

Python Tsunami

Installation and Tools

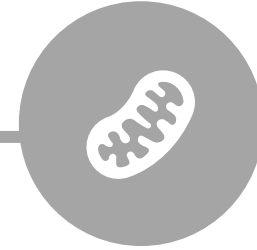
– June 7th-9th –



How do I work with Python?



SETTING UP
PYTHON



VIRTUAL
ENVIRONMENTS



JUPYTER
NOTEBOOKS

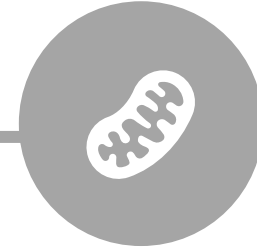


GOOGLE COLAB

How do I work with Python?



SETTING UP
PYTHON



VIRTUAL
ENVIRONMENTS



JUPYTER
NOTEBOOKS



GOOGLE COLAB

Setting up Python



Pros:

- Your **own local** installation
- Easy **access to data** stored locally
- More **control** over installed versions
- **Not difficult** if done **right**

Cons:

- May require solving **issues** (sometimes obscure) if **incorrectly installed**
- Each **Operating System** (Windows, MacOS, Linux, ...) may require **specific steps**

Setting up Python

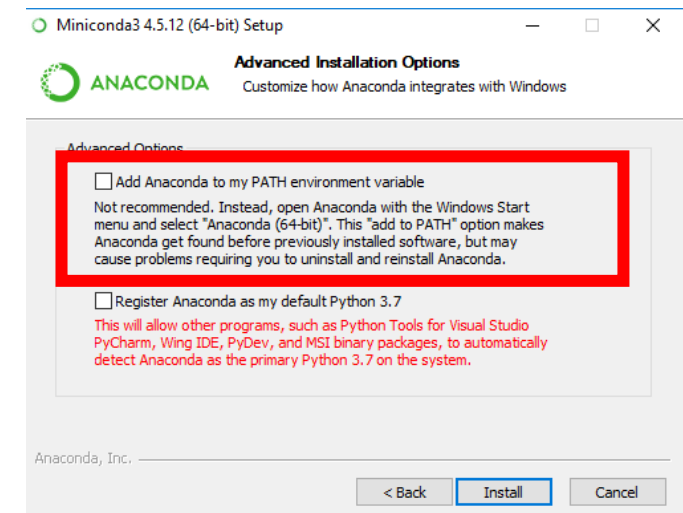


- **Installation:**
 - **Miniconda:** a free minimal installer for **conda**

an open source **package management system** that quickly **installs, runs and updates** packages and their dependencies. Runs on **Windows, macOS and linux**.

<https://docs.conda.io/en/latest/miniconda.html>

- **Find your Operating System**
- **Download Python 3.8**
- **Install it following the steps and make sure to check:**



Setting up Python

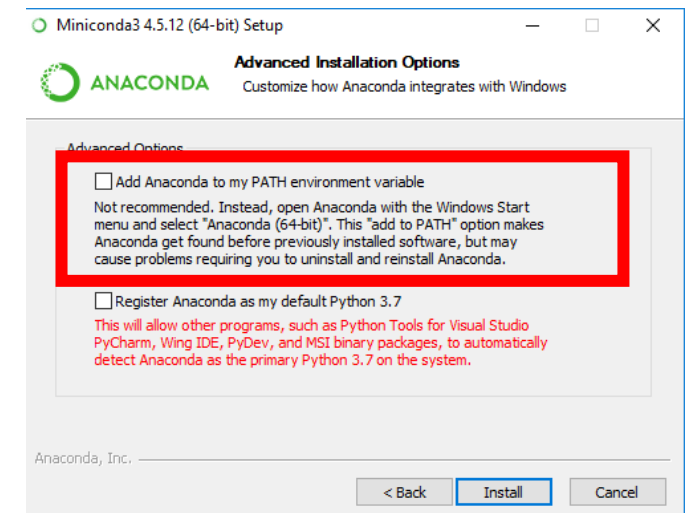


- **Installation:**
 - **Miniconda:** a free minimal installer for **conda**

an open source **package management system** that quickly **installs, runs and updates** packages and their dependencies. Runs on **Windows, macOS and linux**.

<https://docs.conda.io/en/latest/miniconda.html>

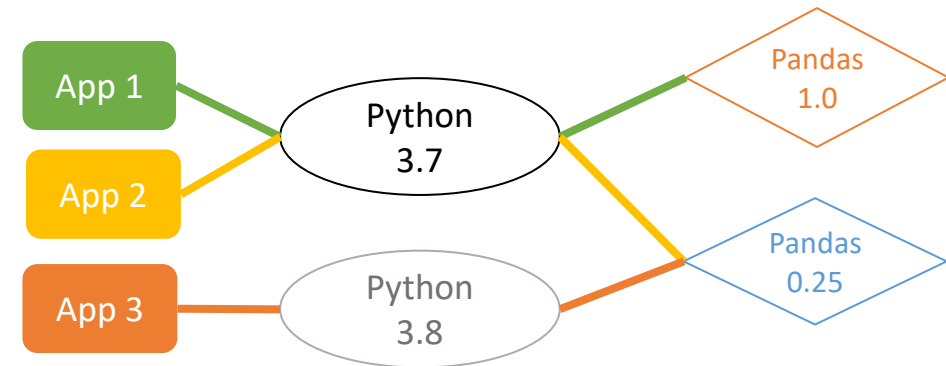
- **Find your Operating System**
- **Download Python 3.8**
- **Install it following the steps and make sure to check:**



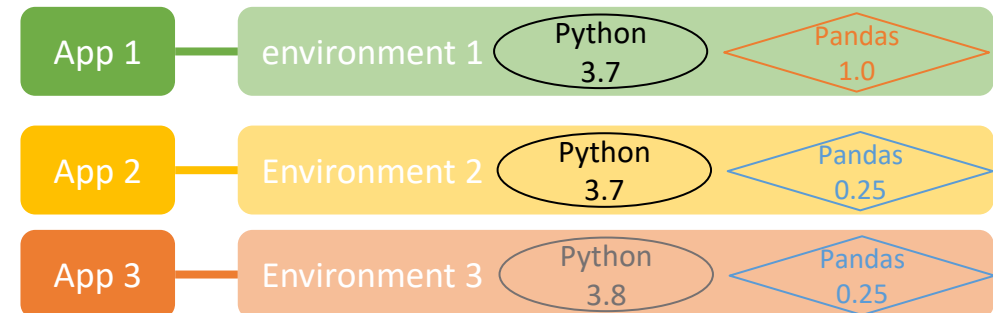
Virtual Environments



- **Python apps** will often use **packages and modules** that are **not part of the standard library**
- Apps sometimes need a **specific version of a library**
- This means, **one single Python installation may not meet the requirements of every application**



- The **solution** is to **create a virtual environment**, a self-contained directory that contains a **Python installation** (version) and the **necessary packages**



Virtual Environments



- Let's create a virtual environment:
 - To **create an environment** we use:
conda create -n name_env python==version
 - **Open a Terminal** window
 - Type *conda create -n myenv python==3.7* and 'y'

```
## Package Plan ##
environment location: C:\Users\sande\.conda\envs\myenv
added / updated specs:
- python==3.7

The following packages will be downloaded:

package | build | size
python-3.7.0 | hea74fb7_0 | 16.6 MB
Total: 16.6 MB

The following NEW packages will be INSTALLED:

certifi | pkgs/main/win-64::certifi-2020.12.5-py37haa95532_0
pip | pkgs/main/win-64::pip-21.0.1-py37haa95532_0
python | pkgs/main/win-64::python-3.7.0-hea74fb7_0
setuptools | pkgs/main/win-64::setuptools-52.0.8-py37haa95532_0
vc | pkgs/main/win-64::vc-14.2-h21ff451_1
vs2015_runtime | pkgs/main/win-64::vs2015_runtime-14.27.29016-h5e8377_2
wheel | pkgs/main/noarch::wheel-0.36.2-pyhd3eb1b0_0
wincertstore | pkgs/main/win-64::wincertstore-0.2-py37_0

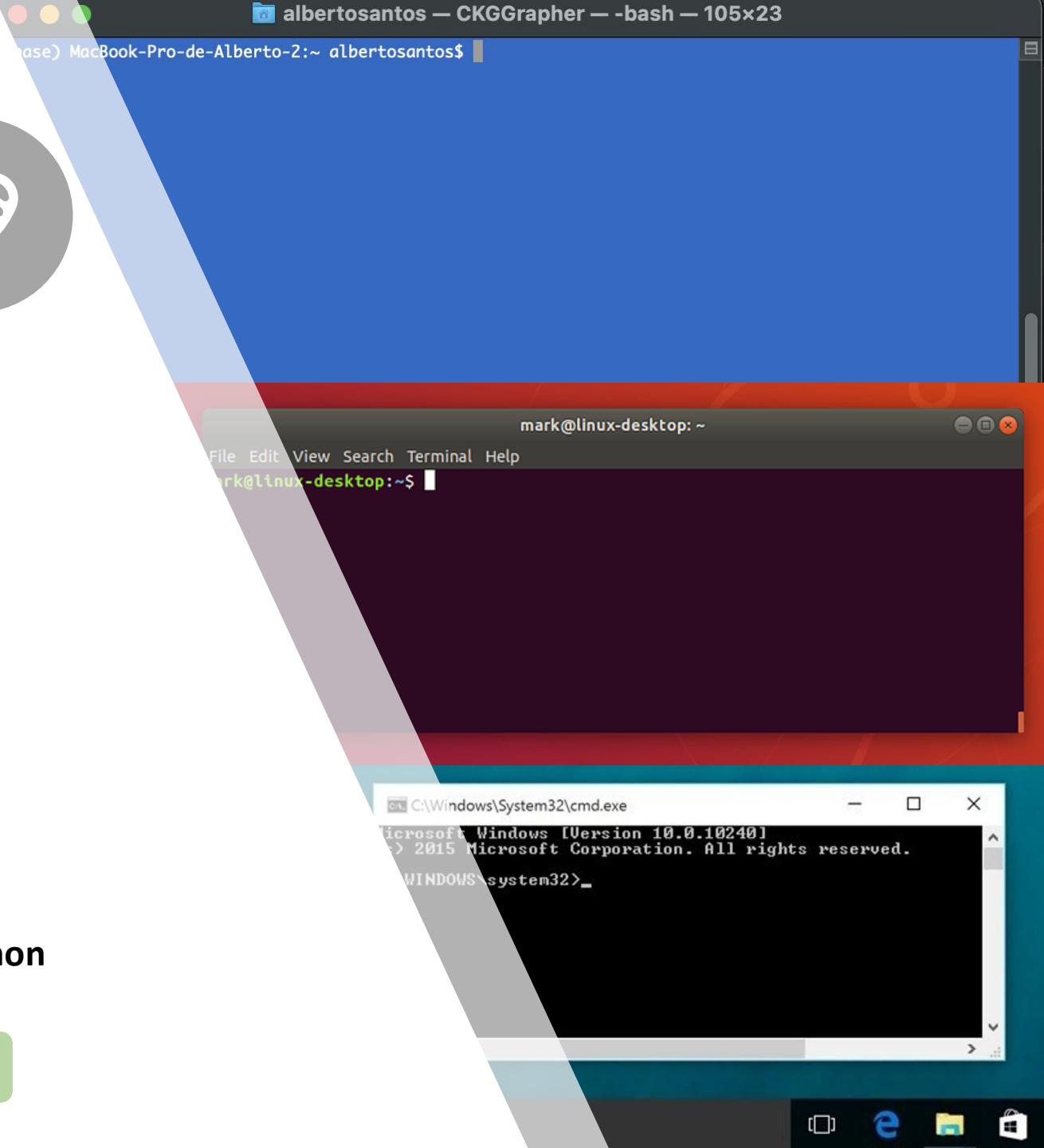
Proceed ([y]/n)?
```

- To **activate the environment** type:
conda activate myenv
- Now, if we type *python* we have the **right python version**

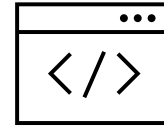
App 1

environment 1

Python
3.7



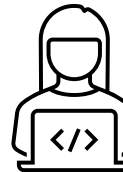
What tool do I use for coding?



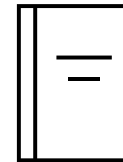
- **Terminal** → 2 options:
 - **Shell:** *python*
 - **Interactive shell:** *ipython*



- Simple **text editors**:
 - **Notepad++**
 - **Vim**



- **Integrated development environments (IDEs)** → Several options:
 - IDLE
 - PyCharm
 - Spyder
 - Visual Studio Code



- **Notebooks**

Jupyter Notebooks



The **Jupyter Notebook** is an **open-source application** to **create and share** documents that contain **code, equations, visualizations and text** (markdown)

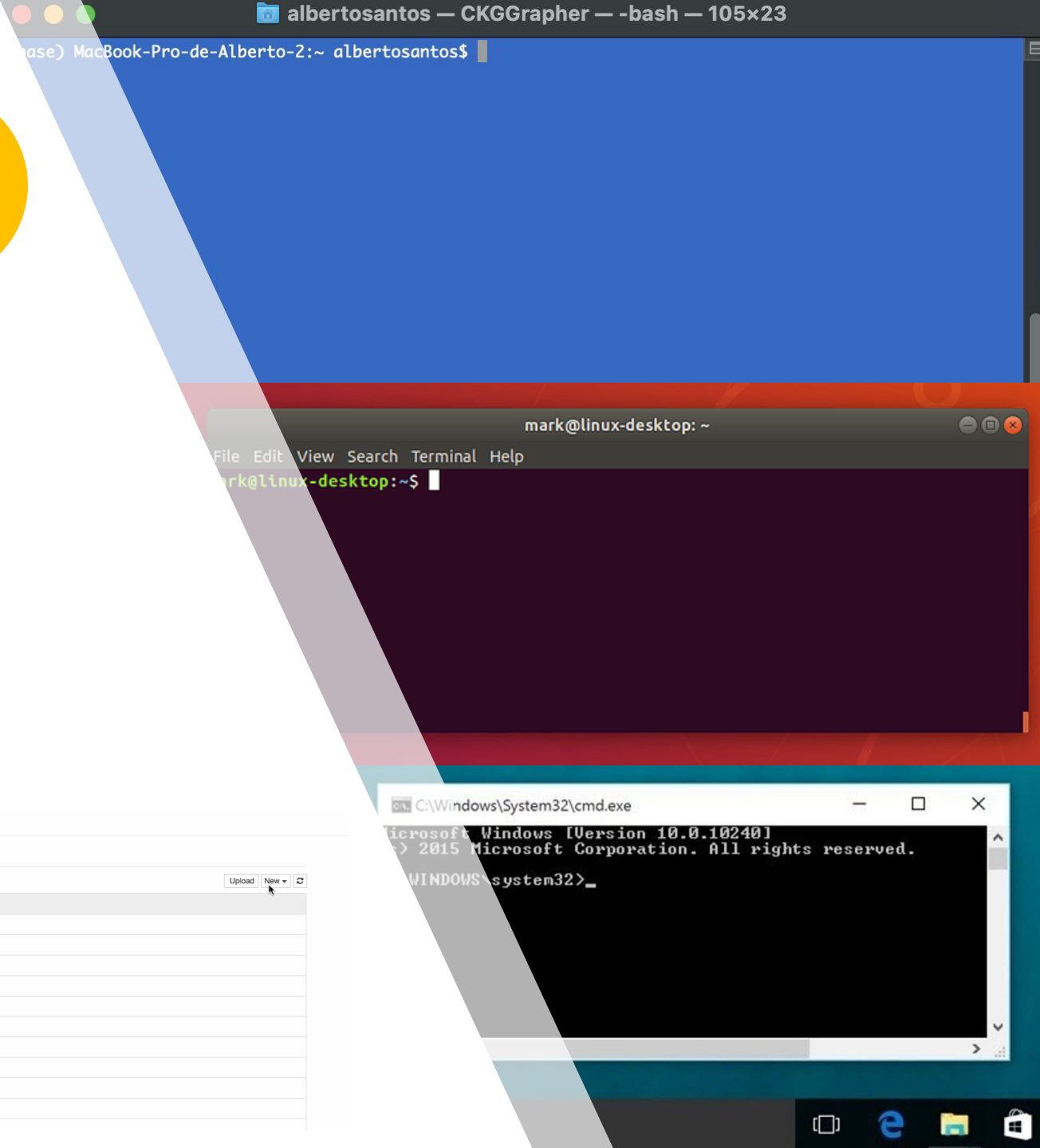
Jupyter Notebooks



- Let's install Jupyter Notebooks:
 - Open a Terminal window
 - Activate your environment:
conda activate myenv
 - Use *pip* to install Jupyter*:
pip install notebook
 - To start notebooks:
jupyter notebook



*This is where some OS may start complaining



Google Colab



- **Colab** (Colaboratory) is a tool to **write, execute** and **share python code** through the **browser**
- Colab is especially well suited to **machine learning, data analysis** and **education**.
- Colab is a **hosted Jupyter notebook service** that requires **no setup** to use, while providing **free access** to computing **resources** including GPUs
- It is **connected** to a **Google account** and **data** can be **accessed** through **Google Drive**.

<https://colab.research.google.com/>

Taking Control of Your Code



manuscript_draft.doc



manuscript_draft2.doc



manuscript_draft3.doc



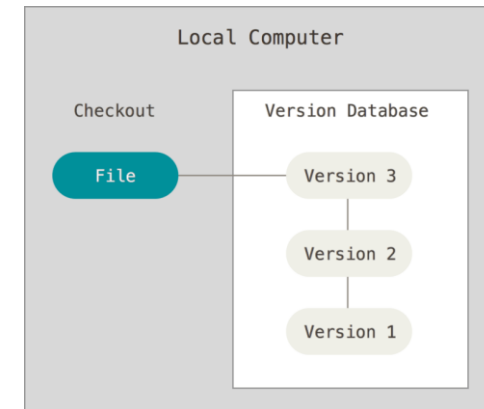
manuscript_final1.doc



manuscript_final33_mod_JADH_GGS_final44_reviewed_FH_2.2.1.doc

Version control

- It is a system that **records changes** to a file or set of files over time so that you can **recall specific versions** (history)
- **Important** in coding, especially when the code is accessed by “many” people (>1)
- A good way to **share your code** with others



Git is a software for **tracking changes** and **coordinating** work among collaborators developing code



GitHub facilitates access to Git's functionality

<https://github.com/Center-for-Health-Data-Science/PythonTsunami>

Recommended



Workshop

How we learned to stop worrying and love the git



