

Building a Global HEP Software Training Community

Kilian Lieret^{1,2}, Wouter Deconinck, Peter Elmer^{1,2}, Sudhir Malik³, Alexander Moreno⁴, Stefan Roiser⁵, Jason Vaetch⁶, Michel Hernandez Villanueva⁷
for the IRIS-HEP/HSF Training Group

¹Princeton University

²IRIS-HEP

³University of Puerto Rico

⁴Universidad Antonio Nariño

⁵CERN

⁶Universität Göttingen

⁷DESY

⁸University of Manitoba





$O(10k)$

HEP people worldwide

$O(6k)$

Undergrads

$O(3k)$

Ph.D. students

$O(1k)$

Postdocs

$O(\text{depressing})$
Faculty

need to be trained in software engineering & computing
to tackle the computing challenges ahead

This should be a **community** effort!

Why not leave everything to the experiments?

- General software skills matter as much as experiment-specific training
- Experiments share a lot of software topics
- Software topics are vast & evolving! (e.g., ML, GPUs, FPGAs, ...)
- **We can cover more ground together** instead of reinventing the wheel...

$O(6k)$

Undergrads



Teach prerequisites &
Democratize science



$O(3k)$

Ph.D. students



Impart best practices



$O(1k)$

Postdocs



Put in touch with
most recent developments



We need a **unified, scalable, and sustainable** software training framework

We need a **unified**, **scalable**, and **sustainable** software training framework

Unified

- Material and events should be **centrally listed** & **discoverable**
- Concentrate efforts by developing **cross-experiment** content
- A **community** must guide, support, and coordinate

Scalable

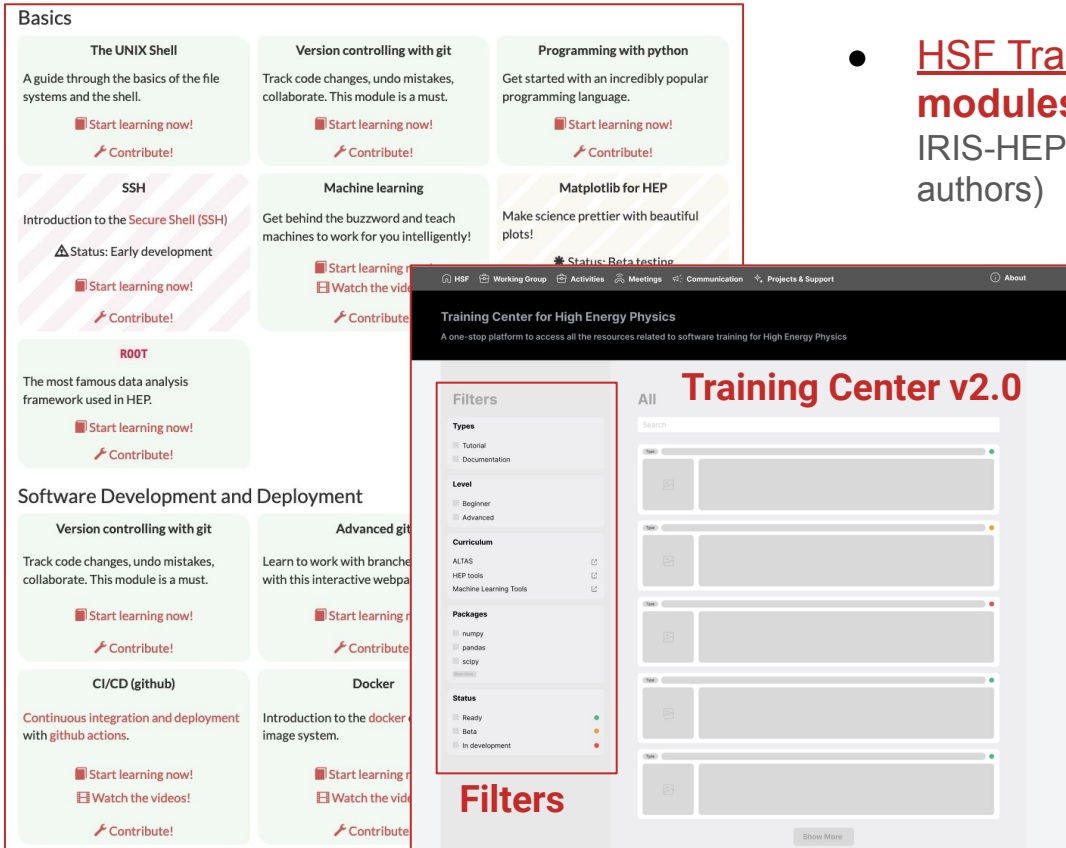
- Material must be teachable by **multiple instructors**
- **Self-study** must not be an afterthought

Sustainable

- Material must be **open source** and **maintained collaboratively**
- **Incentives & recognition** important motivators

The **IRIS-HEP/HSF Training** group is building a community around these principles

A unified Training Center for HEP



The image shows a collection of training module cards and a mockup of the Training Center v2.0 interface. The cards are organized into sections:

- Basics**
 - The UNIX Shell**: A guide through the basics of the file systems and the shell. (Start learning now! Contribute!)
 - 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!)
 - SSH**: Introduction to the Secure Shell (SSH). Status: Early development. (Start learning now! Contribute!)
 - Machine learning**: Get behind the buzzword and teach machines to work for you intelligently! (Start learning now! Watch the video! Contribute!)
 - Matplotlib for HEP**: Make science prettier with beautiful plots! Status: Beta testing. (Start learning now! Watch the video! Contribute!)
 - ROOT**: The most famous data analysis framework used in HEP. (Start learning now! Contribute!)
- Software Development and Deployment**
 - Version controlling with git**: Track code changes, undo mistakes, collaborate. This module is a must. (Start learning now! Contribute!)
 - Advanced git**: Learn to work with branches with this interactive webpage. (Start learning now! Contribute!)
 - CI/CD (github)**: Continuous integration and deployment with github actions. (Start learning now! Watch the videos! Contribute!)
 - Docker**: Introduction to the docker image system. (Start learning now! Watch the videos! Contribute!)

The mockup of the **Training Center v2.0** interface shows a navigation bar with links for HSF, Working Group, Activities, Meetings, Communication, Projects & Support, and About. Below the navigation bar is a search bar and a list of training modules. A **Filters** sidebar is visible on the left, with sections for:

- Types**: Tutorial, Documentation
- Level**: Beginner, Advanced
- Curriculum**: ALTA5, HEP tools, Machine Learning Tools
- Packages**: numpy, pandas, scipy, ROOT
- Status**: Ready, Beta, In development

- HSF Training Center currently lists **21 training modules** (including material developed by IRIS-HEP/HSF, The Carpentries, and individual authors)


- Goal: Training Center as a **focal point for all HEP Training activities**
 - Needs to list more
 - Add filters & dynamic sorting
- Central **list of training events** (everyone can add)

Mockup by Aniket Rana,
(anumbott @ github)
Inspired by learn.astropy.org

Training Modules

HSF Home Code of Conduct Setup Episodes - Extras - License Improve this page

Machine Learning on GPU



< Complete video walkthroughs!

This tutorial explores Machine Learning using GPU-enabled PyTorch for applications in high energy physics. It follows directly from the [Introduction to Machine Learning lesson](#) written by Meirin Evans.

Prerequisites

- A [Kaggle account](#). Click [here to create an account](#)
- Basic Python knowledge, e.g. through the [Software Carpentry Programming with Python lesson](#)
- Basic ML knowledge, e.g. through the [Introduction to Machine Learning lesson](#)

^ Lessons build on each other

Introduction

For physicists working on analysis in data-intensive fields such as particle physics, it's quite common these days to start developing new machine learning applications. But many machine learning applications run more efficiently on GPU.

The aim of this lesson is to:

- demonstrate how to move an existing machine learning model onto a GPU
- discuss some of the common issues that come up when using machine learning applications on GPUs

The skills we'll focus on:

1. Understanding a bit about GPUs
2. Using Python & PyTorch to discover what kind of GPU is available to you
3. Moving a machine learning model onto the GPU
4. Comparing the performance of the machine learning model between the CPU and the GPU

Most of our modules embrace the framework of The Carpentries

- Built from **markdown** files
- Rendered as a **webpage** with Jekyll (interesting discussion topic: The Carpentries have changed their framework recently)
- Verbose and **self-study ready**

Other training material written with LaTeX, sphinx, JupyterBook.

Example **Workshop**: C++ Training

Intro base OO More exp Tool conc py

542 slides, 688 pages, > 1k commits

HEP C++ course

B. Gruber, S. Hageboeck, S. Ponce
sebastien.ponce@cern.ch

CERN












March 10, 2023

6 events till now

- 6th HEP C++ Course and Hands-on Training (2023 March - essentials)
- 5th HEP C++ Course and Hands-on Training (2022 October - advanced)
- 4th HEP C++ Course and Hands-on Training (2022 March - essentials)
- 3rd HEP C++ Course and Hands-on Training (2021 August)
- 2nd HEP C++ Course and Hands-on Training (2021 January)
- 1st HEP C++ Course and Hands-on Training (2020 October)

- Has been taught **in-person, virtual and hybrid**
- **Life lectures and exercise sessions**
- Full videos available
- Next events:
 - 7th: May 15-19 @ JLAB
 - 8th: Aug 28 - Sep 1 @ Manchester
 - ...

Originally developed by **S. Ponce**, now community effort

 Abhishek L	 Attila Krasznahorkay	 Bernhard Manfred Gruber	 David Chamont	 Enrico Guiraud	 Graeme A Stewart	 Kilian Lieret
 Sebastien Ponce	 Stefan Roiser	 Stephan Hageboeck	 bcouturi			



SIDIS
Software Institute for
Data-Intensive Sciences



Weekly meetings

October 2022	
04 Oct	Training WG Planning Meeting
17 Oct	Training WG Planning Meeting
10 Oct	Training WG Planning Meeting
03 Oct	Training WG Planning Meeting
September 2022	
26 Sept	Training WG Planning Meeting
19 Sept	Training WG Planning Meeting
12 Sept	Training WG Planning Meeting
05 Sept	HSF Training Containerization Hackathon
05 Sept	Training WG Planning Meeting

Monthly Hackathons

Overview
 Timetable
 Registration
 Participant List
 Videoconference
 Code of Conduct
 Group Photo

Contact
 iris@hsftraining.edu
 training@hsftraining.edu
 @hsftraining
 @hsftraining

Containerization Training Hackathon

The big goal!

Training in software and computing is an essential ingredient for the success of any HEP experiment. As most experiments have similar basic prerequisites we want to join our efforts and create one introductory software training curriculum that serves HEP newcomers from all experiments. This curriculum is made up of independent training modules and should contain all software skills needed as they enter the field while installing best practices for writing sustainable software.

We have started this work here and have completed and tested several of our modules to great success.

Platforms

Community pages

Our community

Amber Roepe (she/her) | Andrea Valassi | Clemens Lange | Dan Guest
 Daniel S. Katz | David Chamont | David Yakobovitch | Giorion Stark
 Graeme A Stewart | Henry Schreiner | Jackson Burzynski | Jim Pivarski

Increasing our reach

HSF Training @HSFTraining · Sep 12
 77 participants joined our event in July! Learning #python, #bash, #git, #ROOT, #ScikitHEP. Thanks to @raynamharris, @mightaswellcode, Rayvn Manuel, Alexander Moreno and @guillo99 for teaching!

HSF Training · Sep 13
 We were presenting at #PYHEP2022 where we talked about #indico, curriculum as well as how to become a part of...

Join us in building a better future for HEP! Find out more here: indico.cern.ch/event/1150631/

Learning Python the Sustainable Way: Hands Learned at HSF Training

HSF Training @HSFTraining · Sep 13
 Many thanks to Alexander Moreno, @Guillo99, and @mightaswellcode for presenting us!

Recognition

Authors

Thanks goes to these wonderful people (emoji key) who contributed to the content of the lesson:

Meirin Dan Evans | amonador | Guillermo A. Fidalgo-Rodriguez | MarchusGöt | ekauffm | Aman Desai | Kilian Lieret
 Wouter Deconinck | Michel J.L. Villanueva

How-to guides

HSF Training Workshop Checklist

Let's streamline our organization and make sure we don't forget anything!

Note: there's also a [Hackathon checklist](#).

Before the workshop

Setting up documents and more

- Create a new folder in our drive in the folder "20YYour_workshops/YM_name_of_workshop"
- Copy this document there and call it "YYMM_name_checklist"
- Create an overview document "YYMM_name_planning". Use this for all the relevant planning info
- Create a document "YYMM_name_post-mortem". Collect everything that goes wrong in this document

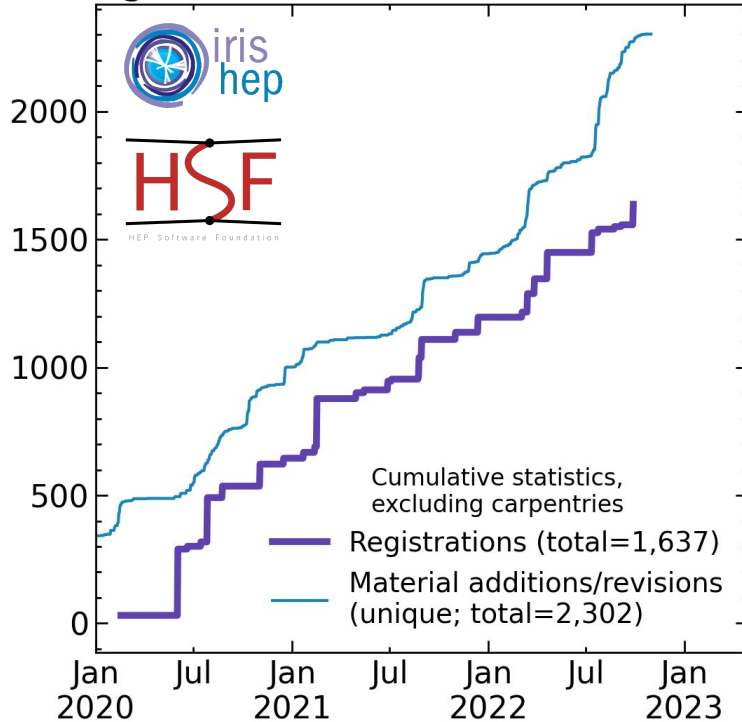
Create a slack channel for the event

Recruitment

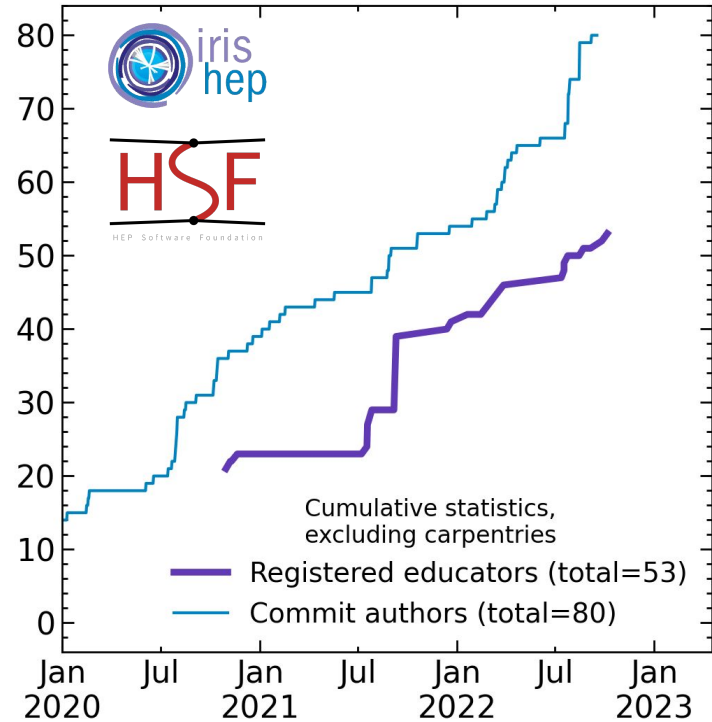
- Announce on twitter
- Announce on mailing list

We scale!

Registrations and material revisions

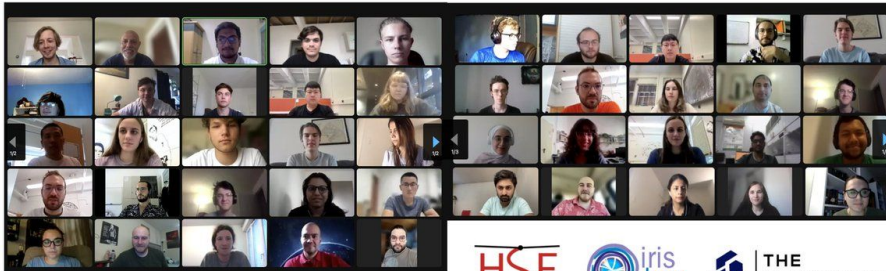


Educators



Conclusion

- We have established a **community of educators** working on cross-experiment training material
- **Join us** to
 - Make your training **discoverable**
 - **Avoid reinventing the wheel**
 - Ensure **sustainable maintenance**
 - Get help with **logistics & advertisement**
- Covering most basics/intermediate needs, now aiming at expanding intermediate/advanced training
→ **Seeding small groups to develop it**
- **Fill our survey to participate:**
bit.ly/state-of-training



Join us!

Take the survey: bit.ly/state-of-training



@hepssoftfound



@hsf-training



hepssoftwarefoundation.org

