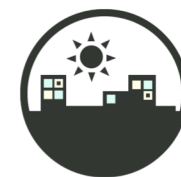


Python Virtual Environment Workshop

With Conda



日不落开发工作室
sunneversets.studio

张智翼 (116010308@link.cuhk.edu.cn)

Overview

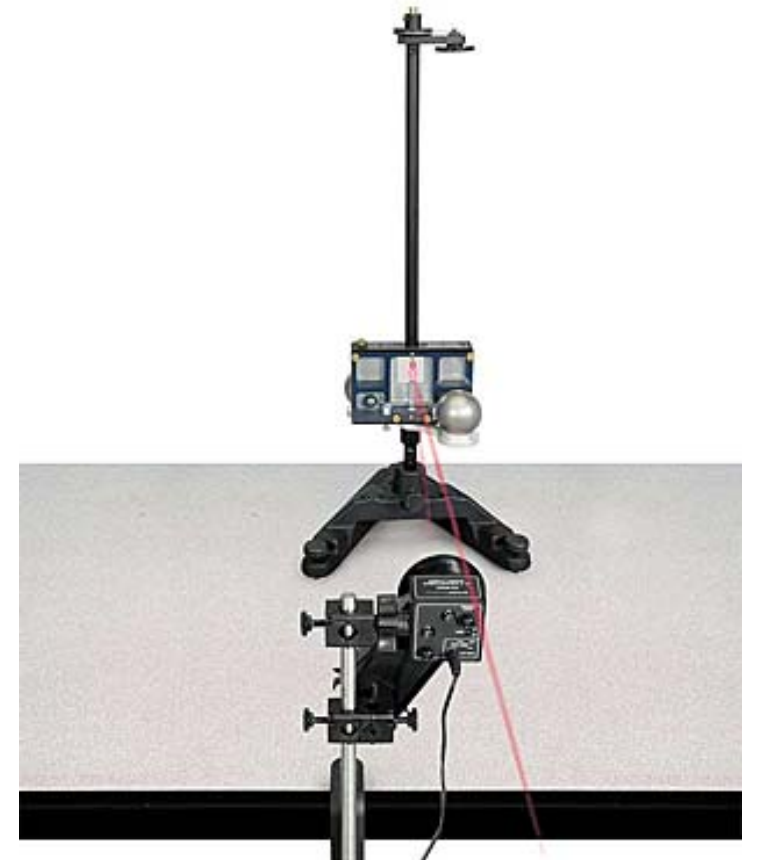
- Why we need virtual environment?
- Manage virtual environment with Conda
 - Create, List, Activate/Deactivate, Install/Remove, Export, Delete
- Further info.
 - Other environment manager: virtualenv, Pipenv
 - How it works (simplified)
- Reference

Note

- Before the workshop, ensure that you've installed Anaconda from <https://www.anaconda.com/distribution/#download-section>
- You can check the guideline for installation via:
<https://github.com/Sunneversets-Studio/workshop/tree/master/Python%20Virtual%20Environment%20with%20Anaconda>
- We recommend the latest version for python 3.

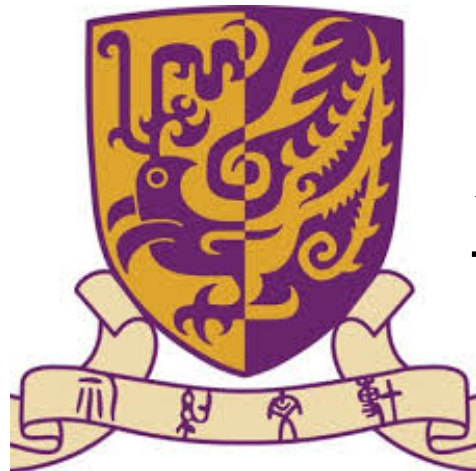
Why we need virtual environment?

- Let's consider a situation in PHY1002



Why we need virtual environment?

- Let's consider a situation if we have a campus named CUHK(LA)



香港中文大学（洛杉矶）

The Chinese University of Hong Kong, Los Angeles

Note: it is a virtual campus.*

Why we need virtual environment?

- Equipment in lab <—> Packages in Python
- Influences between different equipment <—> Packages conflict
- Isolate equipment <—> Create virtual environment
- Purchase the same equipment <—> Export/Import from requirements.txt

Note: it is a virtual campus.*

Manage virtual environment



Anaconda is a free and open-source distribution of Python and R for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. Conda is a tool of Anaconda.

Manage virtual environment

- The following instructions are run in Anaconda Prompt (Windows) / Terminal (MacOS/Linux).
- Create virtual environment
 - `conda create -n <env_name> <package_names>`
- Note:
 - `-n` can be replaced by `--name`, with environment name following it;
 - **<env_name>** is the environment's name. Prefer to set it in English without blank space. Do not add unnecessary `<>`.
 - **<package_names>** is the package to be installed. Do not add unnecessary `<>`.
 - If you need to specify the version, use `=` syntax with the version number. Eg. `conda create --name python2 python=2.7` means to create a virtual environment with python 2.7.
 - If you need to install several packages in the environment, you can just join their names with blank space. Eg. `conda create -n python3 python=3.5 numpy pandas` will create a virtual environment with python 3.5, numpy and pandas.

Manage virtual environment

- List all the virtual environments in the system (current user).
 - `conda env list`


A list similar to the following is displayed:

```
conda environments:
myenv              /home/username/miniconda/envs/myenv
snowflakes         /home/username/miniconda/envs/snowflakes
bunnies            /home/username/miniconda/envs/bunnies
```

Manage virtual environment

- Activate a virtual environment
 - For conda 4.6 and later version
 - `conda activate <env_name>`
 - For conda versions prior to 4.6
 - MacOS / Linux
 - `source activate <env_name>`
 - Windows
 - `activate <env_name>`

By default, the active environment---the one you are currently using---is shown in parentheses () or brackets [] at the beginning of your command prompt:



```
(myenv) $
```

Manage virtual environment

- Install packages in current environment
 - Use conda to install:
 - (<env_name>) \$ conda install <package_name>
 - Or:
 - (<env_name>) \$ pip install <package_name>

Manage virtual environment

- Import/Export packages information from/to requirements.txt
 - Import:
 - (<env_name>) \$ pip install -r requirements.txt
 - Export
 - (<env_name>) \$ pip freeze>requirements.txt
- List all the packages in current environment
 - (<env_name>) \$ conda list

Manage virtual environment

- Exit the virtual environment
 - For conda 4.6 and later version
 - `$ conda deactivate`For conda versions prior to 4.6
 - For conda version prior to 4.6
 - MacOS / Linux
 - `$ source deactivate`Windows
 - Windows
 - `> deactivate`

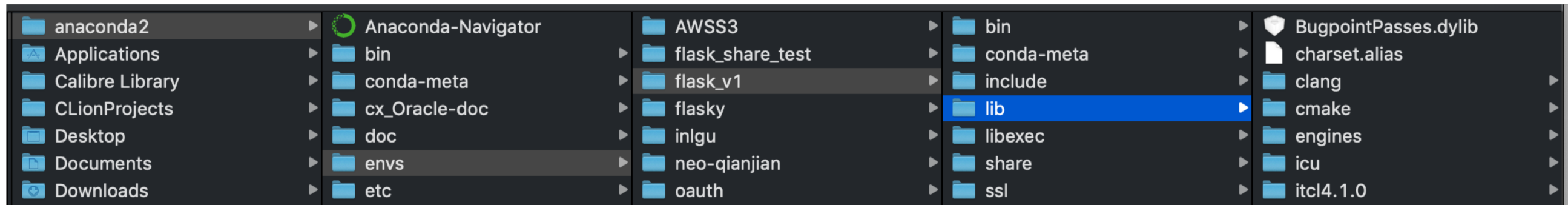
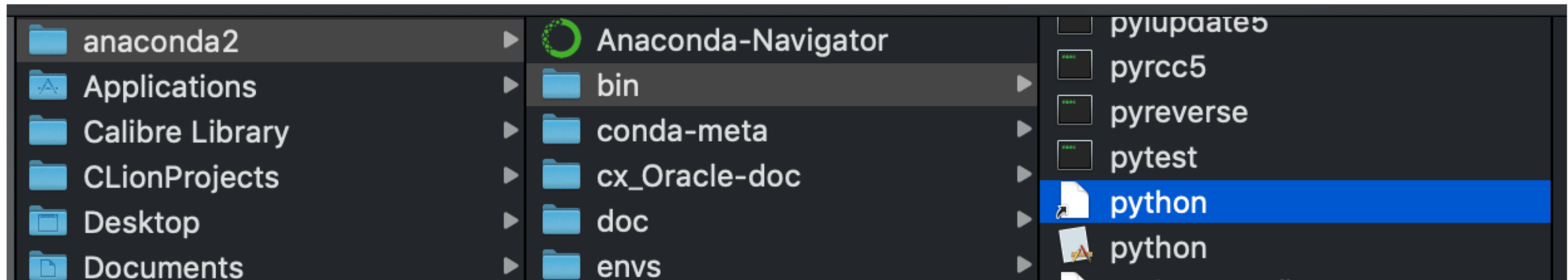
Manage virtual environment

- Delete packages in current environment
 - (<env_name>) \$ conda remove <package_name>
- Delete virtual environment
 - \$ conda remove --name <env_name> --all
 - \$ conda env remove --name <env_name>

Other Python virtual environment manager

Item	Pros.	Cons.
Anaconda	Python version + package version	Occupies a lot of space
Pipenv	Emerging official tool	Cannot manage Python version
pip + virtualenv	Traditional official tool	Cannot manage Python version

How it works (brief)



Summary

- The reasons to use virtual environment?
- Steps to manage virtual environment with Conda
 - Create, List, Activate/Deactivate, Install/Remove, Export, Delete
- Further info.
 - Other environment manager: virtualenv, Pipenv
 - How it works (simplified)

Q&A

Reference

- Anaconda 完全入门指南
- 关于 conda 和 anaconda 不可不知的误解和事实——conda 必知必会
- Anaconda 介绍、安装及使用教程