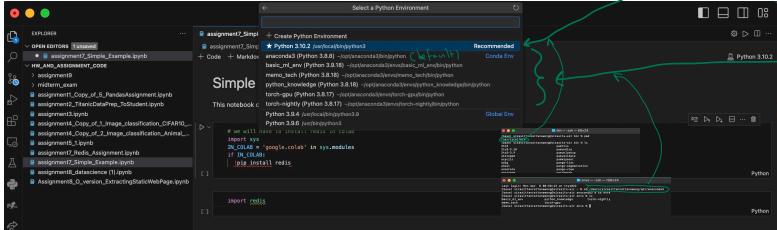


still waiting for update Nov at programming | ENV And Path

- file path import writing including .yaml file opennhoispt
and --init--.py
- when deploy you can use requirements.txt but you need to
specify python version

Part 1

There are 2 main types of using python



六四

（输入法消息处理）（在 Python 中实现）

- 1. python env. /usr/local/bin (run from default)

2. **anacoda env** (used in data science project)

/Users/sirasittanrattanawong/opt/anaconda3/bin

(you need to install `macosx` first)

ପ୍ରମାଣିତ ହିଁଛି !!

What is anaconda env

Sources

Getting started with Python environments...  towardsdatascience... • 1

Conda create environment and...  machinelearning... • 2

Managing environments - Anaconda...  docs.anaconda... • 3

  View 2 more

Answer

Anaconda Env is a feature provided by the Anaconda distribution of Python, which allows users to create and manage separate environments containing specific versions of packages and dependencies     . This feature is useful for managing multiple projects with different requirements, as it enables users to create isolated environments for each project, preventing conflicts between package versions and dependencies.

- For both /usr/local/bin and python at /Users/sirasittanattanawong/opt/anaconda3/bin

What would happen after we use pip install will the package lost when we close the program?

ChatGPT

When you use `'pip install'` to install a package while a specific Python environment is activated (like an Anaconda environment or a system-wide Python), the package is installed in that environment's `'site-packages'` directory and **remains installed until you explicitly uninstall it**. The package does not get lost when you close the program or terminal. Instead, it remains accessible to any Python script or program executed **within that environment**. So, if you install a package using `'pip'` while your Anaconda environment is active, it will be installed in `'/Users/sirasittanattanawong/opt/anaconda3/lib/pythonX.Y/site-packages'` (where X.Y is the Python version), not in the system-wide `'site-packages'` directory at `'/usr/local/bin'`.

you can create env for each project

Conda environment command

1. Create conda create --name myenv python=3.9
2. Activate You can activate your env by using conda activate <env name>
3. Show all env conda env list
4. Deactivate conda deactivate

The screenshot shows a Jupyter Notebook interface with two code cells. The first cell displays environment information:

```
import tensorflow as tf
print(tf.__version__)
print(tensorflow.__version__)

2.4.1
2.4.1
```

The second cell contains Python code related to tensor creation:

```
tensor1 = tf.constant([1, 2])
tensor2 = tf.constant([3, 4])
tensor3 = tf.add(tensor1, tensor2)
print(tensor3)

print(type(tensor1))
print(type(tensor2))
print(type(tensor3))

Dimensions of tensor 1: 1x2
Dimensions of tensor 2: 1x2
Dimensions of tensor 3: 1x2
```

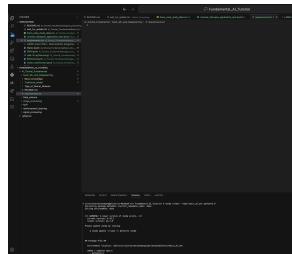
Output from the second cell shows:

```
Dimensions of tensor 1: 1x2
Dimensions of tensor 2: 1x2
Dimensions of tensor 3: 1x2
```

សូមរាយនៃ terminal នៃ conda env ឱ្យ

```
Last login: Mon Apr  8 10:08:55 on ttys021
(base) sirasittanrattanawong@sirasits-air ~ % python --version
Python 3.8.8
(base) sirasittanrattanawong@sirasits-air ~ % conda deactivate
(sirasittanrattanawong@sirasits-air ~ % python3 --version
Python 3.10.2
```

រាយការណ៍វិវាទ project



(skip this part if you have env)

- When you create a new project that need to be deployed you need to create a new environment

```
conda create --name myenv python=3.9
```

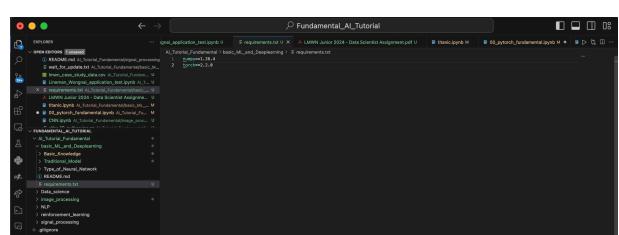
```
sirasittanrattanawong@Sirasits-MacBook-Air Fundamental_AI_Tutorial % conda activate basic_ml_env
(basic_ml_env) sirasittanrattanawong@Sirasits-MacBook-Air Fundamental_AI_Tutorial %
```

- You can activate your env by using
conda activate <env name>

- For best practice, you should create requirements.txt file to install necessary library to the environment you created using conda

You can use this website to select the suggested version In requirements.txt, it should be something like this

The screenshot shows the PyTorch GitHub repository. A specific release, "torch 2.2.0", is highlighted. Below the release, there's a "pip install torch" button and a note: "Tensors and Dynamic neural networks in Python with strong GPU acceleration". The "Project description" tab is selected, showing a brief overview of PyTorch's features.

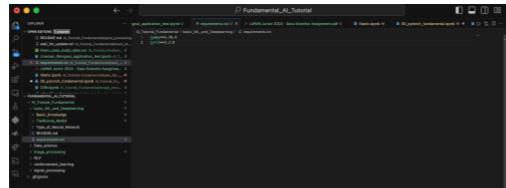


Then you can navigate to the folder of requirements.txt and use command pip install -r requirements.txt

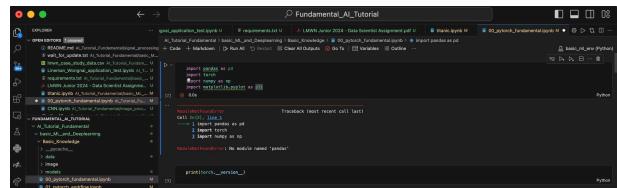
your current env

Next when you want to install more libraries you could add them to the requirements.txt file then use
pip install -r requirements.txt
all of your library will be in your env

ex. current requirements.txt

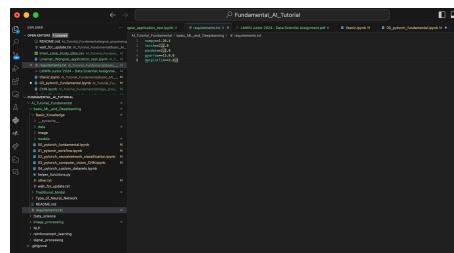


you don't have pandas library



update requirements.txt and

```
pip install -r requirements.txt
```

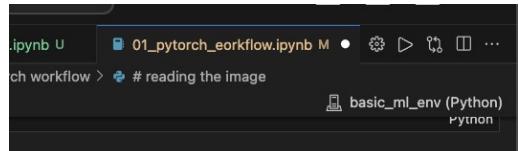


for running with symbol

not my turn command?

you need to change env

for every notebook you run



వ్యాపారికి మెంటెక్

ఫోన్‌లో సొర్ట్

ఇంసెస్ట్జిషన్

1. conda create env & new project

2. new file requirements.txt

3. pip install -r requirements.txt

(or) pip install -r requirements.txt

prerequisite

1. Install Python

2. Install **homebrew**

3. Install anaconda

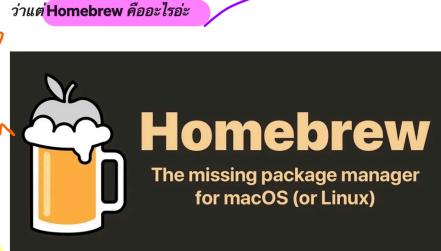
దీనిలోను / లోకు

(can be download with and without homebrew)

Part 2

ກົດປັບດີກາຣາໂອທ່ານີ້ສະບັບໄເລ່ກ = project/env

ໃຫຍ່ກໍາ (ຫຼືໃຫ້ກໍາດົບດັດລວມມີດັງໄຟເຊິ່ງ globally) `brew`



ກົດຕົກຕັ້ງຂາຍເຊື້ອ

(ເຂົ້າຮັບສັບ)

(ຕະຫຼາກປິ່ງຕົວ)

Homebrewແມ່ນເຂົ້າໃຈໄດ້ຈາຍໆ ມັນຄືອ Software ຕັ້ນນີ້ທີ່ກໍາໄໃຫ້ຮາມາດໃນປິດຕັ້ງ Software ຜົນ້າໃນເຄືອງ Mac ຂອງຮາໄຕ໌ (ຂຶ້ນ Node.js Docker ແລະ ນຸ່າງ) ຜ່ານ Command line ແລ້ວ

Check homebrew version

```
[(base) sirasittanrattanawong@sirasits-air ~ % brew --version
Homebrew 4.2.20
```

Update

How to Update Homebrew

Updating Homebrew is pretty straight forward:

```
brew update
```



This updates homebrew itself.

You can then upgrade all individual packages and formula with the following:

```
brew upgrade
```

What is homebrew



ulyssesric · 3y ago

HomeBrew is the easiest way to access open source community resources on your Mac.

Open source community handles software distribution in a completely different way. There are a lot projects opened to public on the internet; everyone can participate and make contributions. Usually, a project is focusing on one single purpose, for example, HTTP GET/POST handling, and such project is packaged with programmatically callable APIs, so that other people can embed it into their code. We called such software a "library".

Now try to think as a coder; if you're going to build something, say, a web browser, and you don't want to reinvent the wheel, what will you do? You install libraries on a computer, such as the one that handles HTTP GET/POST, so that you can write codes to combine them together.

The problem is: you need to download and install these libraries on your computer, but these libraries may need another different set of libraries to be installed first. Each library will have a list of needed libraries, so you have to download and install them one by one.

Yep, it's tedious and takes a lot of efforts.

And now imagine that one component on these installation chain needs to be updated.

HomeBrew is exactly a tool designed for such tasks. **It can automatically download all necessary projects, and automatically keep them up-to-date, in a single command.** And that's why is a priceless treasure for coders.

Aside from coders, there are a lot of very powerful tools distributed this way. For example, **FFmpeg** and **ImageMagick**. The former is an omnipotent video/audio converter, and the latter is an omnipotent image editor. Both works under command line and doesn't have GUI. Packages like these are essential for power users to make regular tasks into automation procedures, with scriptable tools like Automator, BetterTouchTool, Keyboard Maestro, Hazel or Alfred.

ຫົວໜາລາຍການ package ອັດໄດ້ (ໃນ github-classroom)

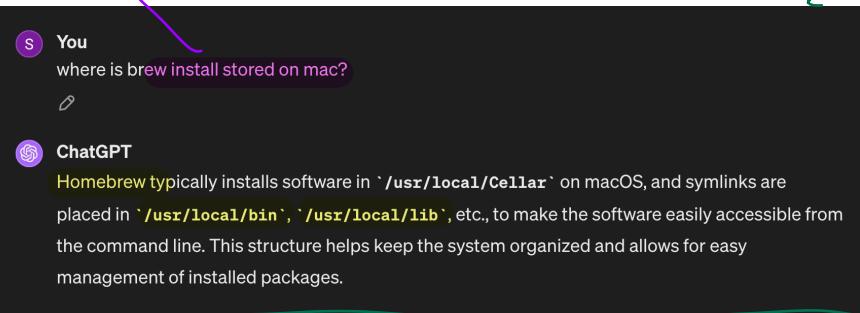
(ຮູບພົບໃຈນີ້ແມ່ນບໍ່ໄດ້ໃນ project ໂອງ) *

ໃຫ້ກຳສຳເນົາ brew install

path doesn't matter

Slide 6 back-end
software prac

ລວມຈຸດຂະໜາດທີ່ຈະໄດ້



```
[(base) sirasittanrattanawong@sirasits-air ~ % which gh  
/usr/local/bin/gh]
```

ເກົ່າຫຼື check version ສັນນິກົງ path doesn't matter

```
[(base) sirasittanrattanawong@sirasits-air ~ % pwd  
/Users/sirasittanrattanawong  
[(base) sirasittanrattanawong@sirasits-air ~ % gh --version  
gh version 2.43.1 (2024-01-31)]
```

git:(main) \$ brew install newman

{ brew install newman }

Slide 18 back-end

soft devops

path doesn't matter

v1 vs v1 vs v1

```
http://github.com/011111/releases/tag/v2.1.0.1  
[(base)] sirasittanrattanawong@sirasits-air ~ % which newman  
/usr/local/bin/newman  
[(base)] sirasittanrattanawong@sirasits-air ~ % newman --version  
6.1.2
```

not check won't return

path doesn't matter

Part 2.2 download anaconda mac

2.2.1 using homebrew

Search command : how to check anaconda version using homebrew

Link : <https://medium.com/@adhal/setting-up-anaconda-on-macos-ventura-13-2-1-0de3baa81d52>

2.2.2 no homebrew

Search command : how to install anaconda macos

Link : <https://docs.anaconda.com/free/anaconda/install/mac-os/>

→ way 1. graphical

→ way 2. command line

Part 3

ໄລ້ npm ຈີ່?

npm ລາຍງານໃຈຕົວລາຍລະອຽດເຊີ້ມ

∴ npm ບໍລິສັດ global (ນິການ condal Tools -g)

npm install ... → within a project

npm install -g ... → globally

Web programming env



javascript need not to be downloaded
not sure yet why

pre-requisite

1. install nodejs (itself comp-hr)

2. install extension

1. Bracket Pair Colorizer

2. DotENV → parse env

3. ExpressSnippet → highlight express code

4. JavaScript (ES6) code snippets → highlight modern script

5. Prettier - Code formatter → format code

6. REST Client → client





in project vs

1. `node_modules` folder

2. `npm init`

3. `npm install ...` → package / lib directory

global backend

soft dev proc

slide b

private

`npm install ...` → within a project

`npm install -g ...` → globally

What is sudo

More information about linux from : OS/activity/activity2_linux_sudo_apt/
Activity_2_UNIX_and_C_compiler-335-17049563016604.pdf

sudo Command in Linux with Examples

Last Updated : 24 April, 2024

sudo (Super User DO) command in Linux is generally used as a prefix for some commands that only superusers are allowed to run. If you prefix any command with "sudo", it will run that command with elevated privileges or in other words allow a user with proper permissions to execute a command as another user, such as the superuser. This is the equivalent of the "run as administrator" option in Windows. The option of sudo lets us have multiple administrators.

These users who can use the sudo command need to have an entry in the sudoers file located at "/etc/sudoers". Remember that to edit or view the sudoers file you have to use the sudo command. To edit the sudoers file it is recommended to use the "visudo" command.

By default, sudo requires that users authenticate themselves with a password that is the user's password, not the root password itself.

Detail of sudo /apt install is in

OS/activity/activity2_linux_sudo_apt/
Activity_2_UNIX_and_C_compiler-335-170495630
16604.pdf

Detail of portnumber, localhost, ip address configuration to start mn server
- mongo
- redis (in memory data storage)