Get ready for Lauenburg

The aim of this guide is to help you set up a jupyter notebook on your laptop. Links to different methods are presented. Feel free to approach this problem in any other way, as long as you will have a jupyter notebook ready for the days in Lauenburg.

✏️ To do

□ Install a conda package manager

□ Install jupyter

□ Install the essential packages

□ Familiarize yourself with **`git`**

□ (Familiarize yourself with **`vim`**)

🐍 Anaconda

Anaconda is a python distribution based on the conda package manager. Follow the install guide provided by the project. Ready made installers are provided for the most common operating systems. There is also a lightweight version, called miniconda. but if you are new to this, go with anaconda.

For users of **Homebrew**:
There might be issues with the parallel usage of **`conda`** and **`brew`**. They can be solved, though.

For users of a **new MacBook**:
If you have an M1 processor, there might be issues because it has another architecture than most common processors. You might want to look into these articles describing work arounds: 1, 2, 3.

📑 Notebook / Lab

To install jupyter notebook

```
conda create -n <name>
conda activate <name>
conda install jupyterlab
```

and run with:

```
jupyter notebook
```

Alternatively use the command **`jupyter lab`** to start the newer, more IDE-like user interface that combines the notebook with a file browser and other elements. Most of you might finding themselves spending most of their time working with a notebook hosted on a HPC system (jupyterhub.dkrz.de). Nevertheless it is useful to have a local notebook set up. Levante is not always available.

📦 Packages

You just installed a very handy package manager: **`conda`**. There is another one, which you might already know: **`pip`**. **`conda`** does a bit more and it it thus preferred to use **`conda`**. If a package is only available via pip, use pip. Installing a package is straightforward. If you know the package name, just try the following (here with numpy):

```
conda install numpy
```

The essential packages for the weekend will be: **`numpy`**, **`matplotlib`** and **`xarray`**. If you are already addicted to python packages, these are the ones we will use in examples: **`intake`**, **`pandas`**, **`scipy`**, **`datashader`**, **`dask`** and **`cartopy`**.

🛠️ Additional tools

- **IDE**: An integrated development environment (IDE) is a fancy text editor with many added features for software development. It usually has build automation tools and a debugger. The community (free) edition of PyCharm is a powerful IDE for Python. You can install packages through the IDE and it has many plugins (e.g. *CodeTogether* for collaborative coding).

- **git**: is software for tracking changes in any set of files. Even if you never make a branch, git is still useful, e.g. to make backups of your code. A github account can help you synchronise your notebooks over various machines. The DKRZ also provides a git system. If you have no clue what git is, go through some basic tutorial.

- **text editor**: sooner or later you will find yourself wanting to edit a file in the terminal. **`nano`** is an easy editor, but others are much more convenient once you know how to use them. it may be worthwhile to familiarize yourself with **`vim`** a little. Try the build-in **`vimtutor`** command in your terminal, or maybe a game.