

Software Setup

Folder structure for the course

- You should create a **specific folder** for this course under your user folder. Under Windows a typical location could be something like (the ellipsis represents possible intermediate folders):

```
C:/Users/your_user_folder/.../MasterBD/FMAD/
```

while for MacOS users you should use something like:

```
/Users/your_user_folder/.../MasterBD/FMAD/
```

and finally for Linux it would be:

```
/home/your_user_folder/.../MasterBD/FMAD/
```

- **The folder that you have created will always be referred to in these notes as the Course Folder.** In the above examples we named it `FMAD` but you can give it any name you want. As usual, it is a good idea to keep it short and meaningful (and free of spaces, unusual characters, etc.)

- We will be using Git and [GitHub](#) repositories for the Course Notes and Assignments. In particular, the following repository will be used to share all the common course material:

<https://github.com/mbdfmad/fmad2223>

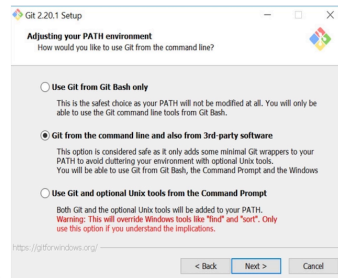
In the next section we will tell you how to get a local copy of this repository to your computer. For the time being you can browse the contents of the repository, but do not download it yet! This repository **will always be referred to in these notes as the Common Repository**. Soon you will be also using your *personal repository* and it is important to get the distinction and name conventions right from the beginning.

Windows computers

Installing Git

Important Note: If you already have a working and recent setup of Git (current version is 2.37), skip to the [Installing Anaconda Python](#) subsection.

- Install [Git for Windows](#) using this link. When asked about "Adjusting your PATH environment", select *Git from the command line and also from 3rd-party*

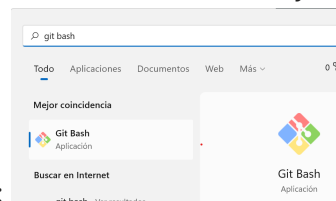


software.

the defaults.

For all the rest of the setup it is ok to accept

- When the Git setup is completed hit the Windows key and begin typing *Git Bash*.



You should see results like this:

Open the *Git Bash* application. A new *terminal* window will appear with a prompt waiting for your commands.



The details of the prompt (user and machine names) will be different in your case, of course. We will return here soon, but first we need a GitHub account.

Create a GitHub account

- **Important Note:** even if you already have a GitHub account you need to have an account that is linked with your university email (the one ending in `@alu.comillas.edu`). You can create that account using [this link](#). Make sure to log out first if you were already logged with a *non-university-associated* account. And we recommend [reading this advice first](#).


Clone the *common repository*

- Using the *Git Bash terminal* navigate to the *Course Folder*. You can get there using the `cd` command. **Make sure that you have changed into the right folder before going further!!** You can use the `pwd` command to check your current location. Ask for help if you run into trouble.

- In that folder run the following command:

```
git clone https://github.com/mbdfmad/fmad2223.git
```

Git will begin cloning (downloading) the contents of the repo to a folder named `fmad2223` in your computer. The process looks like this (the number of cloned objects will surely be greater in your case):



```
MINGW64 ~/Course/fmad
fernandezarapito MINGW64 ~
$ cd fmad/
fernandezarapito MINGW64 ~/fmad
$ git clone https://github.com/mbdfmad/fmad2223.git
Cloning into 'fmad2223'...
remote: Enumerating objects: 20, done.
remote: Counting objects: 100% (20/20), done.
remote: Compressing objects: 100% (19/19), done.
remote: Total 20 (delta 1), reused 16 (delta 1), pack-reused 0
Receiving objects: 100% (20/20), 302.37 KiB | 962.00 KiB/s, done.
Resolving deltas: 100% (1/1), done.
```

The `fmad2223` folder will then be a subfolder of the *Course Folder*. You should consider this `fmad2223` folder as a *read-only* folder. You will soon see how to update its contents as the course progresses, but **you are not expected to directly modify the contents of the `fmad2223` folder** (and you shouldn't). You can *copy* any object from this folder (e.g. the code examples that we will provide) to other locations in your computer if you want to make changes. Finally, if you are a seasoned Git user you can opt for a fork of the repo instead of cloning it.

Keeping your local copy of the *common repository* updated.

- If you are used to auto syncing tools such as Dropbox, OneDrive, iCloud Drive and the like, you must keep in mind that **Git repos are not auto-syncing**. To keep the information in your repos updated you have to manually update them (both ways, from your computer to GitHub and viceversa).
- **This will be very important for our workflow in the lectures of the course.** So make sure to get it right, and ask for help soon if you need it. Every work session for this course will begin with the following ritual:
 - Open a *Terminal* (the Git Bash terminal or, see below, the Anaconda terminal are both ok).
 - Navigate to your local `fmad2223` folder. Do not confuse this with the *Course Folder*.
 - Run `git status` to check that everything is ok.
 - Run `git pull origin main`. Depending on the update status of your local copy of the repo you will either download the new contents from the repo at GitHub or you will receive a message saying that you are **Already up to date**.

Don't worry too much, however. You cannot break anything, since you won't have write permissions for this GitHub repo. Worst case scenario, you can (backup

your local files and) delete the `fmad2223` folder and clone it again to get the latest contents.

Setting up a local Git configuration for the Common Repository

- Using still the *Git Bash* terminal, change into the `fmad2223` folder and set your local Git configuration with these commands (the first one is the folder change from the *Course Folder* to `fmad2223`):

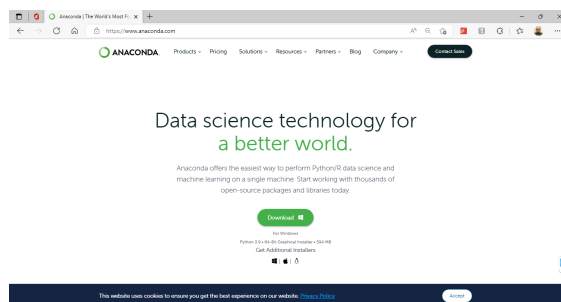
```
cd fmad2223
git config --local user.name 'replace_this_with_your_GitHub_username'
git config --local user.email 'replace_this_with_your_university_email'
git config --global init.defaultBranch main
```

- By the way, if you are doing this setup for the first time, now you can close the *Git Bash* terminal window.

Installing Anaconda Python

Important Note: If you already have a working setup of Anaconda Python (with Python 3.8 or 3.9) then skip to the [Create a Conda Environment for the Course](#) subsection. If your Python version is older: get updated, people!

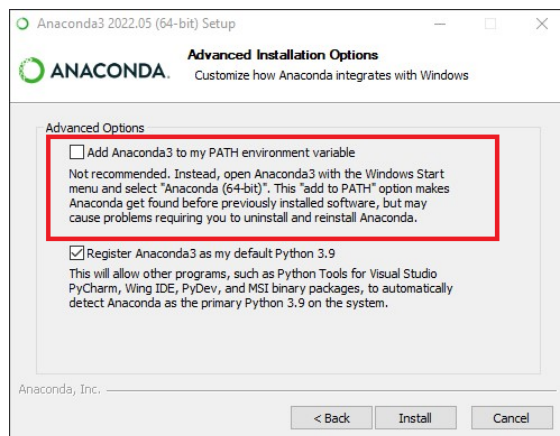
- Begin by navigating to the [Anaconda Web Page](#):



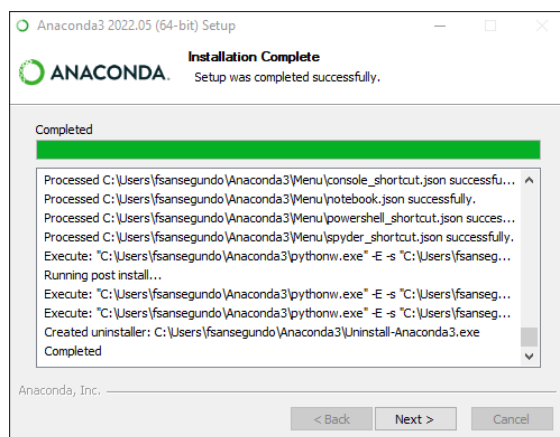
- Click the download link. When the download finishes, run the Anaconda Installer.



- In the following setup dialogs you can (and probably should) accept all the defaults. **Be careful with this step!** Unless you have a good reason not to, please leave the first box unmarked, as illustrated here:



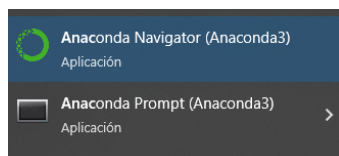
- When setup is completed you will see something like this:



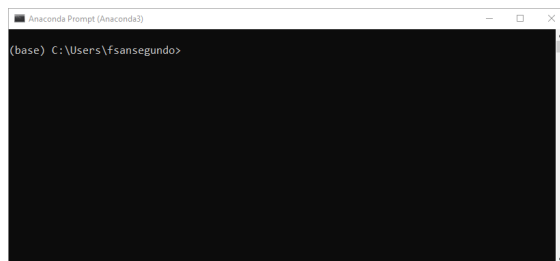
Click on *Next* and do the same in the final dialogs (consisting in software ads and tutorials, we suggest to skip all of them).

Create a Conda Environment for the Course

- A conda environment allows us to organize our work in projects, keeping each project dependencies isolated from the rest of your work. It is also the best way to avoid running into the potentially catastrophic situation in where you have several Python versions and they conflict to [create a mess](#). We urge you to **always use environments** to organize your work in Python. You can learn more about the use of Conda environments [in this link](#). And to learn more about environments in Python in general, you can check these two links:
 - [A Guide to Python's Virtual Environments](#)
 - [The Definitive Guide to Conda Environments](#)
- We will be using a Conda environment named `fmad` for this course. To create the environment we will use the *Anaconda Terminal (or Anaconda Prompt)*. Hit the Windows key and begin typing *Anaconda*. You should see results like this:



The Anaconda Navigator is a GUI interface to manage your Anaconda setup. Some people like it a lot, but we will be using mostly the (old school) terminal-based command-line approach. Thus, start the *Anaconda Prompt* and a window like this one will open:



This terminal window is where most of our *Python management* will occur (but not the actual coding!)

- In that prompt execute the command

```
conda create --name fmad
```

and after some moments something like this should happen

```

Anaconda Prompt (Anaconda3) - conda create --name fmad
(base) C:\Users\fsanseguno>conda create --name fmad
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 4.12.0
  latest version: 4.13.0

Please update conda by running

    $ conda update -n base -c defaults conda

## Package Plan ##

  environment location: C:\Users\fsanseguno\Anaconda3\envs\fmad

Proceed ([y]/n)?

```

For the time being ignore the warning about updating Conda it it appears (we will deal with it in the next step) and just answer *y* and hit *Enter*. A successful environment creation looks like:

```

Proceed ([y]/n)? y
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate fmad
#
# To deactivate an active environment, use
#
#     $ conda deactivate
#
(base) C:\Users\fsanseguno>

```

Update conda

- To update conda (if the warning appeared in the previous step) run this command:

```
conda update -n base -c defaults conda
```

As the following picture illustrates conda will present you with a summary of the update operation. Answer *y* and hit *Enter*.

```

Anaconda Prompt (Anaconda3) - conda install --name fmad numpy - conda update -n base -c defaults conda
(fmad) C:\Users\fsanseguno>conda update -n base -c defaults conda
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\Users\fsanseguno\Anaconda3

added / updated specs:
- conda

The following packages will be downloaded:

package | build | size
-----|-----|-----
conda-4.13.0 | py39h8a95532_0 | 923 KB
pyjwt-2.4.0 | py39h8a95532_0 | 38 KB
-----|-----|-----
Total: | | 961 KB

The following packages will be UPDATED:

conda 4.12.0-py39h8a95532_0 --> 4.13.0-py39h8a95532_0
pyjwt 2.1.0-py39h8a95532_0 --> 2.4.0-py39h8a95532_0

Proceed ([y]/n)? y

```

Then conda will download the required software to autoupdate. The expected result looks like this:

```

Total: 961 KB

The following packages will be UPDATED:

conda                  4.12.0-py39haa95532_0 --> 4.13.0-py39haa95532_0
pyjwt                  2.1.0-py39haa95532_0 --> 2.4.0-py39haa95532_0

Proceed ([y]/n)? y

Downloading and Extracting Packages
pyjwt-2.4.0                  38 KB | ##### | 100%
conda-4.13.0                 923 KB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
(fmad) C:\Users\fsansegundo>

```

Activate the environment

- When we created the `fmad` environment in a previous step, Conda showed us the activation command:

```
conda activate fmad
```

Run this command now. This will activate the newly created environment `fmad`. Note that the name of the active environment appears in parenthesis at the beginning of the prompt line:

```

(base) C:\Users\fsansegundo>conda activate fmad
(fmad) C:\Users\fsansegundo>

```

Make sure that the environment is correctly activated before proceeding!! If you have any problem ask for help.

Install the first library

- The first library that we will install is [NumPy](#), which provides the foundation for much of the work we will doing in this course. We will discuss NumPy in coming sessions of the course. But to use it we first need to install it in our `fmad` environment. Execute this command now:

```
conda install --name fmad numpy
```

A summary of the libraries that need to be downloaded for the NumPy setup will appear

```

(fmad) C:\Users\fsansegundo>conda install --name fmad numpy
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\Users\fsansegundo\Anaconda3\envs\fmad

added / updated specs:
- numpy

The following packages will be downloaded:

package                                     build
-----
ca-certificates-2022.4.26                  haa95532_0 124 KB
certifi-2022.6.15                          py310haa95532_0 153 KB
libffi-3.4.2                               hd77b12b_4 107 KB
mkl-service-2.4.0                          py310h2bfff1b_0 48 KB
mkl_fft-1.3.1                              py310ha0764ee_0 136 KB
mkl_random-1.2.2                           py3104e4d8f06_0 221 KB
numpy-1.22.3                               py310h6d2d95c_0 25 KB
numpy-base-1.22.3                         py310h06c741_0 4.9 MB
openssl-1.1.1a                             h2bfff1b_0 4.8 MB
pip-22.1.2                                 py310haa95532_0 2.5 MB
python-3.10.4                              hbb2fff3_0 15.9 MB
setuptools-61.2.0                          py310haa95532_0 1.8 MB
sqlite-3.38.5                              h2bfff1b_0 798 KB
tk-8.6.12                                  h2bfff1b_0 3.1 MB
wincontstore-0.2                           py310haa95532_2 15 KB
xz-5.2.5                                    h8cc29b1_1 246 KB

```


followed by a rather lengthy list of the libraries about to be installed (in most cases these two lists are almost equal)

```

Anaconda Prompt (baseconda) - conda install --name fmad numpy - conda update --base -c defaults conda - conda install --name fmad numpy
ca-certificates pkgs/main/win-64::ca-certificates-2022.4.26-haa95532_0
certifi pkgs/main/win-64::certifi-2022.6.15-py310haa95532_0
intel-openmp pkgs/main/win-64::intel-openmp-2021.4.0-haa95532_3556
libffi pkgs/main/win-64::libffi-3.4.2-hd77b12b_4
mkl pkgs/main/win-64::mkl-2021.4.0-haa95532_640
mkl-service pkgs/main/win-64::mkl-service-2.4.0-py310ha0764ee_0
mkl_fft pkgs/main/win-64::mkl_fft-1.3.1-py310ha0764ee_0
mkl_random pkgs/main/win-64::mkl_random-1.2.2-py310ha0764ee_0
numpy pkgs/main/win-64::numpy-1.22.3-py310ha0764ee_0
numpy-base pkgs/main/win-64::numpy-base-1.22.3-py310ha0764ee_0
openssl pkgs/main/win-64::openssl-1.1.1q-h2b6ff1b_0
pip pkgs/main/win-64::pip-22.1.2-py310haa95532_0
python pkgs/main/win-64::python-3.10.4-h2b6ff1b_0
setuptools pkgs/main/win-64::setuptools-61.2.0-py310haa95532_0
six pkgs/main/noarch::six-1.16.0-pyhd3eb1b0_1
sqlite pkgs/main/win-64::sqlite-3.38.5-h2b6ff1b_0
tk pkgs/main/win-64::tk-8.6.12-h2b6ff1b_0
tzdata pkgs/main/noarch::tzdata-2022a-hda174b7_0
vc pkgs/main/win-64::vc-14.2-h21ff451_1
vs2015_runtime pkgs/main/win-64::vs2015_runtime-14.22.29016-h5e58377_2
wheel pkgs/main/noarch::wheel-0.37.1-pyhd3eb1b0_0
wincertstore pkgs/main/win-64::wincertstore-0.2-py310haa95532_2
xz pkgs/main/win-64::xz-5.2.5-h8cc25b1_1
zlib pkgs/main/win-64::zlib-1.2.12-h8cc25b1_2

Proceed ([y]/n)?

```

Answer yes to make the setup begin and wait patiently while NumPy and all of its requirements are downloaded and installed. The process should end like this:

```

Downloading and Extracting Packages
ca-certificates-2022 124 KB ##### 100%
certifi-2022.6.15 1.0 MB ##### 100%
intel-openmp-2021.4.0 4.9 MB ##### 100%
libffi-3.4.2 2.5 MB ##### 100%
mkl-2021.4.0 221 KB ##### 100%
mkl-service-2.4.0 136 KB ##### 100%
mkl_fft-1.3.1 25 KB ##### 100%
mkl_random-1.2.2 153 KB ##### 100%
numpy-1.22.3 15.9 MB ##### 100%
numpy-base-1.22.3 4.8 MB ##### 100%
openssl-1.1.1q 48 KB ##### 100%
pip-22.1.2 246 KB ##### 100%
python-3.10.4 780 KB ##### 100%
setuptools-61.2.0 15.9 MB ##### 100%
six-1.16.0 107 KB ##### 100%
sqlite-3.38.5 107 KB ##### 100%
tk-8.6.12 107 KB ##### 100%
tzdata-2022a 107 KB ##### 100%
vc-14.2 107 KB ##### 100%
vs2015_runtime-14.22.29016 107 KB ##### 100%
wheel-0.37.1 107 KB ##### 100%
wincertstore-0.2 107 KB ##### 100%
xz-5.2.5 107 KB ##### 100%
zlib-1.2.12 107 KB ##### 100%

Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(fmad) C:\Users\fsansgundo>

```

Install additional libraries

- NumPy is just the first of the libraries that we will use in the course, but many others will also come into play. To name a few of the most relevant:
 - NumPy
 - Pandas
 - Matplotlib
 - Scikit-Learn

Take a few moments to visit the homepage for each of these libraries.

Exercise: install all these libraries to the `fmad` environment. To install several libraries you don't need to repeat `conda install` for each library; you can use a single install command and separate the names of the libraries with a space.

Install the Jupyter Notebook & Jupyter Lab

- In this course we will talk to Python using the onterface provided by *Jupyter Notebooks*. The [Jupyter Project](#) provides a general purpose development environment which is well suited for Python (but not limited to it). It will play a

central role in our work with Python, but it installs (to the `fmad` environment) just like any other library.

```
conda install --name fmad jupyter
```

- When the setup ends execute the following two commands in order. The first one changes your current directory to your user folder in Windows. The second command opens the Jupyter Notebook in your *default Web Browser*:

```
cd %HOMEPATH%
jupyter notebook &
```

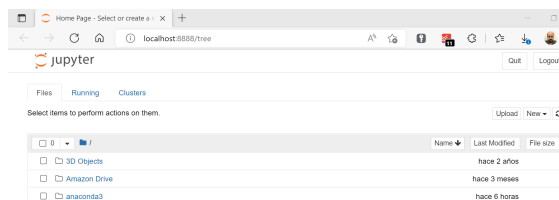
Like this:

```
C:\Users\ferma> jupyter notebook &
[16:35:11.429 NotebookApp] Serving notebooks from local directory: C:\Users\ferma
[16:35:11.429 NotebookApp] 0.0.0.0
[16:35:11.429 NotebookApp] http://localhost:8888/?token=7806ec1dc9a5e17a32a0e098a8ff1889f
[16:35:11.438 NotebookApp] or http://127.0.0.1:8888/?token=7806ec1dc9a5e17a32a0e098a8ff1889f
[16:35:11.438 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[16:35:11.468 NotebookApp]

To access the notebook, open this file in a browser:
file:///C:/Users/ferma/AppData/Local/jupyter/runtime/observer-15260-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=7806ec1dc9a5e17a32a0e098a8ff1889f
or http://127.0.0.1:8888/?token=7806ec1dc9a5e17a32a0e098a8ff1889f
```

Note: We recommend to always start Jupyter from a folder high enough in your folder structure: you won't be able to easily move up from the starting point later. Make sure, however, that you are at the very least, at the *Course folder* level.

In a few moments your default *Web Browser* will open to a new page that shows the entry point for the running Jupyter server that we have just started:



You can use that page to navigate your folder structure until you get to the *Course Folder*. Once you are seeing the contents of that folder,

In []:

MacOS computers

- MacOS includes a *system version* of Python. We will not be using that version but we need to make sure that it does not interfere with our work and that the basic functioning of your computer is not affected. Always keep recent backups of your data before any complex setup like this!
-

Linux computers

- Most recent versions of Linux already come

```
In [1]: pip install nbconvert[webpdf]
zsh:1: no matches found: nbconvert[webpdf]
```

```
In [ ]:
```