## Unknown session

* Different packages, software versions etc. A bit consistency in working environment would be nice, e.g. virtual machine
* Thanks for the Workshop! It was great, I learned a lot. However problems with having some functions working on Windows, and some content was too fast for following. I will look with more calm at the material later
* Had a great time at the workshop and hope to implement some of it into my work. Thank you very much.

## Git intro

* Good to have enough time planned in so that everybody can get help and catch up
* Very good that you used an example all can understand (txt file)
* Good that you started from zero
* Nice idea to practice through an example
* At some points a bit fast (but I am still a beginner...)
* Bad branch illustration
* Relevant content
* Some parts a bit fast on the explanation but it is compensated by the text material and help from the staff
* Very well organized
* Good structure
* Goot examples
* Online documentation makes it easy to follow / catch up
* Easy to understand example
* Short intro to different editors and basic commands (quit etc) would help to complete newbies
* Good introduction to repositories
* Accessible to all skill levels
* A bit too slow at times (sometimes due to people having issues, which is understandable)
* Great course! Good to have it interactive
* Good introduction to git
* Typing a bit fast sometimes - hard to follow
* Good presentation
* Perhaps provide a docker as backup solution
* Great entry-level instructions!
* Regarding online course materials:
  + Perhaps a quick explanation of what each thing we are installing is
  + Start with quick intro on how to fork course notes on GitHub so we can take notes during the session
* Great course so far! Very good lecturer, and I got the help I needed
* Good pace
* Can’t think of any criticism now
* Very detailed, easy to follow and interactive, with practice examples
* Some confusion with text editors in the beginning. Maybe some general information about this and shortcuts in the command line at the beginning will make things simpler and save time

## Documentation

* Was good
* Perhaps give some example repos for what to include in a research doc, setup, …
* Very important topic
* It was a little too fast
* It could also be nice to include a little general advice on what to document and how
* The github pages was new to me as well as the jekyll/etc services, which are very useful for me
* Adaptive and interactive teaching
* Superb overview of different solutions (readthedocs + github pages)
* This section went a bit rushed and I was lost a few times
* Project website section went a bit fast for me
* Very helpful tips, though

## Git-collaborative

* Very detailed; good to have the groupwork session
* Would be nice to have all single steps included in the documentation because sometimes not everything is noted down
* I’ve used git before, but didn’t know much about forking. Good with insight!
* Visual representation
* Talk louder
* Well structured and easy to follow learning materials
* Good exercise
* Maybe you should use a “real” live example where changes are done on branches. Because I’m a bit confused if I should do pull requests on branches or master
* Collaborative example (everyone making pull requests) was fun and got the point across
* Remote repo exercise was a bit confusing -> maybe have step-by-step explanation of the required commands at the end as a “solution set”?¨
  + Making a local copy of remote repo was also a bit confusing, but well explained by instructor!
* This session was dense, maybe we could have had 2 breaks (since we had time to do so)
* The lecture was easier to follow than yesterday. I liked the more hands-on coding
* Patience in explaining -> thank you :)
* Find out how to install nano on bash
* Relevant content!
* More best practice examples

## IDE

* It seemed to only make sense if you had used it before
* Would be good to have each step written down explicitly somewhere when doing interactive programming. Now, if one gets behind 5 seconds and misses one command, one is lost
* Good with interactive coding
* Some words about IDEs for other languages and/or multi-lingual IDEs would be nice
* A bit more about “debugging for dummies”
* The pycharm lesson wsa hard to follow as someone that had never seen it before
* Requirements file and setting up env
* Too fast, i couldn’t follow what was going on on the screen
* A little bit hard to follow the explanations while doing it on own computer

## Reproducibility

* Sounds useful, will read more and try out...
* But coulldn’t get it to run in the session
* Great introduction to the thinking pattern of reproducible research
* Makefiles might be useful
* Docker and CWL seems way too involved and a struggle to set up that I won’t even try to set it up
* Variety of tools
* Often too fast and too little detail (did not quite get what make, docker and cwl are useful for);
* maybe better to not clear terminal so that people can follow previous code parts
* Interesting topic especially make
* Was lost half through containers
* Maybe have a windows computer for the presentation
* Mostly people having linux have more experience already
* Windows users are just lost sometimes looking at the screen
* I couldn’t follow due to system (no admin rights, couldn’t get docker or cat+python to work)
* Content was a bit advanced, perhaps taking some more time to set the scene (what is Docker, CWL, what are the differences between these) - especially in context of what we’ve learned in research process
* Very insightful explanation of workflow tools and how to structure to make work reproducible
* Docker is still a bit mysterious to me. Not sure how I can/should use it
* In general very interesting and well explained,
* But sometimes a bit harder to follow than yesterday because of technical issues
* Very useful presentation of makefiles
* Docker: not possible to make it work on Windows (daemon not running)
* CWL: maybe too advanced for my needs

## Git archeology

* Very useful!
* Fun to learn new things in py scripts
* Very useful!
* Good with 1st exercise, exercise simple enough to get the point!
* Interesting and useful stuff!
* Really useful lesson and exercise!
* In general very interesting and well explained, b

## 

## Testing

* Very useful!
* Good notes and exercises
* An indentation error in the exercise when uncommenting in no. 9
* Very useful topic and exercise
* Maybe useful to go a bit more in depth into what Travis and Coveralls do? Even quickly go over the test code perhaps so we see what is going on under the hood?
* Good exercise, easy to see success
* Some more intro how to think about tests (test design). Would be useful for researchers without and formal programming education
* Very good exercise. That included a lot of stuffs we covered before
* The exercises were really helpful for understanding
* The lecture was too slow without too little background information
* I liked the pair work of breaking and fixing
* Additional more complex function tests even if it is just as optional suggested reading

## Jupyter

* The exercises could be more challenging :)
* Perfection! Well done!
* Applicable content
* Interesting and comprehensive
* I find jupyter stuff hard to google, maybe add something about helpful search terms
* Very interactive -> good
* Maybe a little more info on when it is useful to use Jupyter
* Really interesting topic
* Maybe a little bit long on magic
* Great intro, great pace! (and very useful topic)

## Modular code development

* Interesting
* Really good session on modularity
* Could be nice to have some best practices for modularity for each(some) programming languages
* More time for this session -> It is very useful
* General part was very good

## Software Licensing

* Never thought about this before
* Special situation in Norway?
* Learned a lot of new things - very interesting
* Feedback on line - survey?
* Great work - usefull links
* Wish we had more time

* Good introduction
* Totally run out of time
* Good but short
* Too ambitious to think there would be time for discussion...