Complexity talk:

* pseudo code may be hard to read and follow, lots of code and indentation missing
* we might need some smoother transition from Git intro to complexity talk, it is quite a step
* Mention on a very general level what \_\_init\_\_.py does and is when talking about structure
* Use from somelib import \* as an example, import somelib is valid use of many libraries
* The BDFL endorsed way to do loop vs map/filter would be list comperehensions

Screen is super important, two screens are very important. Screen is too small and only one. We need to insist on rooms with large and ideally two screens.

Jupyter notebooks:

* Excellent lesson and presentation but it became a data carpentry lesson. Which is perhaps good :-)
* IMO (Radovan) we could show more examples in which one would want to use notebooks. In other words the question is what is the goal of the Jupyter lesson really. But let’s wait for the feedback. It looks like people like it and let us not change a winning team.

Git collaborative:

* Radovan: I think we should brush over or skip bare/non-bare. I think it is not necessary to understand them for real life of most researchers and they might really confuse beginners.
* Talk has at times skipped essentials (merge commits when you git pull) and went into non-essential side-roads (git rebase).
* Jyry: an example repo on which to demonstrate things would be optimal
* A travis check for tweet lengths for <https://github.com/coderefinery/forking-workflow-exercise> would concretize the place where tests are usually done in a pr workflow

General notes:

* Manage your time, do not insist that we traverse every line of our material if we are completely over time
* We need a musical instrument to announce end of breaks

General notes on changing instructors in a lesson:

* We need documentation.
* Old and new instructor need to talk about it and single out what is essential to make sure the focus is not lost.

Testing:

* Took too long to arrive at first hands-on, it is hard to keep attention for more than 30 mins

IDE:

* In the IDE session we show unittest but in the testing session we use pytest. Radovan: IMO pytest is simpler so perhaps we can streamline this.

Documenting:

* Jyry: The talk was quite high-level, perhaps a few words about concrete practices
* Could we have an example function with a documented version and an undocumented version? Something classic like Newtonian motion with reference to the original Principia