LaboratoryOfComputationalPhysics

Notebooks guiding students through the world of data analysis with python.

This repo should be forked by each individual student. Exercises should be committed to the student's repo and notified to the professor by a pull request.
Such pull request should be made on this remote repo under the corresponding student branch (Dedicated branches will indeed be created in due time).

IPython notebooks instructions and tips
Notebooks are extremely powerful tools, you may find useful to discover some of their functionalities on this tutorial [page](https://nbviewer.jupyter.org/github/ipython/ipython/blob/3.x/examples/Notebook/Index.ipynb) or on this by checking this [list](https://www.dataquest.io/blog/jupyter-notebook-tips-tricks-shortcuts/) of tips

Git Instructions

To start with, you need to have a github account. If you don't have one, go to [github](github.com) and follow instructions on how to create it.

Suggestion: use a reasonable username that resembles your actual name.

Once you have your github, as first thing fork this repository, i.e. go [there](https://github.com/mzanetti79/LaboratoryOfComputationalPhysics) and click on the top-right button *fork*

Setting up a local repository

What follows needs to be done any time a new local repository is created. In particular, if you are working in a location where such repo already exist, what follows doesn't need to be repeated.
  * Clone your (forked) repository (i.e. create a local repository cloned from the remote one)

`git clone https://github.com/YOUR_GIT_ACCOUNT/LaboratoryOfComputationalPhysics.git`

 where YOUR_GIT_ACCOUNT it your account on github. Now you can get to your local working folder:

 `cd LaboratoryOfComputationalPhysics/`

   * Configure your username and email:

`git config --global user.name "YOUR_GIT_ACCOUNT"`

`git config --global user.email "YOUR_EMAIL_ADDRESS"`

(you must have understood what capital-letters-words stand for). Your git configuration is stored in `.gitconfig`, a file that you can alwasy edit by hand or via the `git config ..` commands.

* Define mzanetti79's repo as the upstream repository (you may need to set the url too), check that actually succeeded and get (fetch) the updates that have been done on the remote repository:

`git remote add upstream https://github.com/mzanetti79/LaboratoryOfComputationalPhysics.git`

`git remote set-url origin https://YOUR_GIT_ACCOUNT@github.com/YOUR_GIT_ACCOUNT/LaboratoryOfComputationalPhysics.git`

`git remote -v`

`git fetch upstream`

  * The default branch is `master`, you should now create your development branch where to play and exercise with the code. Note that however you have a branch corresponding to you (name_surname) in the upstream repository (`upstream/name_surname`): that is the branch you should point the pull request to. In order to set up a proper development cycle, you must create a branch (in the example below called TRACKING_BRANCH_NAME) that *tracks* `upstream/name_surname`:

`git branch -vv`

`git checkout -b TRACKING_BRANCH_NAME upstream/name_surname`

Note that the case you decide do make your development in a branch that does NOT track `upstream/name_surname`, you'll eventually need to merge your changes into the branch tracking `upstream/name_surname` which is the one you'll make the pull request for (see later).

Standard development cycle

  * Before starting with the development you could check whether the orginal repository (mzanetti79's one) have been updated with respect to your forked version (that's likely to be the case prior to every lab class). If it had, then merge the chances into your master.

  `git fetch upstream`

  `git checkout master`

  `git merge upstream/master`

  The idea is that your master always reflects `upstream/master`, i.e. it keeps a local copy of the reference code as a starting point for your developments (i.e. solving the assigned problems).
  Note that in order to update your repository on github, you need to push the local version (see later).

  * In the case a pull request of yours to mzanetti79 has been recently approved, you also need to synch your development branch:

  `git checkout TRACKING_BRANCH_NAME`

  `git merge upstream/name_surname`

  * You may also need to get the updates from the master, i.e. need to merge the master:

  `git merge master`

  * Now do the real stuff, i.e. developing some code. Image you create a NEW_FILE. Add the file to your local repository and stages it for commit (To unstage a file, use 'git reset HEAD NEW_FILE)'

  `git add NEW_FILE`

  * Commits the (tracked) changes you made to the file and prepares them to be pushed to your remote repository on github

  `git commit -m "Add existing file"`

(what follows after `-m` is a comment to later remind what was that commit about)

 * Now you want to propagate (push) your local changes to your remote repository on github (`origin`)

 `git push origin TRACKING_BRANCH_NAME`

 * Finally you may want to propagate your development also to the repo you originally forked from, i.e. mzanetti79's one (this is likely to happen anytime you'll be asked to deliver your homework!). For that you need to go for a "pull request", which is done from github itself. Be carefull to point your pull request to `mzanetti79/name_surname`