã tạo (cài đặt pip nếu chưa có[[3]](#footnote-3)):



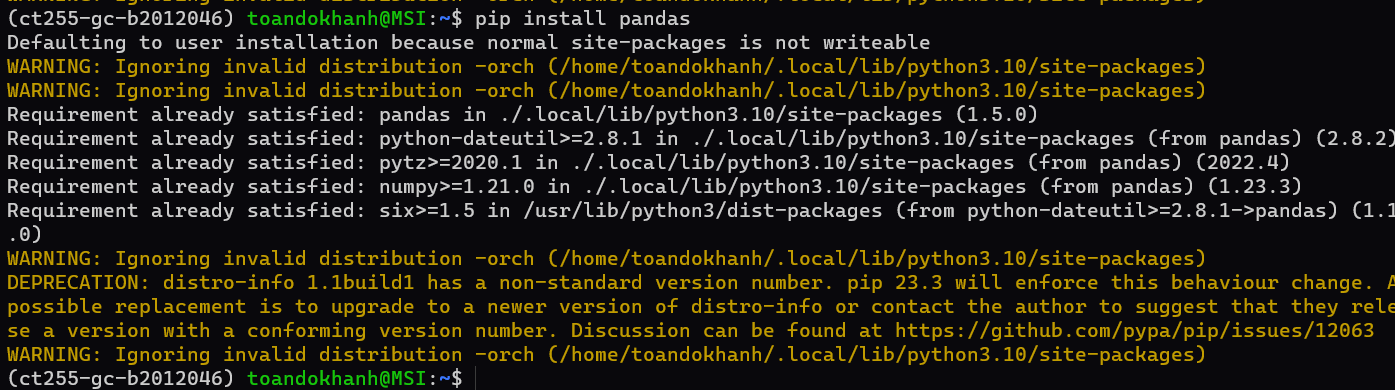
* pip install numpy
* pip install matplotlib



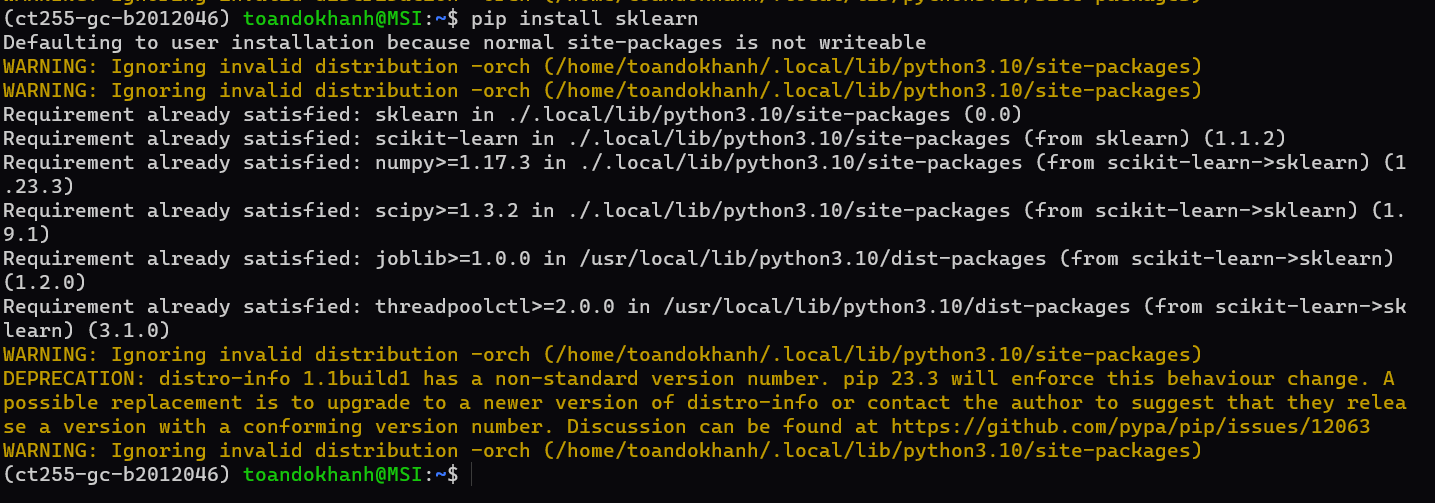
* pip install ConfigParser



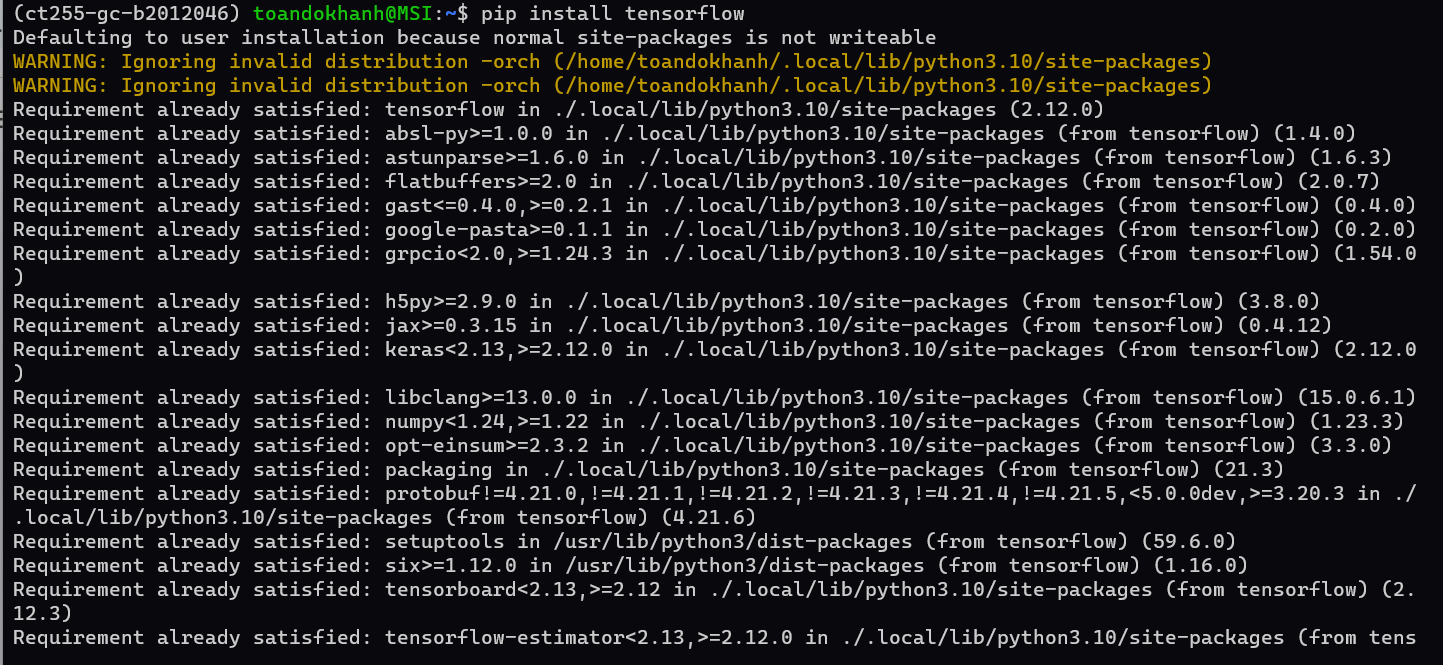
* pip install pandas



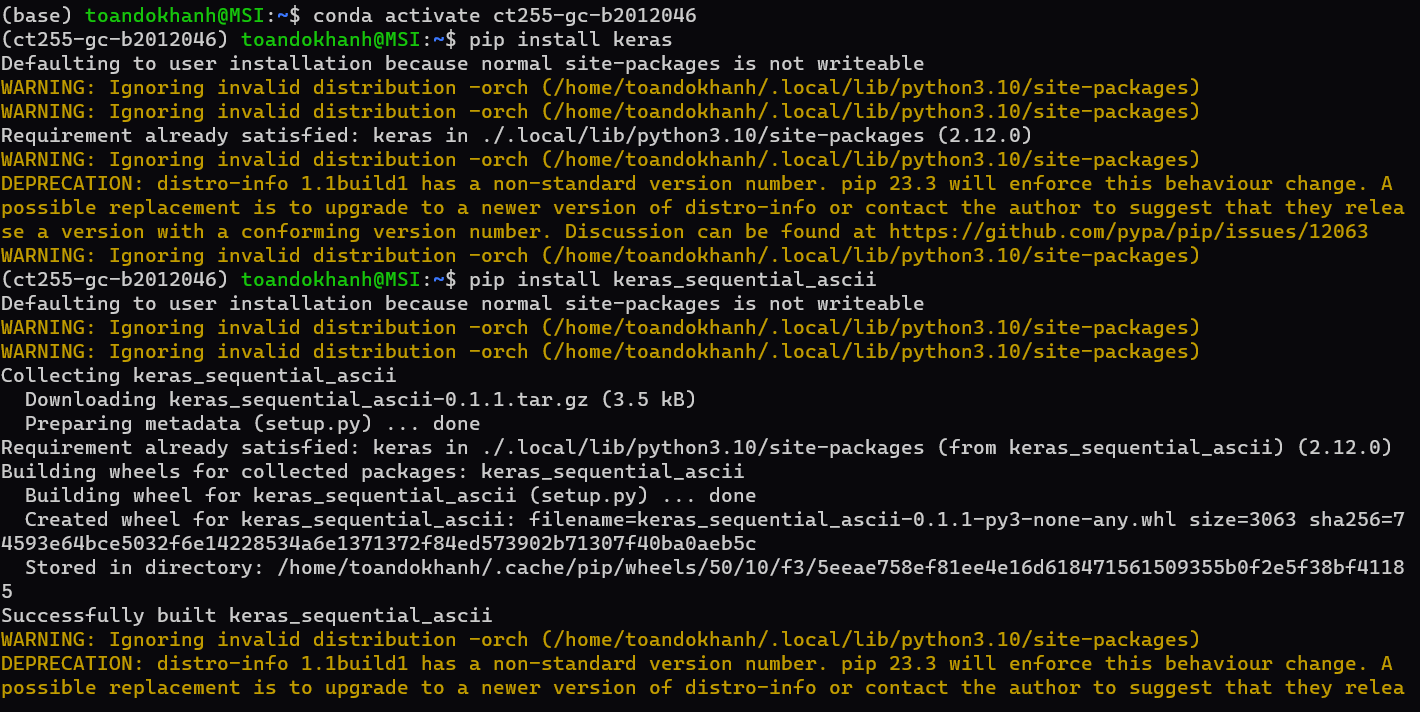
* pip install sklearn



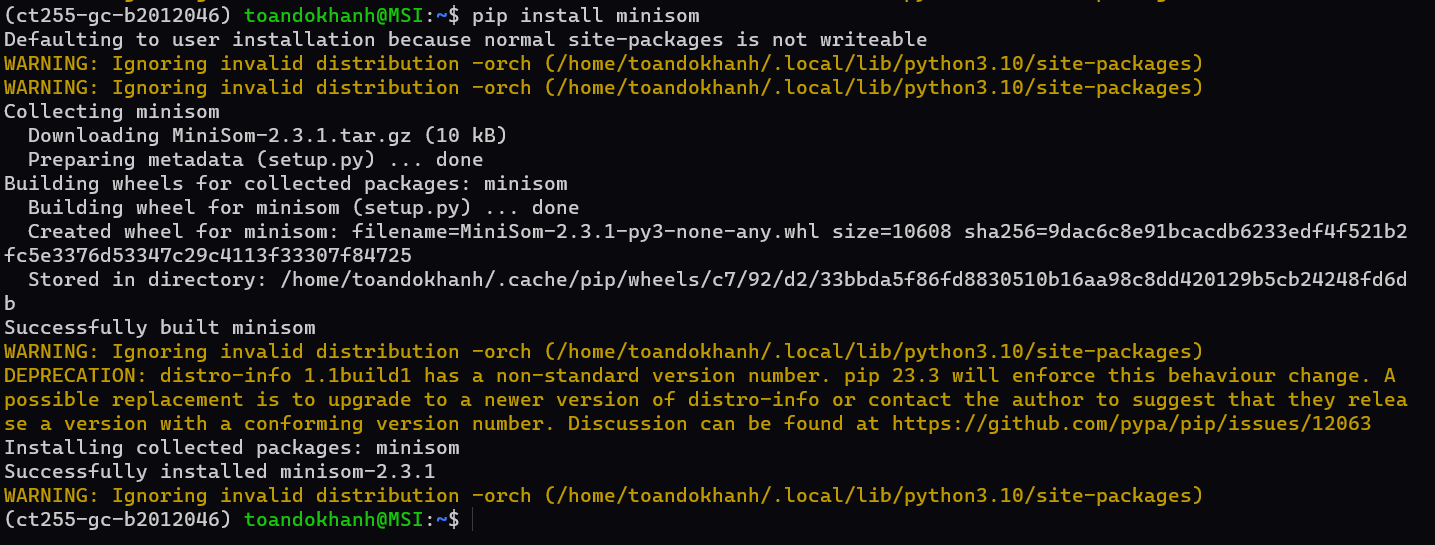
* pip install tensorflow



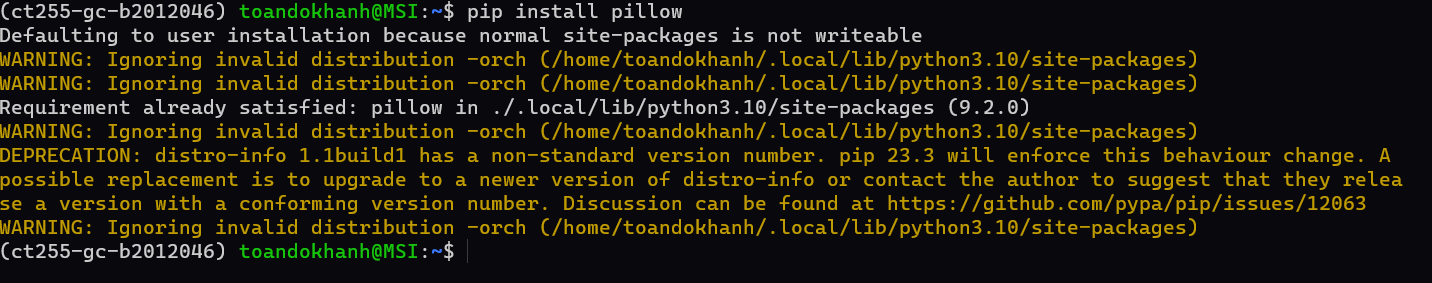
* pip install keras
* pip install keras\_sequential\_ascii



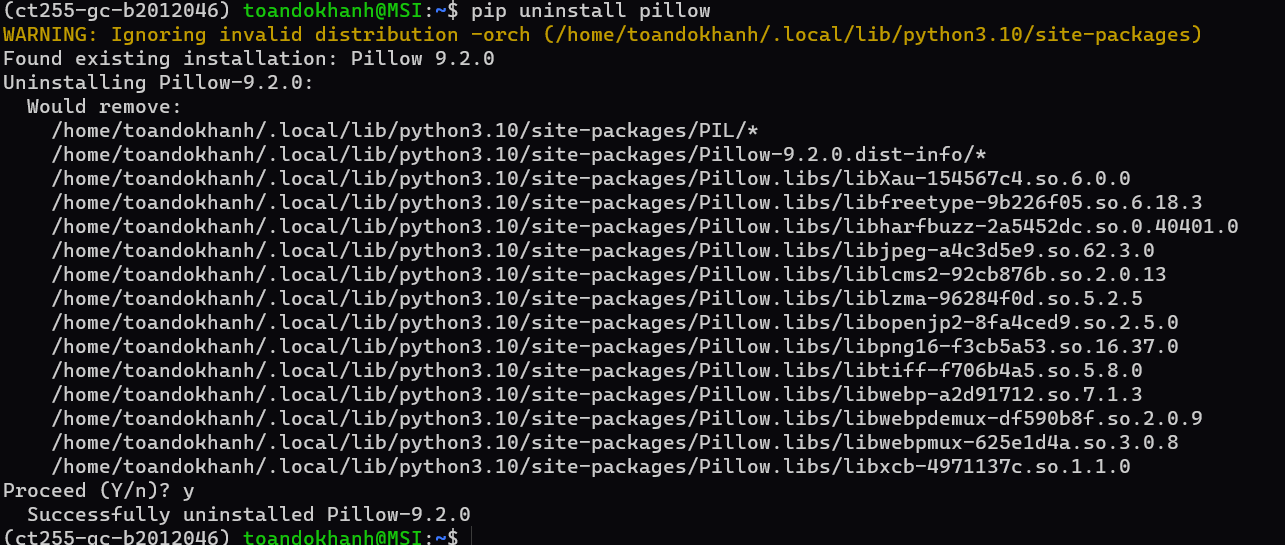
* pip install minisom



* pip install pillow



Giờ thử dùng lệnh pip uninstall <tên gói> để gỡ cài đặt và hãy thay thế bằng lệnh conda install <tên gói>, quan sát kết quả có gì khác nhau.



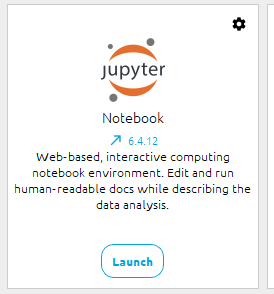
1. Jupyter là gì? Cho biết lịch sử ra đời, và phiên bản tính đến hiện tại. Hiện đã hỗ trợ bao nhiêu ngôn ngữ lập trình?

- Jupyter là một nền tảng tính toán khoa học mã nguồn mở, với khả năng nổi bật cho phép tương tác trực tiếp với từng dòng code (interactive), hỗ trợ **hơn 40 ngôn ngữ lập trình**, trong đó tập trung vào 3 ngôn ngữ là **Ju**lia, **Py**thon và \*\*R, \*\*cái tên Jupyter bắt nguồn từ cách chơi chữ I Python, You R, We Julia, hay một công thức ngắn gọn Jupyter = Julia + Python + R. Bên cạnh đó, Jupyter cũng là một công cụ hoàn toàn miễn phí, được tạo ra với mục đích nhắm đến khoa học dữ liệu và giáo dục, giúp mọi người cùng học lập trình dễ dàng hơn. (cụ thể ở đây là Python). Jupyter có tính tương tác nên có thể sử dụng làm môi trường chạy thử và giảng dạy.

- Notebook vốn có tên là IPython Notebook, đến năm 2014 tách ra khỏi IPython và đổi tên thành Jupyter Notebook.

- Jupyter phiên bản mới nhất là : Jupyter 7.0.2

1. Hãy cài đặt Jupyter và chụp lại quá trình cài đặt. Mô tả rõ các thành phần trong Jupyter[[4]](#footnote-4),[[5]](#footnote-5). Sau khi cài đặt xong, hãy trình bày cách để chạy Jupyter từ một môi trường anaconda từ dòng lệnh (cmd hoặc terminal)



- Các thành phần trong Jupyter:

● Notebook interface: Là giao diện trực quan cho phép bạn viết và chạy mã, và tạo các tài liệu đồ họa và văn bản.

● Code cells: Là những ô chứa mã mà bạn có thể viết và chạy. Bạn có thể viết bất kỳ ngôn ngữ lập trình nào hỗ trợ bởi Jupyter, bao gồm Python, R, Julia và hơn thế nữa.

● Markdown cells: Là những ô chứa văn bản dùng để tạo ra các phần mô tả, ghi chú và tiêu đề. Bạn có thể sử dụng Markdown để tạo ra các bảng, danh sách hoặc cấu trúc văn bản khác.

● Kernels: Là các thành phần back-end cho phép viết và chạy mã trong các ô code. Mỗi notebook có một kernel riêng biệt, và bạn có thể chuyển đổi giữa các kernel để sử dụng các ngôn ngữ khác nhau.

● Interactive widgets: Là các thành phần giao diện người dùng giúp bạn tương tác với các biểu đồ, sliders, buttons và các điều khiển khác.

- Cách để chạy Jupyter từ một môi trường anaconda từ dòng lệnh (cmd hoặc terminal)

● Mở terminal hoặc Command Prompt.

● Chuyển đến thư mục mà bạn muốn lưu các tập tin Jupyter của mình. Bạn có thể sử dụng lệnh cd để thực hiện việc này.

1. Colab[[6]](#footnote-6) là gì? Hãy cho biết những tiện ích và thuận tiện của colab.

+Không yêu cầu cấu hình

+ Truy cập vào GPU miễn phí, chia sẻ dễ dàng. Cho dù bạn là sinh viên, nhà khoa học dữ liệu hay nhà nghiên cứu AI, Colab đều có thể giúp công việc của bạn trở nên dễ dàng hơn.

- Những tiện ích của sổ tay colab:

- Sổ tay Colab cho phép bạn kết hợp mã thực thi và văn bản đa dạng thức trong một tài liệu duy nhất, cùng với hình ảnh, HTML, LaTeX, v.v. Khi bạn tạo sổ tay Colab của riêng mình, chúng sẽ được lưu trữ trong tài khoản Google Drive của bạn. Bạn có thể dễ dàng chia sẻ sổ ghi chép Colab của mình với đồng nghiệp hoặc bạn bè, cho phép họ nhận xét về sổ ghi chép của bạn hoặc thậm chí chỉnh Bạn có thể nhập dữ liệu của riêng mình vào sổ ghi chép Colab từ tài khoản Google Drive, kể cả từ bảng tính, cũng như từ Github và nhiều nguồn khác và sửa chúng.

- Với Colab, bạn có thể khai thác toàn bộ sức mạnh của các thư viện Python phổ biến để phân tích và trực quan hóa dữ liệu.

1. Graphics Processing Unit (**GPU)** và Compute Unified Device Architecture (CUDA) là gì và dùng để làm gì? Những tiện ích của chúng?

Graphics Processing Unit (GPU) là một loại bộ xử lý chuyên dụng cho việc xử lý đồ hoạ và tính toán song song. Nó được thiết kế để thực hiện các nhiệm vụ liên quan đến đồ họa và tính toán số học một cách hiệu quả và nhanh chóng. GPU thường được sử dụng trong các ứng dụng đòi hỏi xử lý đồ họa cao như trò chơi, thiết kế đồ họa, video và cả trong các lĩnh vực khác như khoa học, tài chính và trí tuệ nhân tạo.

Compute Unified Device Architecture (CUDA) là một mô hình lập trình và một nền tảng tính toán song song phát triển bởi NVIDIA dành cho GPU của họ. CUDA cho phép các nhà phát triển sử dụng ngôn ngữ lập trình như C/C++ để thực hiện tính toán song song trên GPU. Điều này giúp tận dụng khả năng xử lý song song mạnh mẽ của GPU để thực hiện các tác vụ tính toán phức tạp nhanh hơn so với việc sử dụng CPU.

Các tiện ích của GPU và CUDA:

Xử lý đồ họa: GPU được thiết kế để xử lý các tác vụ liên quan đến đồ họa như vẽ, render, xử lý hình ảnh và video. Điều này giúp cải thiện hiệu suất và chất lượng hình ảnh trong các ứng dụng đồ họa.

Tính toán song song: GPU có hàng ngàn lõi xử lý có thể thực hiện các phép tính đồng thời trên nhiều dữ liệu khác nhau. Điều này giúp tăng tốc đáng kể cho các tác vụ tính toán phức tạp như mô phỏng khoa học, tính toán tài chính, trí tuệ nhân tạo và machine learning.

CUDA: CUDA cho phép nhà phát triển tận dụng sức mạnh của GPU để thực hiện tính toán song song thông qua ngôn ngữ lập trình quen thuộc như C/C++. Điều này giúp đơn giản hóa việc phát triển ứng dụng có khả năng tận dụng GPU mà không cần kiến thức chuyên sâu về lập trình GPU.

Tăng hiệu suất: Sử dụng GPU để thực hiện các phép tính đồng thời giúp tăng hiệu suất và tốc độ thực hiện các tác vụ so với việc sử dụng chỉ CPU.

Ứng dụng đa dạng: GPU và CUDA có thể được áp dụng trong nhiều lĩnh vực khác nhau như đồ họa, video, khoa học, tài chính, trí tuệ nhân tạo, machine learning, thăm dò dầu khí và nhiều lĩnh vực khác.

Link tham khảo: [GPU là gì? Chức năng và cách phân biệt giữa GPU và CPU (bizflycloud.vn)](https://bizflycloud.vn/tin-tuc/gpu-la-gi-20220510113330803.htm)

**Chú ý:**

* Các bạn nộp file word: Quy tắc đặt tên file: **<mssv>-<hoten>-<bai><stt\_bai thực hành>.docx** nộp lên Classroom (VD: **B123456-NguyenVanA-bai1.docx**), kèm với các file khác được yêu cầu như phần câu hỏi đã nêu. **Ngoại trừ file word trả lời câu hỏi, các file còn lại các bạn nén vào 1 file zip**. File zip đặt tên như file word.
* Mỗi câu các bạn trả lời bằng hình hoặc dạng text tùy vào yêu cầu của câu hỏi và **TRẢ LỜI THEO ĐÚNG THỨ TỰ CÂU HỎI**. Nếu câu nào không trả lời được các bạn cứ để số thứ tự câu hỏi và bỏ trống phần trả lời.
* Các câu trả lời có tham khảo trên Internet phải trích dẫn link/nguồn.
* *Vi phạm 1 trong các điều sau đây bài thực hành sẽ bị 0 điểm:*
  + Đặt tên KHÔNG ĐÚNG quy tắc được yêu cầu.
  + Bài không đủ các thành phần (word, code+data (nếu có),...) đã được yêu cầu.
  + Bài không thực hiện đúng yêu cầu “**Ngoại trừ file word trả lời câu hỏi, các file còn lại các bạn nén vào 1 file .zip”**
  + Bị phát hiện copy, sao chép từ các bạn khác
  + Phần trả lời không ghi rõ trả lời cho câu nào
  + Thứ tự câu trả lời không đúng thứ tự câu hỏi

1. https://conda.io/docs/user-guide/tasks/manage-environments.html [↑](#footnote-ref-1)
2. https://stackoverflow.com/questions/43081098/activate-anaconda-python-environment-persistently-in-windows [↑](#footnote-ref-2)
3. https://pip.pypa.io/en/stable/installation/ [↑](#footnote-ref-3)
4. https://codelearn.io/sharing/jupyter-notebook-tutorial [↑](#footnote-ref-4)
5. https://jupyter-notebook.readthedocs.io/en/stable/ui\_components.html#notebook-dashboard [↑](#footnote-ref-5)
6. https://colab.research.google.com/ [↑](#footnote-ref-6)