

ASPP 2023



Tübingen AI
Center



FORTH

INSTITUTE OF MOLECULAR BIOLOGY & BIOTECHNOLOGY

15th Advanced Scientific Programming in Python

Summer School

27 August – 3 September, 2023. Heraklion, Crete, Greece

Evaluation Survey Results

Method

The survey has been administered with a web interface created with the LimeSurvey software available at:

<http://www.limesurvey.org>

All answers have been submitted by 26 September, 2023.

No answer was mandatory.

The free-text answers have not been edited and are presented in their original form, including typos.

Attendants and Applicants Statistics

	Attendants		Applicants	
	30	19%	156	
Different nationalities	14		42	
Countries of affiliation	9		32	
Gender: other	0	0%	3	2%
Gender: female	17	57%	76	49%
Gender: male	13	43%	77	49%
Already applied	9	30%	25	16%
Bachelor Student	1	3%	10	6%
Master Student	1	3%	32	21%
PhD Students	20	67%	76	49%
Post-Docs	6	20%	12	8%
Professor	0	0%	1	1%
Technician	0	0%	1	1%
Employee	1	4%	11	7%
Others	1	3%	13	8%
Completed surveys	30	100%		

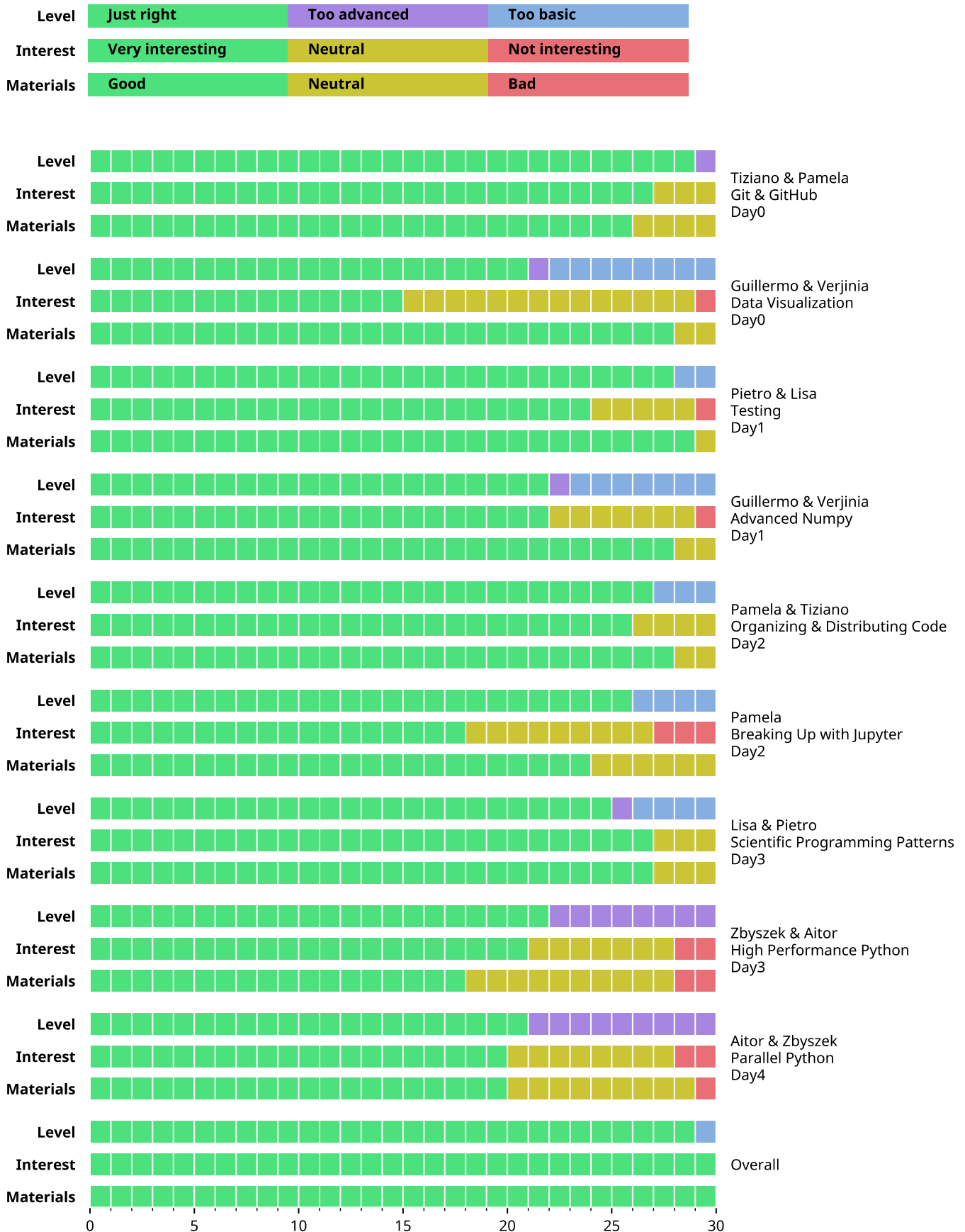
More stats about attendants are available at: <https://aspp.school/2023-heraklion/students.html>

Lectures & Exercises

Q: Grade the **level** of the lectures

Q: Grade how **interesting** were the lectures

Q: Grade the quality of the presentation style and/or of the teaching **material**, e.g. the clarity of the slides/code, the exercises and the solutions, etc.



Q: Are some of the topics presented in the lectures not relevant for a programming scientist?

1. For me, the lectures were fascinating, and deepening on your research, I think different people will benefit from multiple of the lessons.
2. the parts on Cython and Numba can be potentially useful but at the moment I find them to be of limited usefulness in my scientific work
3. In my everyday job right now it is highly unlikely that I will use any of the parallel programming topics. For me a slower paced github intro at the beginning would have been more useful. In overall every topic was interesting, just not all that relevant for me (in the present)
4. Testing is less relevant for the scientists I know, but I like that you included it, especially since from what other participant said it seems to be very relevant in some scientific fields/groups. I am not sure how widely relevant high-performance and parallel python are for many scientists-- there's definitely a subset who will highly benefit, but I suspect it's a relatively small percentage compared to the other topics.

Q: Are there further topics relevant to the programming scientist that could have been presented, given that the total time is limited. Please also mention which topics should be replaced by the new ones.

1. maybe can also talk about Python name reference and memory management
2. I would like more details about how to develop your own package.
3. ,Numpy was explored, but maybe scipy could have been presented a bit?
4. Cloud Computing instead of Breaking Up with Jupyter.:
5. The part on "publishing/documenting code" seems to be included in the ODD lecture according to the slides, but I think we didn't have time to reach that point. And it seems like a very small portion of the course anyway (still only based on the slides). I do think that this is however a crucial skill for a programming scientist, and I would have benefitted a lot for spending the time going over that part of the lecture and have a couple of exercises to practise (e.g., reproduce/run some code only based on doc, rename variables/functions, etc.)

I would suggest spending less time on the (frustrating, per Pamela's terms ;)) "import exercise" to somehow give more room to the documenting part.

6. I would replace half of the plotting lecture (the one that refers to basic plotting commands and how to fine tune figures), because I think that is something that tools like chat gpt for example can do very efficiently and it is a more or less a repetitive task. Perhaps it is a good idea to explore how tools like chat GPT and whatever comes next can facilitate or hinder coding and see if that can go into the program. the part of the lecture about creating subplots and using Gridspec should stay.

In the advanced Numpy section I would put more emphasis on how to methodically produce more vectorized operations, or how to use numpy to perform basic operations faster and explore how to find good opportunities to do vectorization in one's code. perhaps that can substitute some of the too deep detail on high performance python.

An exercise where the students can turn smelly code (disorganized/tons of if-elifs ... etc) into roses and perfume code could be greatly useful.

7. decorators in more depth as one encounters them a lot (within the numba topic maybe?) and an added 10 min about generators (e.g. within the classes topic)
8. I think for the last topic about parallel processing, a rough introduction about the hardware of computers (a general introduction about what RAM, CPU are) would be a good addition. Because there were so many confusion about the relationship between the cpu and the processors.
9. I would be interested in learning about Pandas. It could replace data visualisation and breaking up with Jupiter.

Also I think in many cases we do not write our own code from scratch, but we have to further develop existing code. Often it is very hard to read and impossible to rewrite it. I would be interested in learning how to best integrate my code with other "smelly code". It could become part of the Scientific Programming Patterns.

10. Additional topics:

- Working with legacy code, e.g., code from other researchers.

Candidate for replacement:

- Advanced NumPy:

11. I would have liked to start the NumPy course with more advanced knowledge on the library. However, at the same time, I understand that it depends on the level of the student.
12. connecting to clusters, super computers, networks, etc. Leave out data visualization
13. Maybe pandas? Could replace the matplotlib session
14. My ideal would have been less parallel programming and replace that two lectures with more github practice and programming patterns topics/ data viz.

15. I found all topics helpful, no need to change something
16. I thought we might have more debugging-related content: identifying a problem, how to find it, how to solve...
17. I think the data visualisation session could be replaced by another topic. It's very relevant but I think for this topic it's really easy to read a paper/article and get the important points. I would have preferred to have a little more time in the beginning to practice the git workflow (especially differences in revert, restore etc.). In case there are good solutions available at some point working with git and notebooks would be helpful.
18. I'd have liked some additional tricks working with data science packages, e.g. pandas. How to organize your data in a tidy way and how to make the most out of the package with exercises on grouping functions and so on. I often find myself a bit confused about how to best work with dataframes in a more complex scenario. I'd have replaced maybe the part about parallel computing as it seemed generally we should try to avoid it if possible. It was still very interesting but maybe from my personal point of view one could get some insights if they really needed it. Though I understand that my interest in data science is also biased.
19. No, I think the distribution of topics was excellent.
20. - how to correctly cite packages in publications (when / how)
 - choosing a licence for your code, and what happens if you don't (no one can use your code)
 - considerations / best practices when publishing your code alongside a paper (e.g. how to give versions of your code different (or the same) DOIs and where to put the code)
 - VSCode keyboard shortcuts (I learned some from my pair-programming partners that are super useful!) e.g. tab-complete, ctrl+c in a line copies the line, renaming all instances of a variable
 - configuration files -- types (I think you mentioned this at some point), how to write a good one, how to import them, type validation (a few of my colleagues recently discovered pydantic and it is making their lives a lot easier), saving a config file alongside results to document the settings for different results. This would basically expand what you already covered in the inputs.json part of the Reproducibility section in Programming Patterns.

Most of these are pretty short so I don't think they'd require entire topics to be cut, but possibly combine high-performance and parallel python into one section to save some time?

I have also got into writing small GUIs to make it easier to select parameters, run code, and visualize outputs all at once, basically wrapping the .py files in something visual. e.g. with Dash. Not sure this is applicable to everyone, and maybe Jupyter Notebooks with widgets do a lot of this already, but I like being able to keep the .py files as they are while still getting some visual feedback on inputs/outputs, and it is not so hard to write a basic GUI.

Q: Do you think that pair-programming during the exercises was useful?

Yes, I have learned from my partner / I have helped my partner	93% (28)
No, it was a waste of time for both me and my partner	0% (0)
Neutral. It was OK, but I could have worked by myself as well.	3% (1)
Other	3% (1)

Other:

1. I had several partners where it went great and I learned a lot, and a few where I think my partner and I would have learned more on our own. So my answer is "yes, overall, but sometimes a waste of time"

Q: What do you think of the balance between lectures and exercises? When answering, please keep in mind that the overall time is limited ;-)

Lectures were too long, there should be more time for exercises	10% (3)
Lectures were too short, there should be more time for lectures	3% (1)
The time dedicated to lectures and exercises was well balanced	87% (26)
Other	0% (0)

Q: Any further comments about the lectures and exercises?

1. I think the ratio between lectures and exercises was really great, it made the time fly, considering the sessions were from 09:30 to 19:30 :)
2. Perhaps a bit more theoretical content could be useful to go back home with, so later on students could use the material as their 'python reference'? Just trying to make a suggestion, it's great as it already is.
3. I really liked the collaboration between senior and junior members of the faculty while teaching each module. It shows authentic interest in giving space for a new generation of faculty members. It also brought a great balance between experience and freshness. Unfortunately, few times, the dynamic between junior and senior faculty felt a slightly uncoordinated and lead to minimal misunderstandings. Most times, these misunderstandings didn't affect the learning experience and will hopefully be solved once junior faculty are more integrated into the dynamics of the course.
4. In general I found the lectures and exercises very useful, interactive and fun.
5. Data visualization exercises: I would like to have a bit more on arrangement of subplots, e.g. when you don't know beforehand how many subplots you'll need, how do you arrange/distribute evenly.
6. Overall I think Lectures and exercises are balanced, just that about half of the time the time allocated for the exercises was a bit too short for me to complete them. But then not all of the exercises were meant to / required to be finished for us to be able to continue, so I think it worked out well.
7. Awesome lectures (at least to some extent because of the awesome lecturers) and the exercises really helped to build the confidence in order to apply the content from the lectures - in the course but also in everyday practice after the school :)
8. The average time allocation was great, but lectures were not homogeneous in this sense, so maybe outlying lectures could get closer to the mean.
9. It was in overall fine. I would however move the schedule of the Advanced Numpy not to be from 17 to 19:30 - for me this module required more efforts than the lecture and exercises that were before, however by the time we got to it I was already mentally tired and could not get as much into the exercises as I would have wanted.
10. The exercise level was just right and they were all just relevant to the lecture. I highly enjoyed and learned a lot from both.
11. In some lectures there were too many exercises in my view, that were either quite small or hard to complete. It might be worthwhile to have fewer but longer exercises.
12. The lunch break could have been not as long. Maybe an hour would have been enough
13. I think overall there was a very good balance between lectures and exercises. However, in some cases, I would have liked to do more exercises. Such as debugging with pdb which is very important, but we didn't manage to cover in exercises. The same for debugging with PyCharm, even though it would have been more difficult because PyCharm was not installed in the computers and not everyone is familiar with this IDE.
14. It was such an amazing experience, that I would love to have the summer school running longer!
15. In topics that I was not familiar with I could have used more time for lectures and the same time for exercises. I understand although that the time is limited, ideally a 10 day course with slower paced lectures would have been my dream :D
16. All lectures were interesting and well presented. Some exercises were difficult but in the end well explained.
17. It was a very intense learning experience, the mixture of topics, lectures and exercises was great. I really enjoyed that I knew how to use git workflows in the end of the week I had never heard before the summer school. Also the concepts how to structure and import own code was very good and really helps me with my own project right now.
18. I have to say that I really enjoyed the experience of the summer school. Not only did I feel that I can openly ask questions towards topics that I was interested in but also the level of the lectures were just very fitting. All the lectures were incredibly well prepared and I was interested in all of them. I liked that it was really meant to understand the mechanisms instead of plainly showing some examples using different packages. I found the lecture material original and very intuitive and the atmosphere welcoming to further inquire about more difficult problems I encounter during my PhD. The organizers did not only make sure that the educational value is held to a high standard but also that the atmosphere for social interaction was very pleasant. For me it was my first summer school in person and I really had a lot of fun also interacting with the other students. I will definitely stay in touch with some of them that live closeby. I felt that this is a very valuable experience both for learning and for networking!
19. Perfect mix between lectures and practicals so that I could stay focused the whole day!
20. All the exercises were well thought through and helped me to understand the material covered in the lectures.
21. Perfect balance between lectures and exercises, both in terms of total time spent on each and in terms of the time of each chunk of lecture or exercise. I was also super happy and impressed that nearly all lectures ended on time! The topics for the exercises directly complemented the topics covered in the lectures.

Programming Project

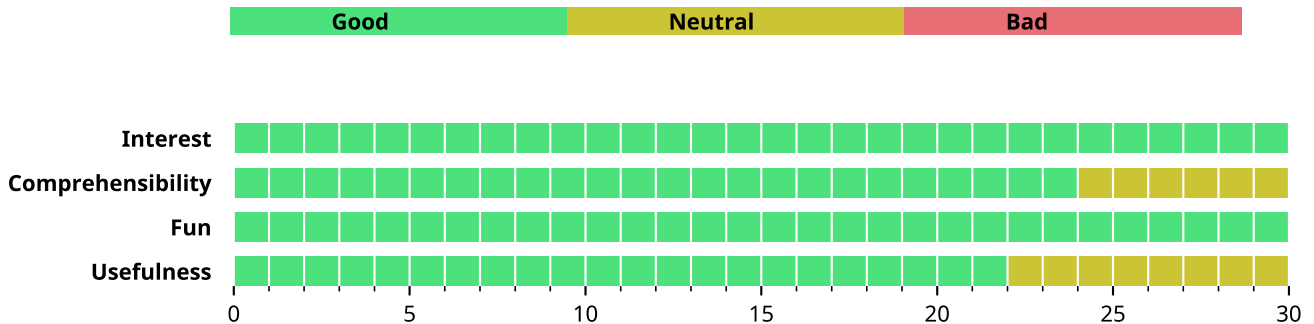
Q: Evaluate the programming project.

Interest: How interesting was the programming project?

Comprehensibility: How clear and comprehensible was the code and the available documentation? Was it easy to work on the programming project

Fun: Was it fun to work on the programming project?

Usefulness: Was it useful to work on the programming project? Do you think you may re-use what you learned?



Q: Do you think the team-programming experience is relevant to your work as a programming scientist?

Yes: 87% (26)

No: 13% (4)

Q: Do you think that the project should be about a real-world scientific problem instead of a video game?

Yes: 3% (1)

No: 97% (29)

Q: Any further comments about the programming project?

1. I really enjoyed the programming project! It was fun and exciting and I learned a lot working in a team.
However, I think depending on the dynamics of the team, you could focus more on the competitive part of the game, which will make you prioritize functionality, instead of testing, creating classes, and other stuff that you learned during the course. If you are just being introduced to these topics, it could still be difficult to implement them when you are in a time pressure scenario, and you just want to ensure your bots will have a smart run :D
But I really enjoyed it! Truly amazing experience!
2. I found that the programming project was useful in cementing what we learned during the week about git, testing, and good practices. It also gave me an idea of what collaborative programming projects could look like in professional settings with a team of several people.
3. I think it would be interesting to try to find a non-competitive approach to the gaming project. Cooperative gaming can be as fun as competitive gaming. A cooperative approach to the gaming project would be congruent to modern academic practices where the biggest problems are solved by large consortia rather than small groups competing against each other.
4. Since time is limited but the programming project is so fun to work on, I would maybe suggest to quickly introduce the project from the very beginning of the school, and somehow make some connections/examples/tips between lectures and the project throughout the week, so that it makes it a bit quicker for the group to start working on the project when the time comes, instead of "wasting time" (quotation marks because it's not a waste of time per se, simply given the limited time we have, it seems like it could be better used) figuring out how we can apply whatever we learned the previous days.
5. In general I found the project to be very satisfying and to mimic very well the pressures and needs of scientific coding. The team experience allowed me to interact with other people's code and to think about how to make my own code more readable and concise. In addition, I found the project to be the one of the highlights of the course and it was a lot of fun. I wouldn't have minded (meaning: it would be great) if the course was extended for half a day to have some more project time, one more 'report' and to allow team members to rotate and optimize, test and make the code readable.

I found that some parts of the course were very well represented in the project while others were not.

For example, what we learned on Git was practiced in the project consistently and helped my learning, while I think across all groups the amount of testing was very limited. Perhaps some conditions could be put in place that stimulate the use of other of the tools learned in the course. For example, create a pull request of a test,

your code has to contain a class, or something funny happens if you have too many 'if, elif' statements, etc...

alternatively, It could be very good the groups could have a meeting with one tutor/faculty for code review/report, where the tutor/faculty checks the team code and notes specific opportunities for improving the coding practices in the team or to use relevant parts of the course in the project.

6. it felt the groups were not really heterogeneous (groups with people with same nationality, same expertise/affiliation, same level of coding skills)
would be better if they were a bit more mixed ;)
but it was great anyway!!
7. The programming project was an awesome finish. Couldn't have wished for anything else. The instructors put a lot of effort and care in the set up of the competition which made the conditions of a group project very realistic, yet lots of fun (with count down, music, and final event). Even though it may feel frustrating at times to not be able to apply techniques learned before, the project is a better example of what is feasible and realistic to do in everyday work life and teaches making compromises between writing code well and finishing work. Also it added a whole other dimension to the course in teaching to work with other non-expert coders, be aware of ones own and others limitations and expectations and make the most of it. Very valuable and educational experience.
8. I think the most I learned in this project was about how to work under time pressure as a team of people who never worked together before :)
9. It was a good experience to work as a team on the project that is not directly relevant to what we professionally do. Essentially though, the programming project required everything that as scientist we practice daily: understanding the problem, coming up with creative solutions, distributing workload fairly among the team members, having lots of discussions.
10. I do not have much experience in working on code together. The programming project was a very useful and fun exercise to improve my collaborative coding skills.
11. I really enjoyed working on the programming project together with my team. Overall, I thought it was very useful to learn how to collaborate in github. If we didn't have time constraints though, I believe I would have put into practice other learnings from the summer school such as doc string documentation and debugging.
12. It's really hard to say if a science problem should be the final project. I guess that if there is only one week of summer school, then the final project should still be the game because even things out.
13. On average most people had more experience in python than me. This meant that I learnt a lot, but it also meant that I was a bit tired at the end of the school for the programming project. I suggested that my pair would do the coding and I would help in thinking about the solutions and only suggest some coding solutions, which worked out quite good and both of us enjoyed a lot the experience. I felt that the git part of the project was overwhelming for me, so I left the work to the git pope and co. Overall, I think there was not enough time to use all the skills, tricks that we learnt. It was a unique work experience with a lot of group dynamics which was a great experience overall, but I sometimes felt that maybe dropping this group project and spend more time with the topics would have been more beneficial for my knowledge level.
14. It was fun and a great way to end the summer school!!
15. The programming project was really fun and I enjoyed working on it. I have to admit that I got really excited about it and felt that I can learn a lot here. The interface was really well documented and it didn't take a long time to get started with the coding once we had thought of a strategy. Thoroughly enjoyed working on it!
16. Since I have never programmed interactively in a team before, this project was a great learning experience. The most important part for me was observing the working style of others, the group dynamics and to assess my own contributions better.
17. I had a blast during the programming project and was surprised by myself how competitive I got. What a fun way to learn how to develop in teams!
18. I enjoyed the programming project more than I expected :) The discussion topics that were suggested for our first meeting, and making the collaboration document, were helpful to clarify our group dynamic. Our different backgrounds and abilities complimented each other well and we had a ton of fun (in addition to being productive!). I learned a lot about how to program effectively with others and how to manage git workflows during a project with multiple contributors -- this will be super useful for my work.

The School in General

Q: How do you overall evaluate the school?

Good: 100% (30)

Neutral: 0% (0)

Bad: 0% (0)

Q: How do you evaluate the general level of the school? Was it too advanced/too basic with respect to your expectations?

Too advanced: 3% (1)

Just Right: 93% (28)

Too basic: 3% (1)

Q: How do you evaluate the general level of the school? Was it too advanced/too basic with respect to what was advertised in the announcement?

Too advanced: 3% (1)

Just Right: 90% (27)

Too basic: 7% (2)

Q: Did you learn more from attending the school than you would have learned from reading books and online tutorials alone?

Yes: 100% (30)

No: 0% (0)

Q: How do you evaluate social interactions and social activities at the school?

Good: 100% (30)

Neutral: 0% (0)

Bad: 0% (0)

Q: Would you recommend this course to other students and colleagues?

Yes: 100% (30)

No: 0% (0)

Q: How did you hear about the school?

Google Search: 4

Professor/Tutor/Supervisor: 5

Colleague/Friend: 17

Website/Mailing list: 7

1. university twitter post
2. Mailing list from my institute
3. Comp neuro mailing list
4. Twitter, cvnet mailing list
5. BCCN Berlin
6. BCCN Berlin
7. mailing list ITB

Q: Any further comments or suggestions?

1. Thank you so much for the fantastic opportunity and all the good vibes everyone on the team had!!!
I had an incredible experience, from the learning aspect but also from the social atmosphere, and I hope you can go to many other countries, so much more people can share this experience.
2. I strongly recommend attending ASPP.
3. There are some particularities in the way a scientists learn how to program in python. These particularities tend to deviate us from true good practices in coding which have been thought of by a community beyond the academic environment . This is the course to get back in the track of good practice in coding and to experience how good practice can positively impact collaboration, efficiency, transparency, and reproducibility. I am truly grateful towards the organizers and faculty.
★★★★★ Totally recommended!!!
4. The school and all of you tutors are great and I can't thank you enough :) I wish I could do it all over again!

5. The format is great! Please keep it that way!
6. It is obvious to me that the course was prepared with dedication and love. Overall I am very satisfied.
I think that I learned critical things that most scientist don't apply but should apply in their work and I met very nice people. I wish many others have the opportunity to attend
7. I liked that the social activities / hotel and the work space were separated by a bus drive.
8. Using Bad/Neutral/Good (or other 3-point scales) in this evaluation survey (instead of e.g. 5 grading levels) leaves me with a Bad feeling, because I couldn't be precise, especially when expressing negative opinions . Also, some of you used fancy or funny examples during lectures, but at some points I felt that this did not result in more fun, just more confusion. Finally and mostly, thanks for the whole experience, it was fantastic!
9. I heard about ASPP from a random friend that I consider it happened just by chance. I think you should advertise this event more rather than "word of mouth" because it was very helpful and informative such that I suppose more people need to hear about it.
10. My answers for the group programming project includes some tips, but these are for people with my level in python: more lecture time and less group programming. The pair programming part is awesome! Also, I think you should consider giving out the specifics for the os and the programmes that will be available on the laptops. I came from a background of Spyder, Anaconda (virtual envs, etc.), some terminal handling. But I was a bit overwhelmed at the beginning with getting used to the os and putting together a reasonable workflow for myself. A specification for available software would have been helpful for me before the school.
11. This summer school was really great, the scientific part but also the social part, thank you !
12. I enjoyed the experience a lot. Keep at it like this :)
13. ASPP team, thanks so much for everything!
14. This school was even more awesome than I expected (and I had high expectations!). The topics covered were incredibly useful -- I learned a lot from the lectures, the pair programming, and the super-knowledgeable instructors. What really brought the course above my expectations was the people (both the instructors and the participants).

The instructors were fabulous! In addition to being wonderful people, I appreciated how willing they were to share their knowledge, and the diversity of their experience in various areas of both academia and industry. No matter what question I had, there was an instructor who could answer it thoroughly.

The participants made up one of the most consistently interesting groups of people I've ever been around. I've gotten jaded about "diversity", but in this case the conscious efforts of the organizers to include participants with widely different experiences absolutely paid off. Nearly everyone I met during this course was super-lovely and a fascinating human.

Enforced mixing of participants during sessions, and the balance of social activities, meant that we got to interact with everyone over the week. This was particularly nice to strengthen the group as a whole, rather than multiple cliques forming.

Plus the location and food were amazing :D

Thanks so much for an incredible week! I would recommend ASPP to any scientist looking to level up their Python skills.

Tl;dr: Loved the food! Loved the people! Loved the useful topics and teaching style! LOVED that everything started and ended on time (omg so rare, and so awesome)! Loved the comfy chairs ;) Thank you so much for an incredible week! I will def be using the skills I learned at ASPP in my future work.