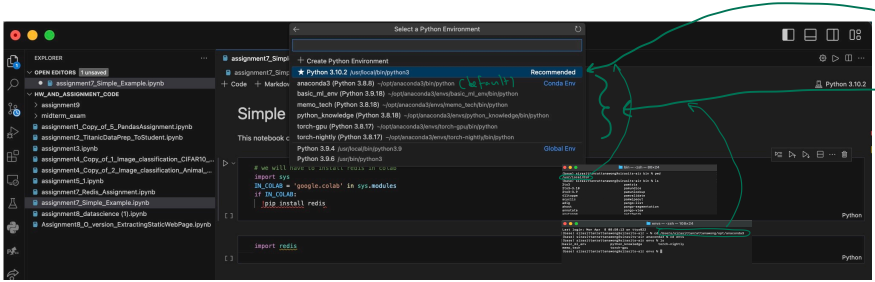
still waiting for update Now at programming | ENV And Path

There are 2 main types of using python


EXPLORER
OPEN EDITORS 1 unsaved
assignment7_Simple_Example.ipynb
HW_AND_ASSIGNMENT_CODE
assignment9
midterm_exam
assignment1_Copy_of_5_PandasAssignment.ipynb
assignment2_TitanicDataPrep_ToStudent.ipynb
assignment3.ipynb
assignment4_Copy_of_1_Image_classification_CIFAR10_...
assignment4_Copy_of_2_Image_classification_Animal_...
assignment5_1.ipynb
assignment7_Redis_Assignment.ipynb
assignment7_Simple_Example.ipynb
assignment8_datascience (1).ipynb
Assignment8_O_version_ExtractingStaticWebPage.ipynb
+ Code + Markdow
Simple
This notebook c
Python 3.10.2
Python

Select a Python Environment

- + Create Python Environment
- ★ Python 3.10.2 /usr/local/bin/python3 — Recommended
- anaconda3 (Python 3.8.8) ~/opt/anaconda3/bin/python (default) — Conda Env
- basic_ml_env (Python 3.9.18) ~/opt/anaconda3/envs/basic_ml_env/bin/python
- memo_tech (Python 3.8.18) ~/opt/anaconda3/envs/memo_tech/bin/python
- python_knowledge (Python 3.8.18) ~/opt/anaconda3/envs/python_knowledge/bin/python
- torch-gpu (Python 3.8.17) ~/opt/anaconda3/envs/torch-gpu/bin/python
- torch-nightly (Python 3.8.17) ~/opt/anaconda3/envs/torch-nightly/bin/python
- Python 3.9.4 /usr/local/bin/python3.9 — Global Env
- Python 3.9.6 /usr/bin/python3

```
# we will have to install redis in colab
import sys
IN_COLAB = 'google.colab' in sys.modules
if IN_COLAB:
  !pip install redis
```

```
import redis
```

bin — -zsh — 80×24

```
(base) sirasittanrattanawong@sirasits-air bin % pwd
/usr/local/bin
(base) sirasittanrattanawong@sirasits-air bin % ls
2to3                          pamtris
2to3-3.10                     pamundice
2to3-3.9                      pamunlookup
411toppm                      pamvalidate
acyclic                       pamwipeout
adig                          pango-list
ahost                         pango-segmentation
annotate                      pango-view
```

envs — -zsh — 108×24

```
Last login: Mon Apr  8 08:50:13 on ttys022
(base) sirasittanrattanawong@sirasits-air ~ % cd /Users/sirasittanrattanawong/opt/anaconda3
(base) sirasittanrattanawong@sirasits-air anaconda3 % cd envs
(base) sirasittanrattanawong@sirasits-air envs % ls
basic_ml_env          python_knowledge        torch-nightly
memo_tech             torch-gpu
(base) sirasittanrattanawong@sirasits-air envs %
```

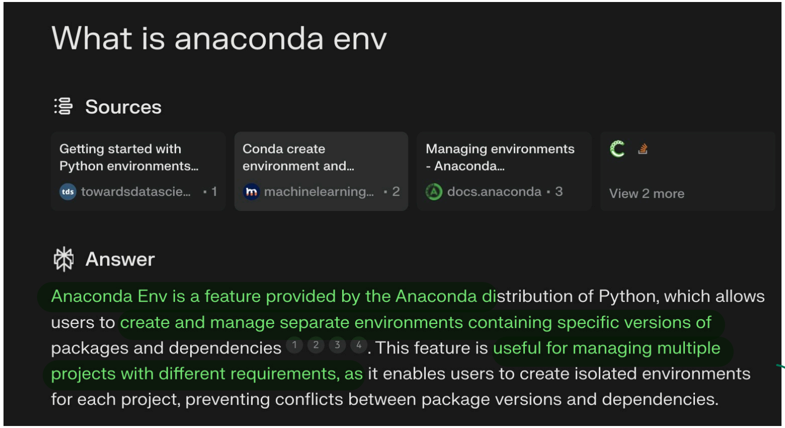
1. python env. /usr/local/bin (แบบปกติ default)

2. anaconda env (used in data science project)

/Users/sirasittanrattanawong/opt/anaconda3/bin

(you need to install anaconda first)

งานเรามักใช้แบบนี้!!



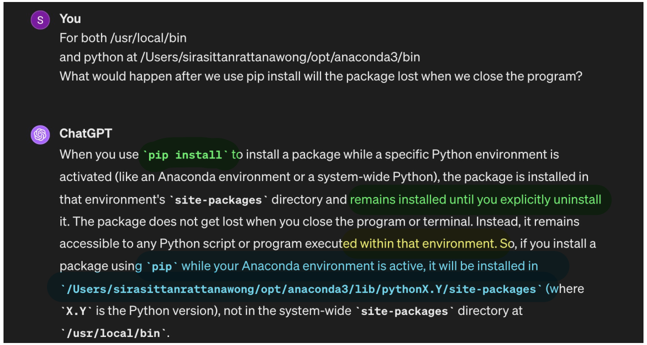
What is anaconda env

Sources

Getting started with Python environments...	Conda create environment and...	Managing environments - Anaconda...	View 2 more
towardsdatascie... · 1	machinelearning... · 2	docs.anaconda · 3	

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 1 2 3 4. 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.



You

For both /usr/local/bin
and python at /Users/sirasittanrattanawong/opt/anaconda3/bin
What would happen after we use pip install will the package lost when we close the program?

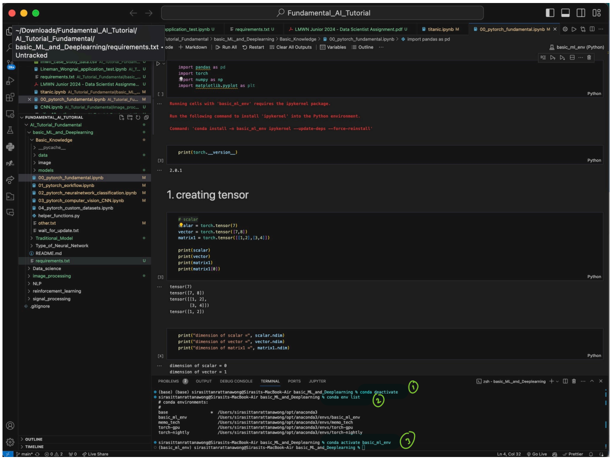
ChatGPT

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/sirasittanrattanawong/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 environemnt 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`


Fundamental_AI_Tutorial
~/Downloads/Fundamental_AI_Tutorial/AI_Tutorial_Fundamental/basic_ML_and_Deeplearning/requirements.txt • Untracked
application_test.ipynb U
requirements.txt U
LMWN Junior 2024 - Data Scientist Assignment.pdf U
titanic.ipynb M
00_pytorch_fundamental.ipynb M
Tutorial_Fundamental > basic_ML_and_Deeplearning > Basic_Knowledge > 00_pytorch_fundamental.ipynb > import pandas as pd
Code + Markdown Run All Restart Clear All Outputs Variables Outline
basic_ml_env (Python)
Lineman_Wongnai_application_test.ipynb U
requirements.txt U
LMWN Junior 2024 - Data Scientist Assignme... U
titanic.ipynb M
00_pytorch_fundamental.ipynb M
CNN.ipynb U
FUNDAMENTAL_AI_TUTORIAL
AI_Tutorial_Fundamental
basic_ML_and_Deeplearning
Basic_Knowledge
__pycache__
data
image
models
00_pytorch_fundamental.ipynb M
01_pytorch_eorkflow.ipynb M
02_pytorch_neuralnetwork_classification.ipynb M
03_pytorch_computer_vision_CNN.ipynb M
04_pytorch_custom_datasets.ipynb
helper_functions.py
other.txt M
wait_for_update.txt
Traditional_Model
Type_of_Neural_Network
README.md
requirements.txt U
Data_science
image_processing
NLP
reinforcement_learning
signal_processing
.gitignore
OUTLINE
TIMELINE
main*
Live Share
Python
PROBLEMS 2
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
JUPYTER
zsh - basic_ML_and_Deeplearning
Ln 4, Col 32
Go Live
Prettier
①
②
③

```python
import pandas as pd
import torch
import numpy as np
import matplotlib.pyplot as plt
```

```
Running cells with 'basic_ml_env' requires the ipykernel package.

Run the following command to install 'ipykernel' into the Python environment.

Command: 'conda install -n basic_ml_env ipykernel --update-deps --force-reinstall'
```

[3]

```python
print(torch.__version__)
```

```
2.0.1
```

1. creating tensor

[3]

```python
# scalar
scalar = torch.tensor(7)
vector = torch.tensor([7,8])
matrix1 = torch.tensor([[1,2],[3,4]])

print(scalar)
print(vector)
print(matrix1)
print(matrix1[0])
```

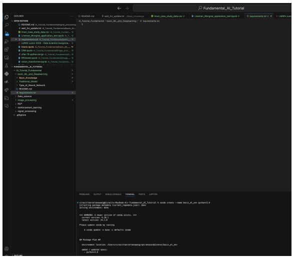
```
tensor(7)
tensor([7, 8])
tensor([[1, 2],
        [3, 4]])
tensor([1, 2])
```

[4]

```python
print("dimension of scalar =", scalar.ndim)
print("dimension of vector =", vector.ndim)
print("dimension of matrix1 =", matrix1.ndim)
```

```
dimension of scalar = 0
dimension of vector = 1
```

```
(base) (base) sirasittanrattanawong@Sirasits-MacBook-Air basic_ML_and_Deeplearning % conda deactivate
sirasittanrattanawong@Sirasits-MacBook-Air basic_ML_and_Deeplearning % conda env list
# conda environments:
#
base                  *  /Users/sirasittanrattanawong/opt/anaconda3
basic_ml_env             /Users/sirasittanrattanawong/opt/anaconda3/envs/basic_ml_env
memo_tech                /Users/sirasittanrattanawong/opt/anaconda3/envs/memo_tech
torch-gpu                /Users/sirasittanrattanawong/opt/anaconda3/envs/torch-gpu
torch-nightly            /Users/sirasittanrattanawong/opt/anaconda3/envs/torch-nightly

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

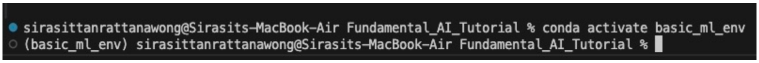


```
sirasittanrattanawong@Sirasits-MacBook-Air Fundamental_AI_Tutorial % conda create --name basic_ml_env python=3.9
Collecting package metadata (current_repodata.json): done
Solving environment: done


==> WARNING: A newer version of conda exists. <==
  current version: 4.10.1
  latest version: 24.1.0

Please update conda by running

    $ conda update -n base -c defaults conda



## Package Plan ##

  environment location: /Users/sirasittanrattanawong/opt/anaconda3/envs/basic_ml_env

  added / updated specs:
    - python=3.9
```

(skip this part if you have env)

1. 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
```

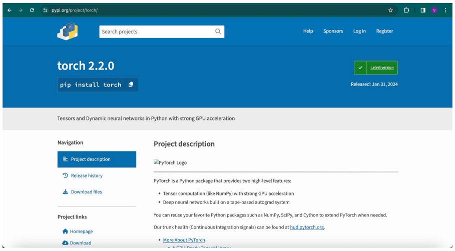


```
sirasittanrattanawong@Sirasits-MacBook-Air Fundamental_AI_Tutorial % conda activate basic_ml_env
(basic_ml_env) sirasittanrattanawong@Sirasits-MacBook-Air Fundamental_AI_Tutorial %
```

2. You can activate your env by using conda activate <env name >

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

You can use this website to select the suggetsed version



pypi.org/project/torch/

Search projects

Help Sponsors Log in Register

torch 2.2.0

`pip install torch`

✓ Latest version

Released: Jan 31, 2024

Tensors and Dynamic neural networks in Python with strong GPU acceleration

Navigation

- Project description
- Release history
- Download files

Project links

- Homepage
- Download

Project description

PyTorch Logo

PyTorch is a Python package that provides two high-level features:

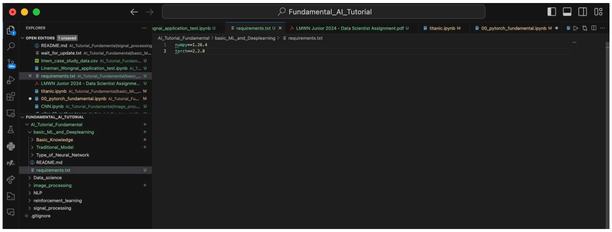
- Tensor computation (like NumPy) with strong GPU acceleration
- Deep neural networks built on a tape-based autograd system

You can reuse your favorite Python packages such as NumPy, SciPy, and Cython to extend PyTorch when needed.

Our trunk health (Continuous Integration signals) can be found at hud.pytorch.org.

- More About PyTorch

In requirements.txt, it should be something like this

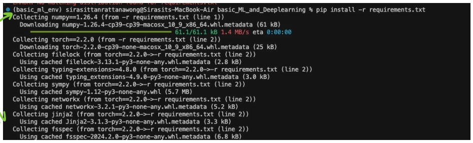


Fundamental_AI_Tutorial
EXPLORER
OPEN EDITORS 1 unsaved
README.md AI_Tutorial_Fundamental/signal_processing
wait_for_update.txt AI_Tutorial_Fundamental/basic_M...
lmwn_case_study_data.csv AI_Tutorial_Fundam... U
Lineman_Wongnai_application_test.ipynb AI_T... U
requirements.txt AI_Tutorial_Fundamental/basic_... U
LMWN Junior 2024 - Data Scientist Assignme... U
titanic.ipynb AI_Tutorial_Fundamental/basic_ML_... M
00_pytorch_fundamental.ipynb AI_Tutorial_Fu... M
CNN.ipynb AI_Tutorial_Fundamental/image_proc... U
FUNDAMENTAL_AI_TUTORIAL
AI_Tutorial_Fundamental
basic_ML_and_Deeplearning
Basic_Knowledge
Traditional_Model
Type_of_Neural_Network
README.md
requirements.txt U
Data_science
image_processing
NLP
reinforcement_learning
signal_processing
.gitignore
AI_Tutorial_Fundamental > basic_ML_and_Deeplearning > requirements.txt

```
numpy==1.26.4
torch==2.2.0
```

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

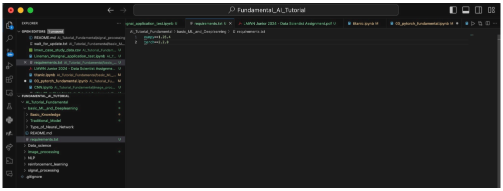
your current env



```
(basic_ml_env) sirasittanrattanawong@Sirasits-MacBook-Air basic_ML_and_Deeplearning % pip install -r requirements.txt
Collecting numpy==1.26.4 (from -r requirements.txt (line 1))
  Downloading numpy-1.26.4-cp39-cp39-macosx_10_9_x86_64.whl.metadata (61 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 61.1/61.1 kB 1.4 MB/s eta 0:00:00
Collecting torch==2.2.0 (from -r requirements.txt (line 2))
  Downloading torch-2.2.0-cp39-none-macosx_10_9_x86_64.whl.metadata (25 kB)
Collecting filelock (from torch==2.2.0->-r requirements.txt (line 2))
  Using cached filelock-3.13.1-py3-none-any.whl.metadata (2.8 kB)
Collecting typing-extensions>=4.8.0 (from torch==2.2.0->-r requirements.txt (line 2))
  Using cached typing_extensions-4.9.0-py3-none-any.whl.metadata (3.0 kB)
Collecting sympy (from torch==2.2.0->-r requirements.txt (line 2))
  Using cached sympy-1.12-py3-none-any.whl (5.7 MB)
Collecting networkx (from torch==2.2.0->-r requirements.txt (line 2))
  Using cached networkx-3.2.1-py3-none-any.whl.metadata (5.2 kB)
Collecting jinja2 (from torch==2.2.0->-r requirements.txt (line 2))
  Using cached Jinja2-3.1.3-py3-none-any.whl.metadata (3.3 kB)
Collecting fsspec (from torch==2.2.0->-r requirements.txt (line 2))
  Using cached fsspec-2024.2.0-py3-none-any.whl.metadata (6.8 kB)
```

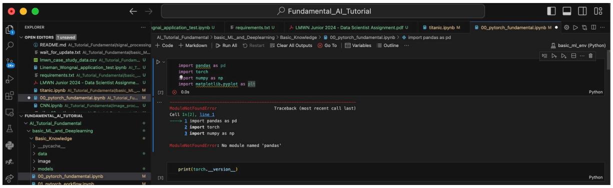
Next when you want to install more libaries you could add them to the requirements.txt file then use
pip install -r requirements.txt
all of your libary will be in your env

ex. current requirements.txt


Fundamental_AI_Tutorial
requirements.txt
AI_Tutorial_Fundamental > basic_ML_and_Deeplearning > requirements.txt

```
numpy==1.26.4
torch==2.2.0
```

you don't have pandas libary


Fundamental_AI_Tutorial
AI_Tutorial_Fundamental > basic_ML_and_Deeplearning > Basic_Knowledge > 00_pytorch_fundamental.ipynb > import pandas as pd
Code
Markdown
Run All
Restart
Clear All Outputs
Go To
Variables
Outline
basic_ml_env (Python)

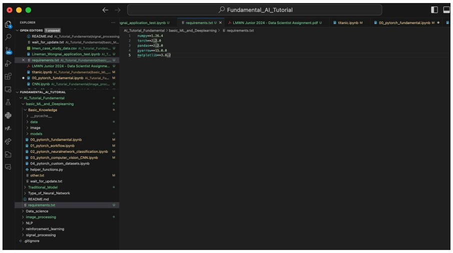
```
import pandas as pd
import torch
import numpy as np
import matplotlib.pyplot as plt
```

```
ModuleNotFoundError                       Traceback (most recent call last)
Cell In[2], line 1
----> 1 import pandas as pd
      2 import torch
      3 import numpy as np

ModuleNotFoundError: No module named 'pandas'
```

```
print(torch.__version__)
```

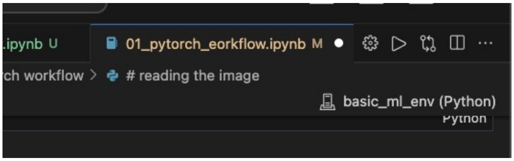
update requirements.txt and

pip install -r requirements.txt


Fundamental_AI_Tutorial
requirements.txt
AI_Tutorial_Fundamental > basic_ML_and_Deeplearning > requirements.txt

```
numpy==1.26.4
torch==2.2.0
pandas==2.2.0
pyarrow==15.0.0
matplotlib==3.8.2
```

for running with symbol not python command you need to change env for every notebook you run


.ipynb U
01_pytorch_eorkflow.ipynb M
ch workflow > # reading the image
basic_ml_env (Python)
Python

ตัวอย่างการทำงานของ memo_tech

1. conda create env สำหรับ project นี้

2. สร้าง file requirements.txt

3. เวลาต้อง install อะไรก็ใส่ลงไปใน requirements.txt

   แล้ว pip install -r requirements.txt

Web programming env