DUNE Computing Training Plans

Claire David David DeMuth

June 14th, 2021

What we have done

October 2020 A tutorial wish-list survey → link | knowing the demand

January 2021 First training | one morning | wiki-based (restricted)

Well received

May 2021 Second training | three consecutive days | markdown-based (public)

Excellent feedback

Self-serve tutorial available anytime for newcomers

- training website remains online
- asynchronous resources

What we plan to do

Good questions to ask (ourselves)

Ideas

How to gauge the attendance: number, level, topic? ⇒ sending a survey

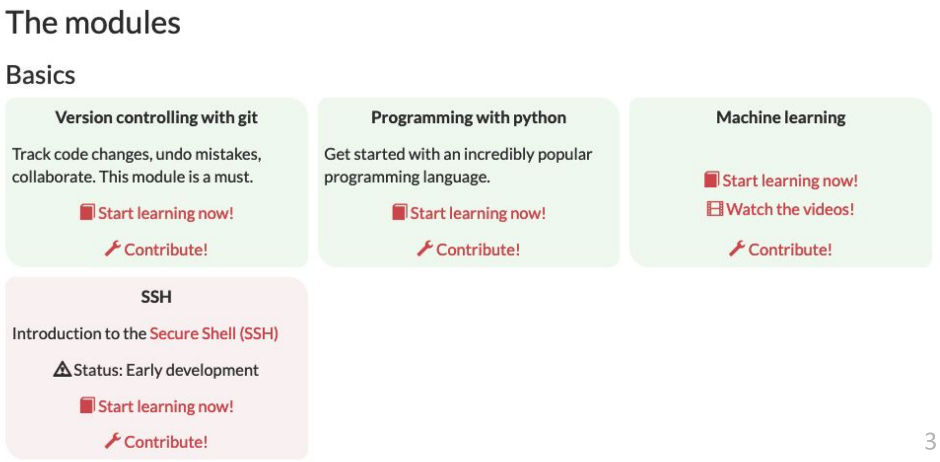
Could we modularize into smaller standalone trainings? ⇒ aim for next iteration (Sept 2021)

Could our trainings be adopted into the HSF curriculum? ⇒ will report to HSF

What HSF curriculum can we leverage?

What DUNE computing trainings could be developed next (low-hanging fruit)?

Find a Q&A platform (Askbot-like)



The modules

Basics

Version controlling with git

Track code changes, undo mistakes, collaborate. This module is a must.

Start learning now!

Contribute!

Programming with python

Get started with an incredibly popular programming language.

Start learning now!

Contribute!

Machine learning

Start learning now!

Watch the videos!

Contribute!

SSH

Introduction to the Secure Shell (SSH)

⚠ Status: Early development

Start learning now!

Contribute!

Next months

May Training 2.0

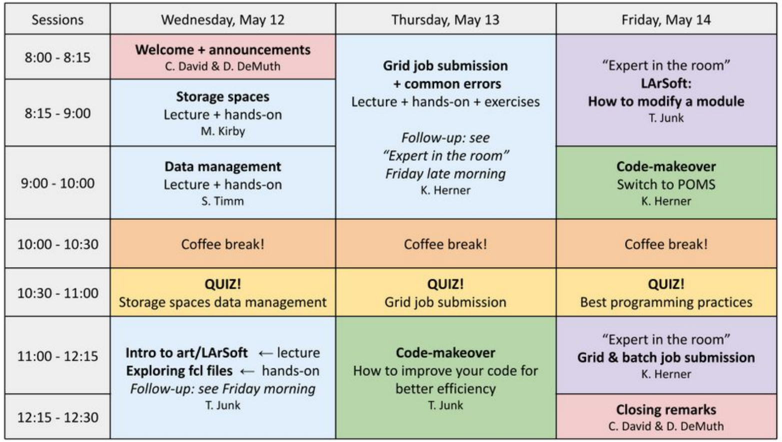
Polishing the material | shorten video-summary

Next 'standard' training | target: September 2021

Three days spread during the entire week

Advertised 6 weeks ahead → FNAL account request

Setup instructions sent 2 weeks before



Sessions	Wednesday, May 12	Thursday, May 13	Friday, May 14
8:00 - 8:15	**Welcome + announcements** C. David & D. DeMuth	**Grid job submission** **+ common errors** Lecture + hands-on + exercises *Follow-up: see* *"Expert in the room"* *Friday late morning* K. Herner	"Expert in the room" **LArSoft:** **How to modify a module** T. Junk
8:15 - 9:00	**Storage spaces** Lecture + hands-on M. Kirby		
9:00 - 10:00	**Data management** Lecture + hands-on S. Timm		**Code-makeover** Switch to POMS K. Herner
10:00 - 10:30	Coffee break!	Coffee break!	Coffee break!
10:30 - 11:00	**QUIZ!** Storage spaces data management	**QUIZ!** Grid job submission	**QUIZ!** Best programming practices
11:00 - 12:15	**Intro to art/LArSoft** ← lecture **Exploring fcl files** ← hands-on *Follow-up: see Friday morning* T. Junk	**Code-makeover** How to improve your code for better efficiency T. Junk	"Expert in the room" **Grid & batch job submission** K. Herner
12:15 - 12:30			**Closing remarks** C. David & D. DeMuth

May 2021 Training schedule

Specific training

Departing from LArSoft → focusing on e.g. GarSoft, Pandora, CAFAna, WireCell, NUISANCE

Covering generators → GENIE, Marley, Cry, CORSIKA etc...

Upgrading from "code-makeover" to a "hackathon format"

⇒ come back home with a analysis workflow

Notes

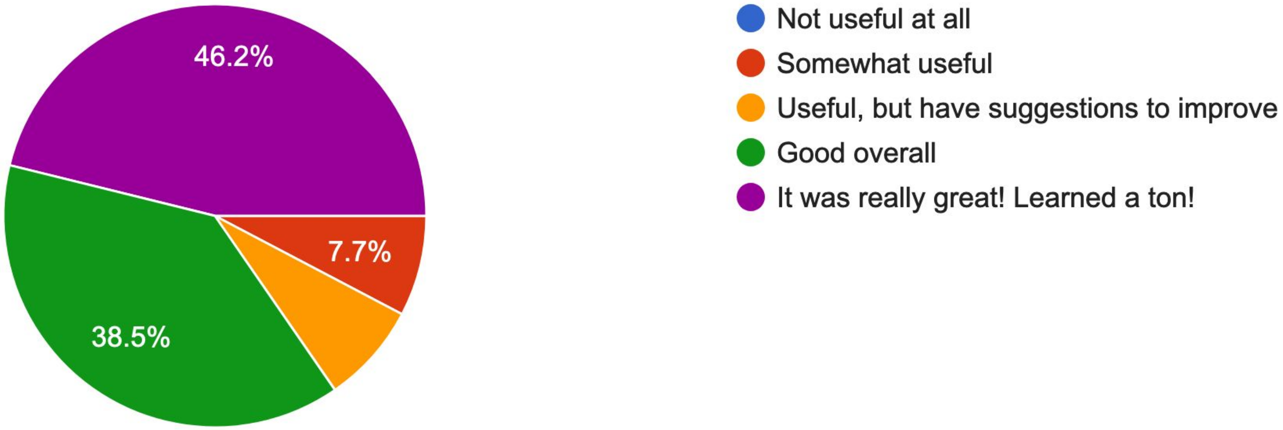
Adding here remarks, comments etc

Backup

General feedback of May training

How was it?

26 responses



We can call this a success.

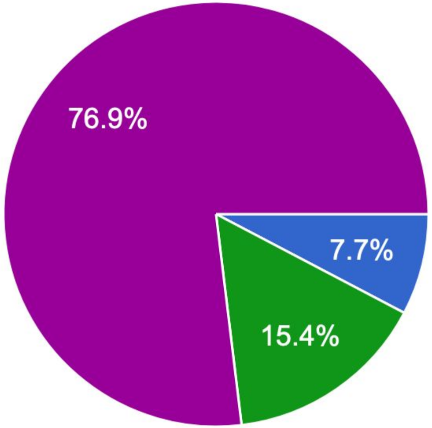
Positive feedback: what people like

- the content: helpful
- the atmosphere: motivating and pleasant
- the training website i.e. github page using Software Carpentry format
- interactivity: seeing experts explaining on their own terminal
- quizzes “More quizzes in the future!”
- the livedocs + timely responses to anonymous questions

Live docs and quizzes

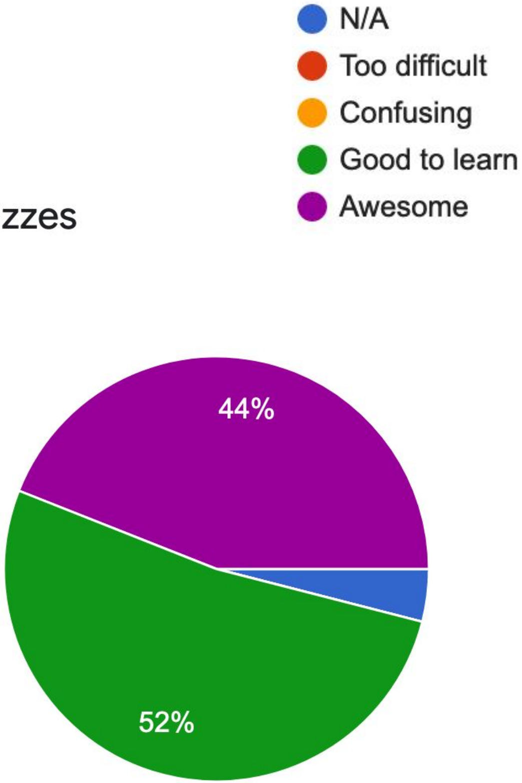
How was the support on the live doc?

26 responses



How were the quizzes

25 responses



Where we could improve

- sending the setup instructions earlier (my bad!)
- insisting (even more) that the setup needs to be done before the event
- more head up time for asynchronous tutorial
- slower pace on the lectures

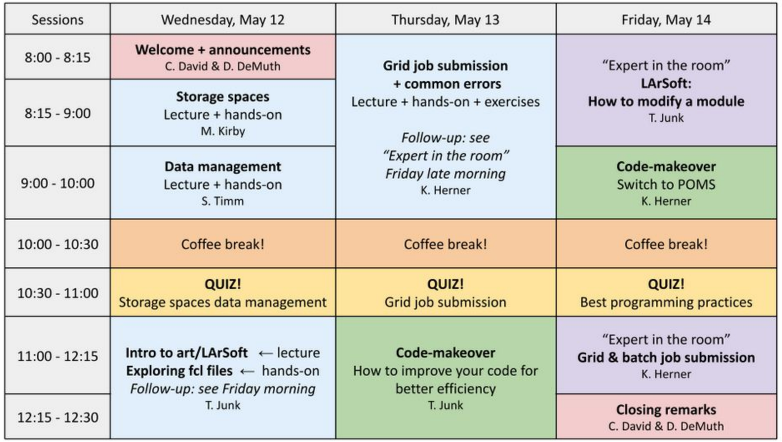
Good suggestions

- spreading the tutorial: Monday - Wednesday - Friday
- break instead of breakout rooms
- more examples / practice problems
- less overlap and better integration of the sections

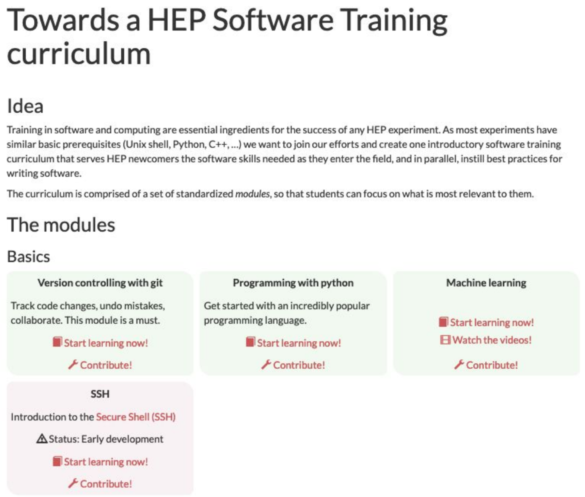
See the full survey on Indico

What Next?

- A three day training developed from past workshops with a markdown-based HSF/Software Carpentries.
  - Zoom sessions captured, post-produced, hosted for asynchronous viewing on YouTube
- Could we modularize into smaller standalone trainings similar to HSF?
  - What HSF curriculum can we leverage?
- Could our trainings be adopted into the HSF curriculum?
- What DUNE computing trainings could be developed next (low-hanging fruit)?



Sessions	Wednesday, May 12	Thursday, May 13	Friday, May 14
8:00 - 8:15	**Welcome + announcements** C. David & D. DeMuth	**Grid job submission** **+ common errors** Lecture + hands-on + exercises *Follow-up: see* *"Expert in the room"* *Friday late morning* K. Herner	"Expert in the room" **LArSoft:** **How to modify a module** T. Junk
8:15 - 9:00	**Storage spaces** Lecture + hands-on M. Kirby		
9:00 - 10:00	**Data management** Lecture + hands-on S. Timm		**Code-makeover** Switch to POMS K. Herner
10:00 - 10:30	Coffee break!	Coffee break!	Coffee break!
10:30 - 11:00	**QUIZ!** Storage spaces data management	**QUIZ!** Grid job submission	**QUIZ!** Best programming practices
11:00 - 12:15	**Intro to art/LArSoft** ← lecture **Exploring fcl files** ← hands-on *Follow-up: see Friday morning* T. Junk	**Code-makeover** How to improve your code for better efficiency T. Junk	"Expert in the room" **Grid & batch job submission** K. Herner
12:15 - 12:30			**Closing remarks** C. David & D. DeMuth



Towards a HEP Software Training curriculum

Idea

Training in software and computing are essential ingredients for the success of any HEP experiment. As most experiments have similar basic prerequisites (Unix shell, Python, C++, ...) we want to join our efforts and create one introductory software training curriculum that serves HEP newcomers the software skills needed as they enter the field, and in parallel, instill best practices for writing software.

The curriculum is comprised of a set of standardized *modules*, so that students can focus on what is most relevant to them.

The modules

Basics

Version controlling with git

Track code changes, undo mistakes, collaborate. This module is a must.

Start learning now!

Contribute!

Programming with python

Get started with an incredibly popular programming language.

Start learning now!

Contribute!

Machine learning

Start learning now!

Watch the videos!

Contribute!

SSH

Introduction to the Secure Shell (SSH)

⚠ Status: Early development

Start learning now!

Contribute!