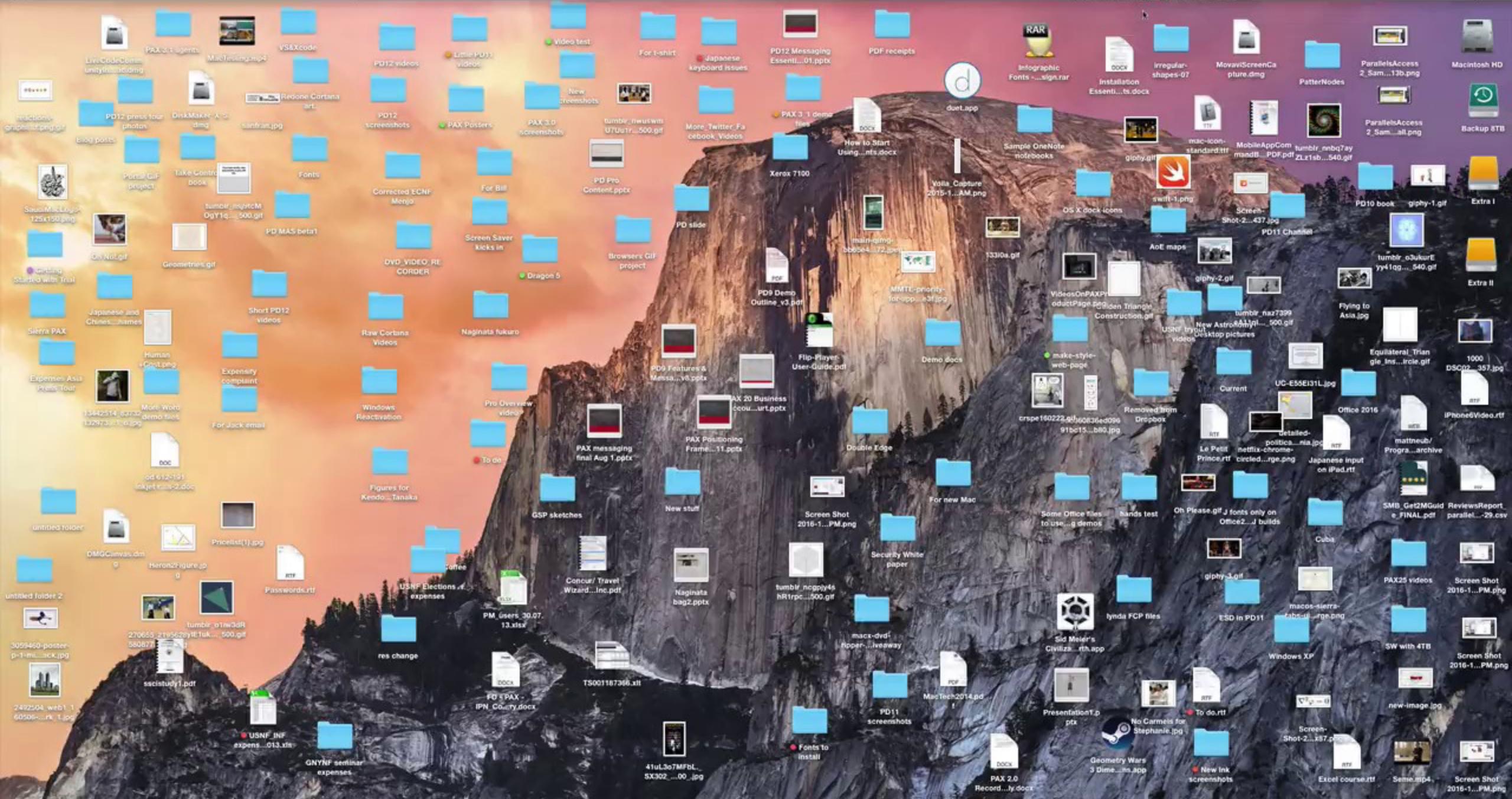


How to keep python projects organized?

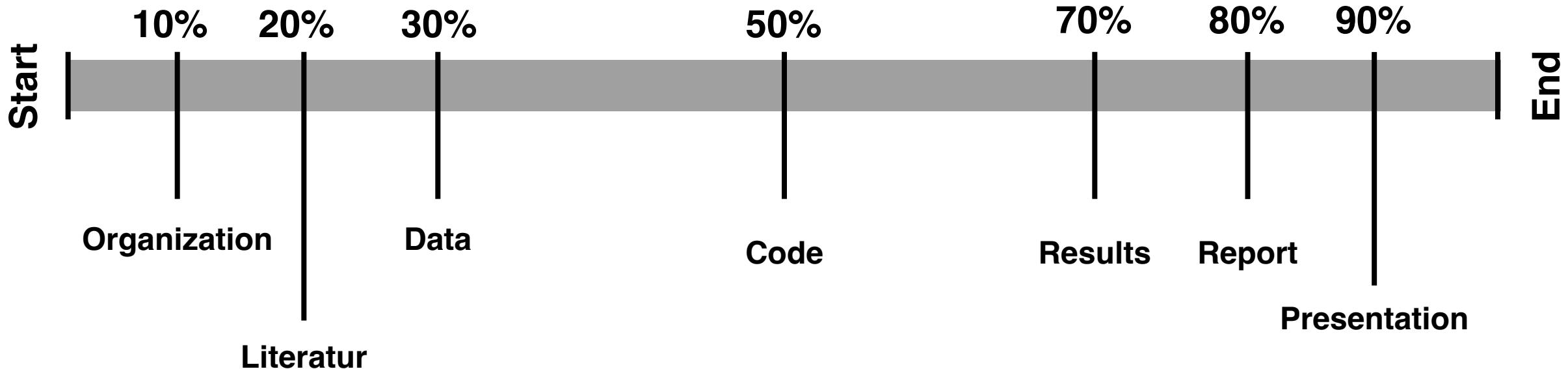
Markus Bonse

09. Jan 2024 @Astrowoche



How to keep files organized?

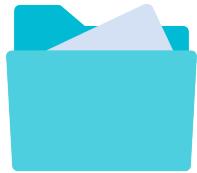
How to keep files organized?



Idea: use the same folder names for each project!



How to keep files organized?



10_orga



20_literatur



30_data



50_code



70_results



80_report



90_presentation

Keep your data separate from your code!

How to keep code organized?

Building code is like building a house...

It's a complicated project...

...but...

...we can divide it into sub-projects.

Building code is like building a house...

...we can divide it into sub-projects.

Plan



Bring material



Dig the hole



Raw construction



Make the roof



Finished



Building code is like building a house...

Each sub-project needs different tools!

Plan



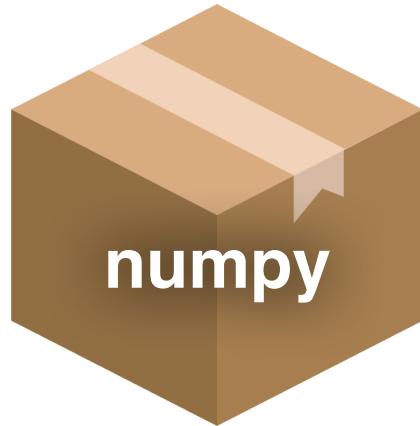
Dig the hole



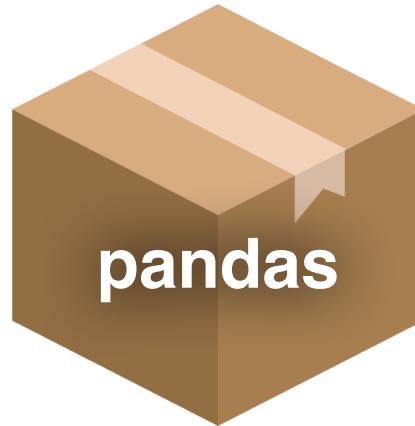
This is the idea of python packages!

Python Packages

Each sub-project needs different tools!



math



**work with
tables**



plot results

How do I manage python packages?

How do I manage python packages?

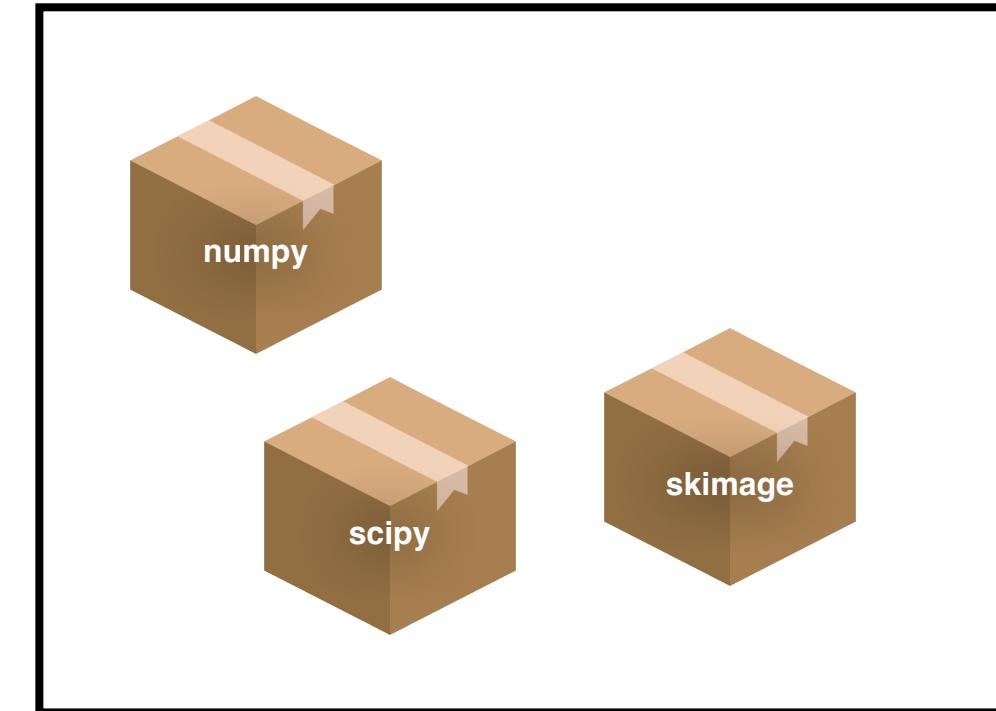


ANACONDA®



Python Virtualenv

**Virtualenv / Conda
environment**



How do I manage python packages?

What's Better ?



dataaspirant.com

<https://dataaspirant.com/anaconda-python-virtualenv/>

Python Installation



Individual Edition

Your data science toolkit

With over 25 million users worldwide, the open-source Individual Edition (Distribution) is the easiest way to perform Python/R data science and machine learning on a single machine. Developed for solo practitioners, it is the toolkit that equips you to work with thousands of open-source packages and libraries.

Anaconda Individual Edition

[Download](#)

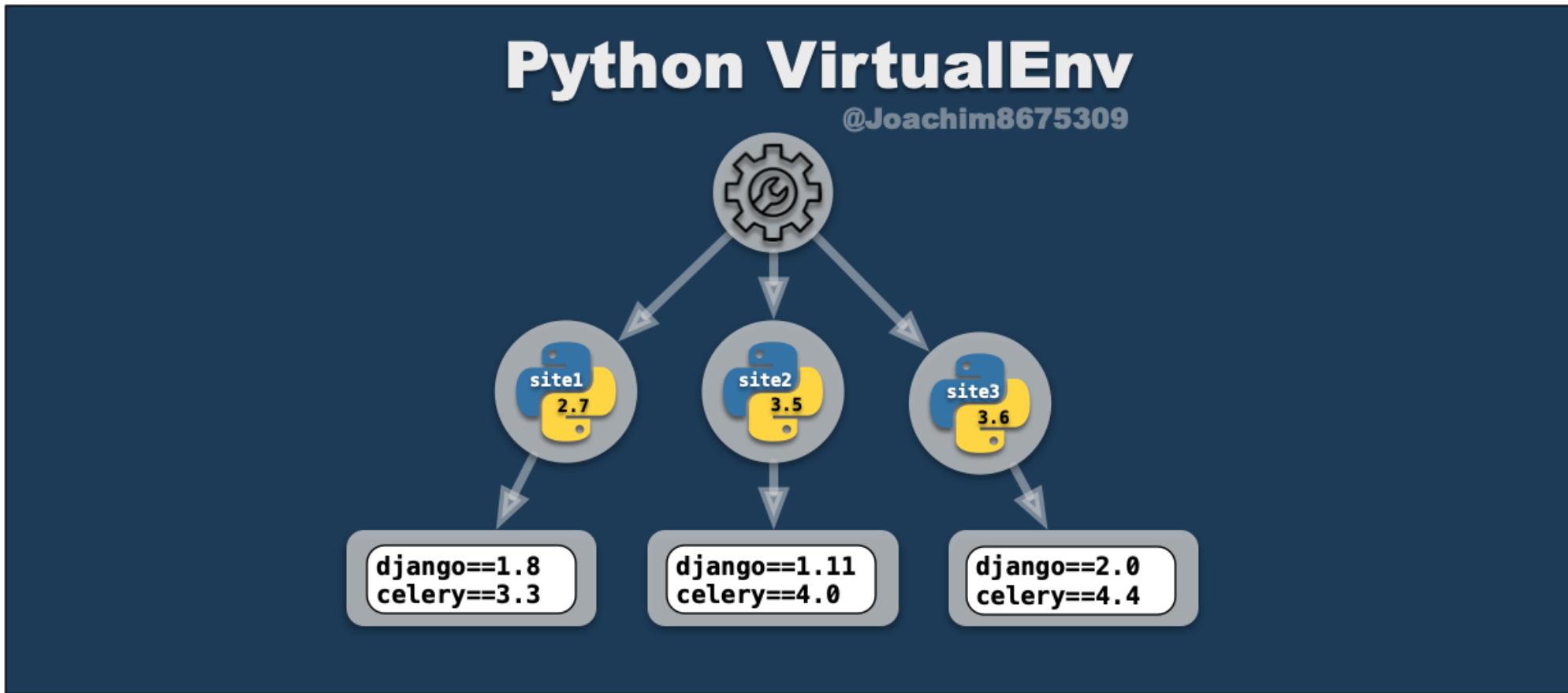
For MacOS
Python 3.9 • 64-Bit Graphical Installer • 515 MB

[Get Additional Installers](#)

| |

<https://www.anaconda.com/products/individual>

Use one virtual environment for each project!!!



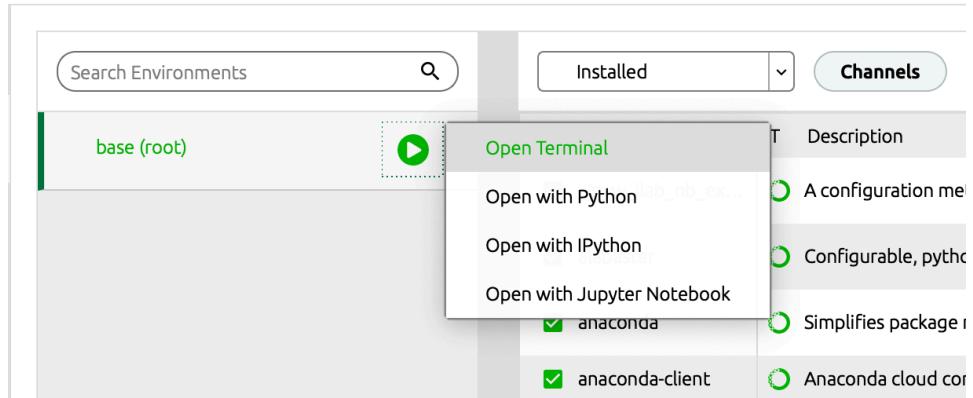
LIVE DEMO



Python Virtualenv

Setup with Virtualenv

(1)



(2)

```
conda install virtualenv
```

(3)

```
cd #50_code#
```

(4)

```
virtualenv venv_data_stats
```

(5)

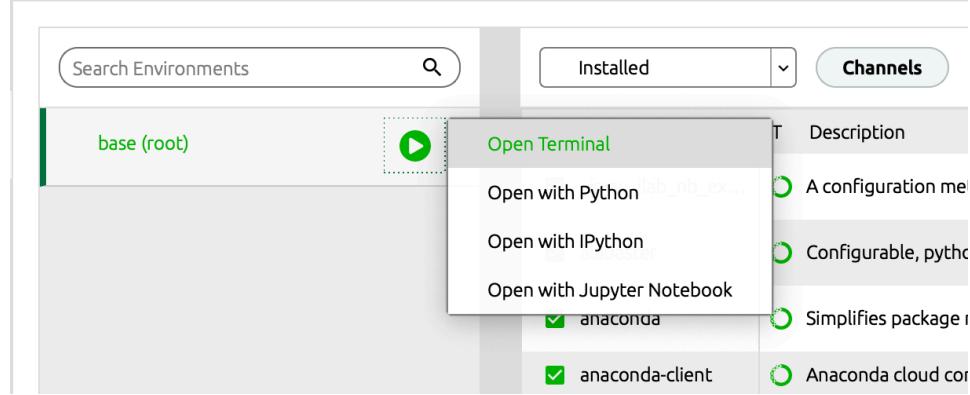
```
source venv_data_stats/bin/activate
```

(6)

```
pip install jupyter numpy matplotlib
```

Setup with Conda

(1)



(2)

```
cd #50_code#
```

(3)

```
conda create -p ./venv_data_stats
```

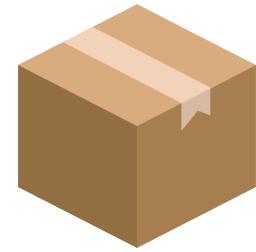
(4)

```
conda activate ./venv_data_stats
```

(5)

```
conda install jupyter pip numpy matplotlib
```

How to keep your own code clean?



Distinguish between prototyping and finished code!

My personal workflow...

My personal workflow...

(1)

Create python package:

See demo (https://github.com/markusbonse/demo_astrowoche)

(2)

Install my own package inside the virtualenv

```
pip install -e .      # make sure to be in the folder with the setup.py
```

(3)

Make virtualenv available in Jupyter

```
python -m ipykernel install --user --name venv_astrowoche --display-name "Astrowoche"
```

(4)

Prototyping in Jupyter notebooks

```
jupyter notebook
```

(5)

Copy stable code into the package

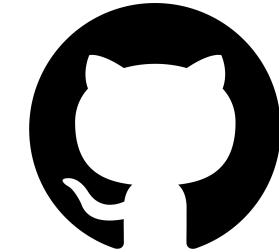
LIVE DEMO

Work in teams!



Google Colab

Work on Jupyter notebooks
like in Google docs



GitHub

Work on packages
in teams