



Virtual environments

02.09.2023



Why do we need virtual environments?





3.10

3.11

pandas

v.1.2.3

biopython

numpy

v.0.1.0

2.0

opencv

https://en.wikipedia.org/wiki/Dependency_hell



Why do we need virtual environments?

Avoiding software dependency hell

Isolation of external code from the PC

Easy to install/move/remove





Which virtual environments do we have?

venv

virtualenv

+ virtualenvwrapper

conda / mamba





Which virtual environments do we have?

venv

virtualenv

+ virtualenvwrapper

conda / mamba





Creating virtual environment

—— Hands on your keyboard ——

```
python3 -m venv my_venv
```

```
ls -al
```

```
cd my_venv
```

```
ls -al
```

What do you see?





Using virtual environment

—— Hands on your keyboard ——

We can activate it and deactivate

```
which python
```

```
source my_venv/bin/activate
```

```
which python
```

```
deactivate
```

```
which python
```





Using virtual environment

—— Hands on your keyboard ——

Let's install some library!

```
source my_venv/bin/activate
```

```
ls my_venv/lib/python3.11/site-packages
```

```
pip install cowsay
```

```
my_venv/bin/cowsay -t hello
```





Using virtual environment

—— Hands on your keyboard ——

Let's install some library!

```
pip install biopython
```

numpy installed as well!

```
python3
```

```
from Bio.Seq import Seq  
my_dna = Seq('AGTACACTGGT')  
print(my_dna.complement())  
quit() / Ctrl + D
```





Sharing virtual environment

—— Hands on your keyboard ——

```
pip freeze
```

```
deactivate
```

```
pip freeze
```

```
source my_venv/bin/activate
```

```
pip freeze ...
```

Saving info about
our venv





Sharing virtual environment

—— Hands on your keyboard ——

```
pip freeze
```

Saving info about
our venv

```
deactivate
```

```
pip freeze
```

```
source my_venv/bin/activate
```

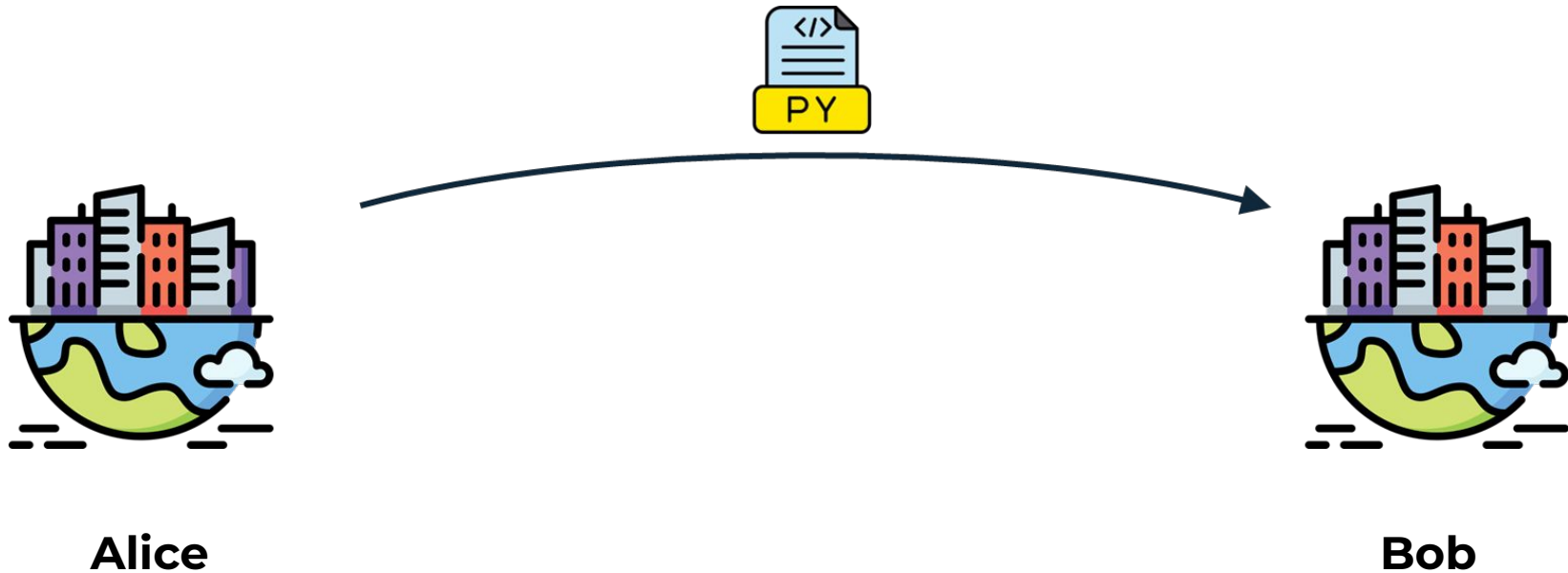
```
pip freeze > requirements.txt
```

```
cat requirements.txt
```





Sharing virtual environment





Sharing virtual environment

—— Hands on your keyboard ——

Reproducing venv

```
python3 -m venv my_venv2
```

```
source my_venv2/bin/activate
```

```
ls my_venv2/lib/python3.11/site-packages
```

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

```
ls my_venv2/lib/python3.11/site-packages
```





Deleting virtual environment





Deleting virtual environment

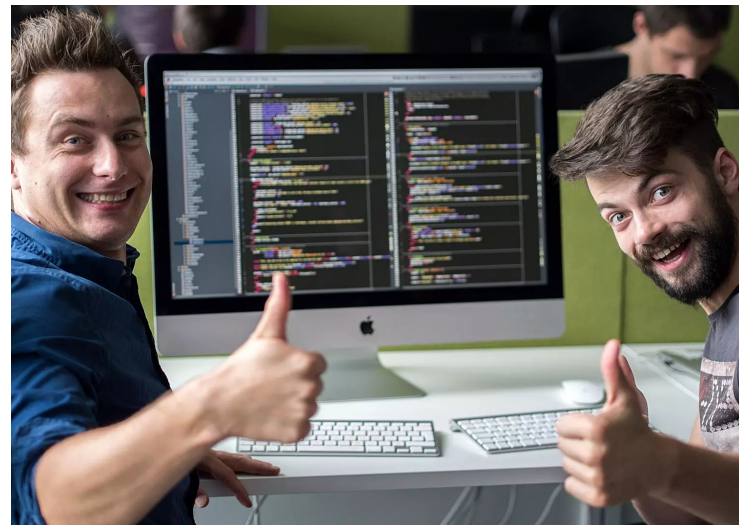
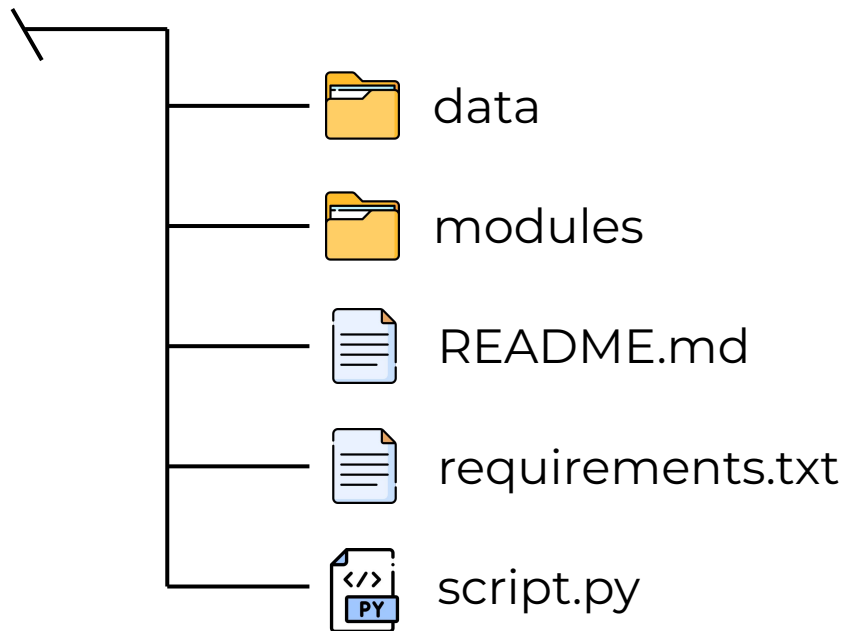
—— Hands on your keyboard ——

```
rm -r my_venv2
```



Example bioinformatics project

https://github.com/pavlovanadia/genotype_to_phenotype





Break!





Which virtual environments do we have?

venv

virtualenv

+ virtualenvwrapper

conda / mamba



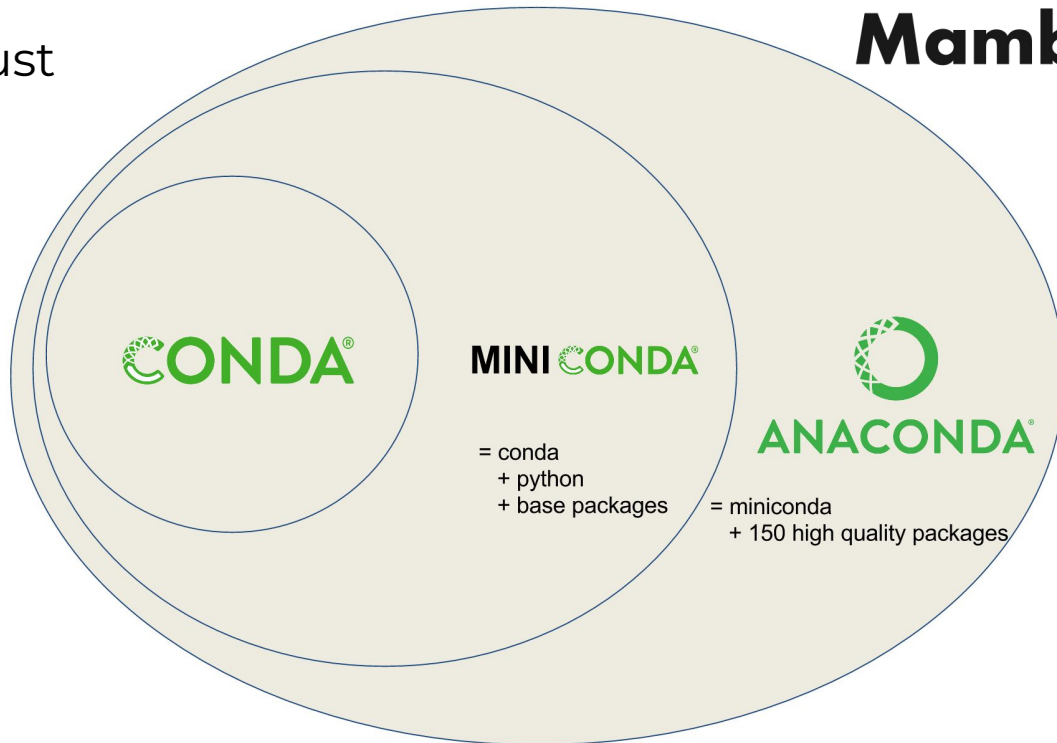


Conda environments

To manage more than just
python packages



Mamba





Two conda strategies

When to use environments?

Always!

When to create new environments?

For every tool

For every project





Using conda/mamba

—— Hands on your keyboard ——

Creating environments

```
conda create -n new_env
```

```
conda activate new_env
```

```
conda deactivate
```

People often make aliases `ca` and `cda`





Using conda/mamba

—— Hands on your keyboard ——

Inspecting environments

```
conda env list
```

```
conda list
```

```
conda info
```

```
conda env export
```



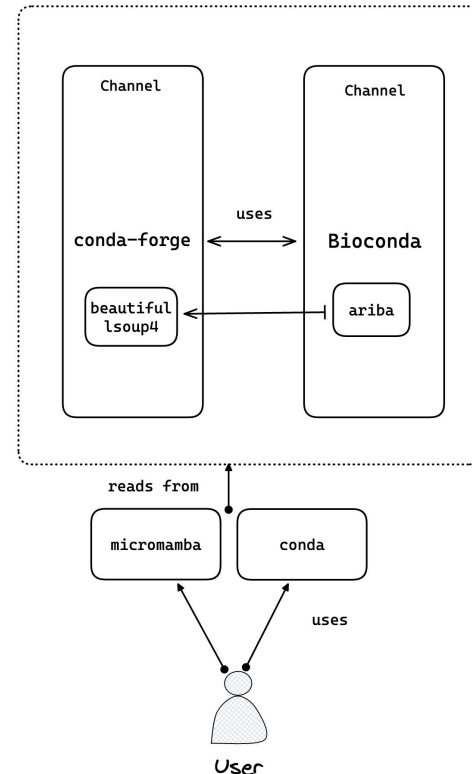
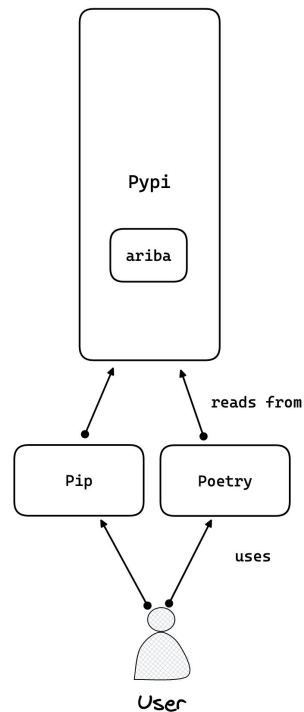


Conda channels

Conda channels are remote repositories with packages



BIOCONDA®





Using conda/mamba

—— Hands on your keyboard ——

Sharing environments

```
conda install -c channel
```

```
conda env export > environment.yml
```

```
conda env create -f environment.yml
```

```
conda env export --from-history
```





Wrap up

You can edit installed modules

Use virtual environments (venv for python and conda for practice)

Always make requirements or yml

