Developing a Curriculum

Introduction and hands-on

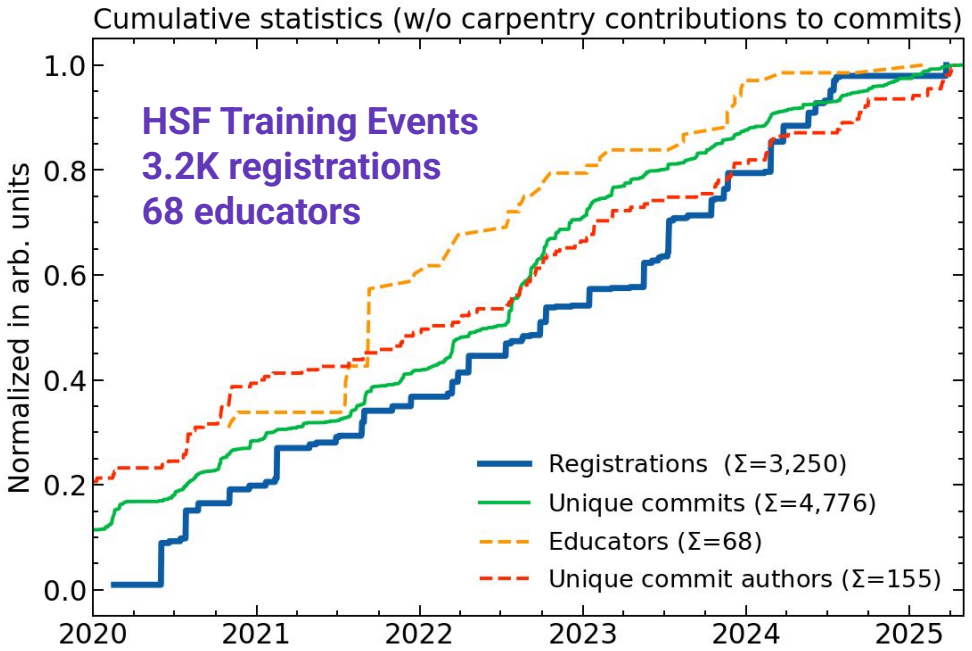
Michel Hernandez Villanueva
Brookhaven National Laboratory

… And many more! (see slide 9)

Deep Learning Train-the-Trainer Workshop
15–19 Sept 2025

The goal of these presentations

- Share our experience in training activities accumulated along the years
- A guide to individuals, groups and projects interested on developing training
  - Based on our howto-event page, offering a map for new explorers
  - Finding common challenges and teaming up!



The HEP Software Foundation (HSF)

https://hepsoftwarefoundation.org/

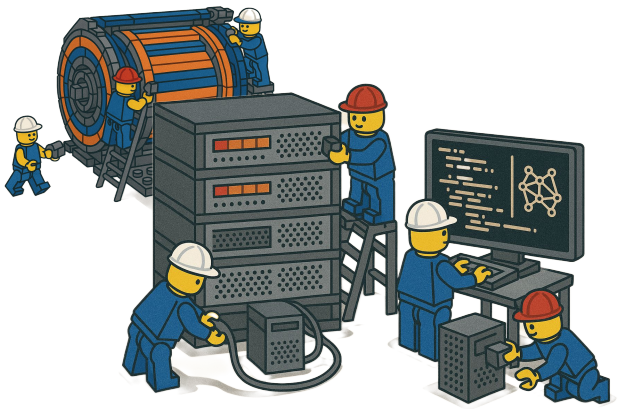
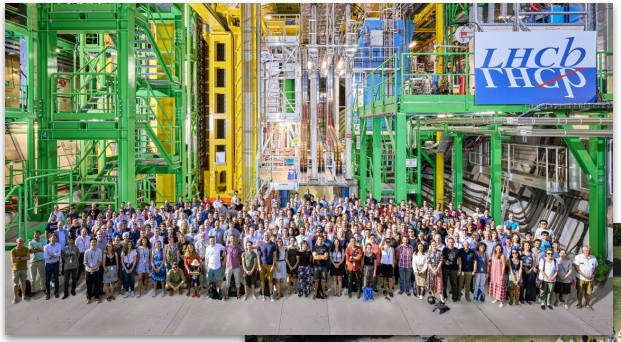


Figure: generated with DALL·E 3

Software Development in HEP

As a key for a sustainable scientific program

- Scientific collaborations are **big and growing**.
  - O(1K) collaborators in hundreds of institutes around the world.



Software Development in HEP

As a key for a sustainable scientific program

- Scientific collaborations are **big and growing**
  - O(1K) collaborators in hundreds of institutes around the world
- High Energy Physics (HEP) and Nuclear Physics (NP) are **computationally intensive** and **data driven** fields
  - A full physics potential requires investment into the software used to collect, process, and analyse data
- **Developers with strong foundation** are critical resources in the success of the current and future experiments
  - The researchers must be brought up to date with new software technologies, concurrent programming, and artificial intelligence
  - They must maintain, improve, and sustain the software

Training and Onboarding Initiatives in HEP

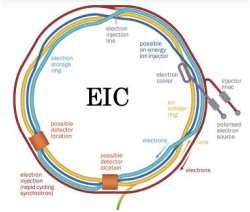
How do experiments teach software?

Virtual — Hybrid — In person

Online book

Starter Kit

Data Analysis Schools



Online tutorials

Software tutorials

Synchronous tutorials
“Carpentries-style”

“Software is different, but challenges are common”

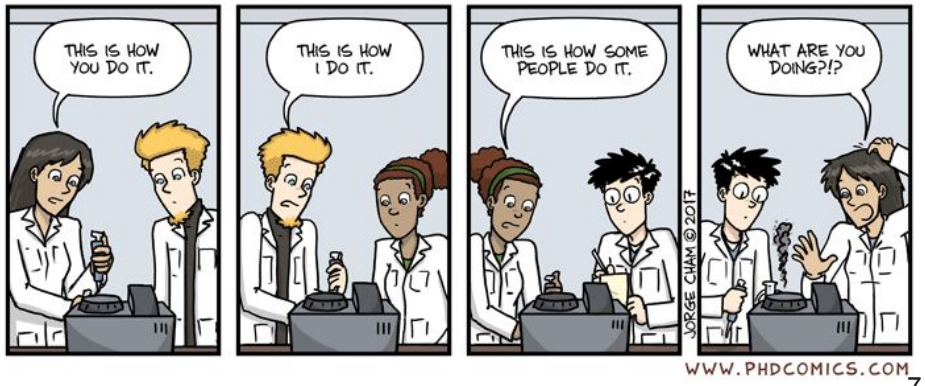
Front. Big Data 8:1497622. doi: 10.3389/fdata.2025.1497622

HEP Software Training

Why not leave everything to the experiments?

- **O(10k) HEP people** worldwide need to be trained in software engineering & computing
- Common challenges faced:
  - Most people developing code have non-permanent positions with **contracts of 2 - 4 years**
  - Material for training is a moving target as technology evolves (e.g., Deep NNs, GPUs, FPGAs, …)
  - Training activities are not the most popular in making career steps and by funding agencies

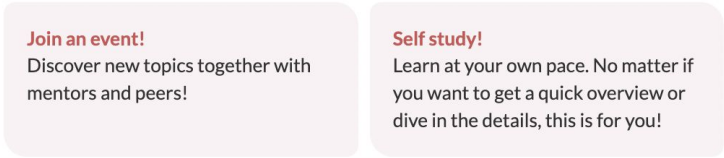
- **This should be a community effort!**



HSF Software Training

Organization

- Established in 2018
- An active community of members supporting training on voluntary basis 🫶
  - **Anyone is welcome to join and contribute!**
- Develops and teaches material for an introductory software curriculum
- Focuses on common software material across HEP, from basic core software skills to advanced training
- Remote weekly public meetings (via Zoom) to plan and assess progress
- Engages with different experimental collaborations and initiatives
  - IRIS-HEP, FIRST-HEP,The Carpentries, etc



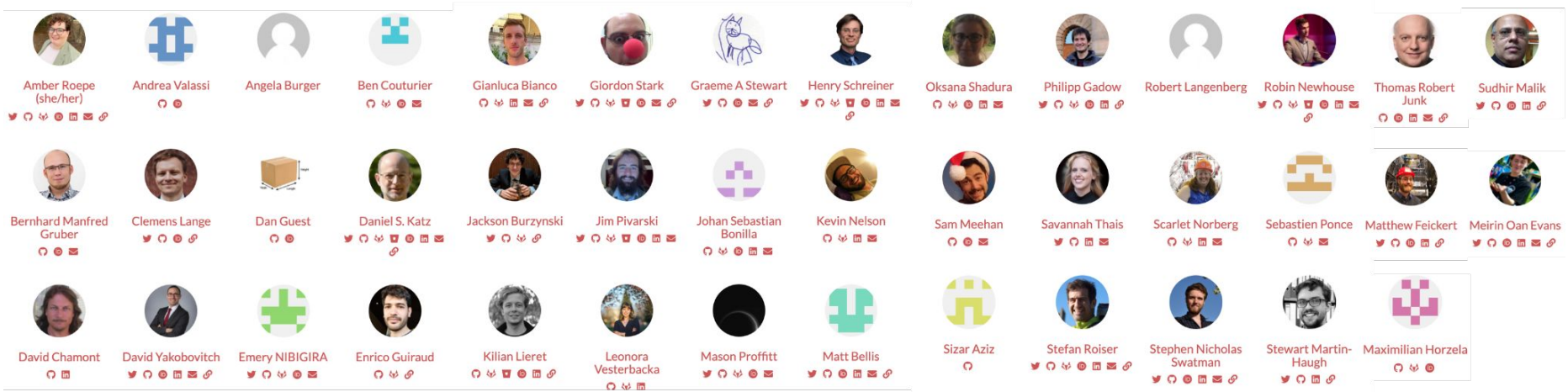
https://hepsoftwarefoundation.org/training

HSF
HEP Software Foundation

HSF Software Training

The community

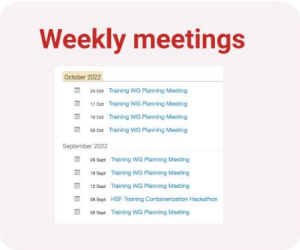
- An active community of members supporting training on voluntary basis 🫶
  - Coming from multiple collaborations, adding value to the training from different environments

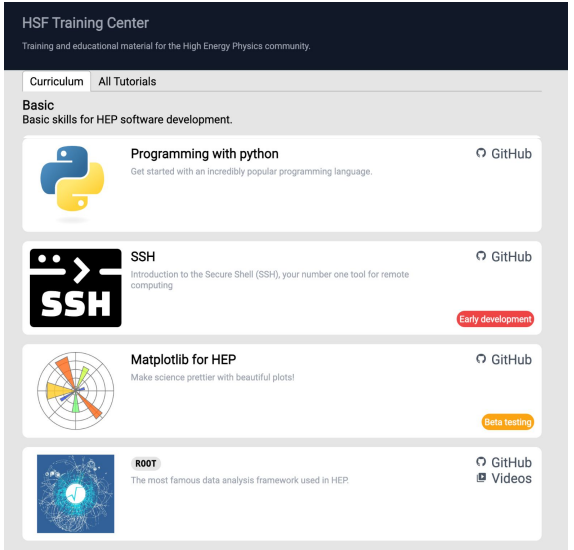


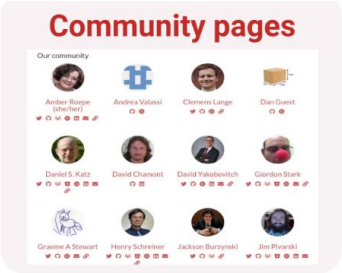
https://hepsoftwarefoundation.org/training/community.html

Training in the HSF / IRIS-HEP

- Significant progress in establishing common training across HEP and Nuclear Physics



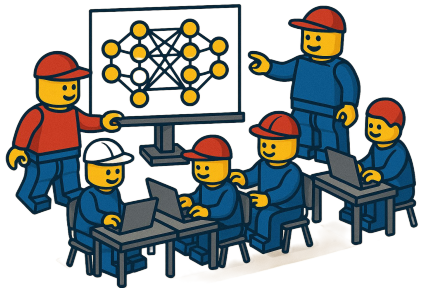




hepsoftwarefoundation.org/activities/training.html

https://hsf-training.org/training-center/

Developing a Curriculum



Developing a Training Program

- ADDIE model
  - Analyze
  - Design
  - Develop
  - Implement
  - Evaluate

Analyze Training needs

- Step 1: let's give some thought to ***why*** we want to develop a training program
- Training is a solution that can fix gaps in knowledge and skills

Current knowledge

- Basic shell commands
- Basic Python

Training!

Required

- Scripting in Bash
- Scientific Python
- Continuous Integration (CI)
- Docker Containerization

Training Goals

- Must be well defined and shared across all stakeholders: leaders, educators, students, funding agencies, etc.
- **When leaders and supervisors do not align with the training goals, it is difficult to put in practice the new knowledge and skills recently acquired**

- It is easier to determine if a goal has been met when the goal is written with SMART objectives
  - **S**pecific: clearly state what the learner will be able to do
  - **M**easurable: the outcome must be observable and quantifiable
  - **A**chievable: should be realistic given the learner's background and the time available
  - **R**esult-oriented: focus on the outcome
  - **T**ime-bound: specify a timeframe for achieving the objective ("by the end of this module")

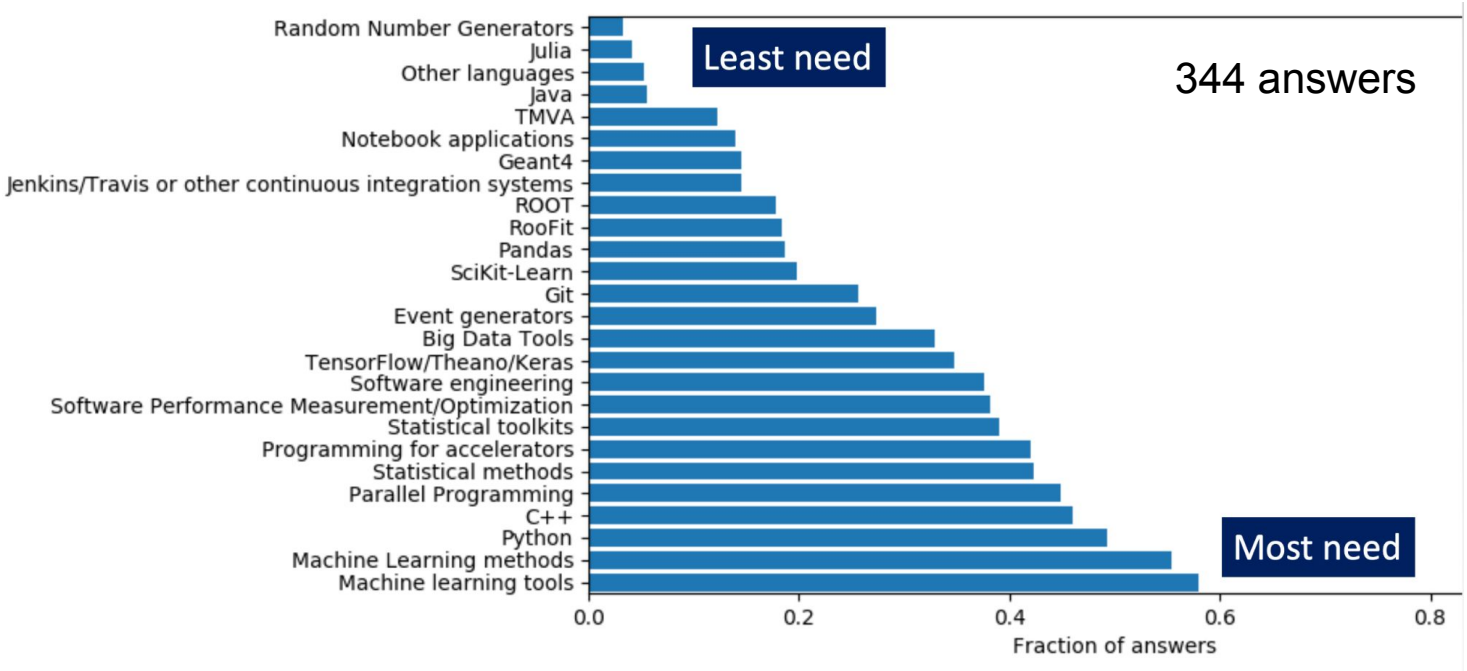
Analyze Training needs

- 4-5 volunteers
  - Your name
  - How are you using / planning to use deep learning
  - Training need and goal(s) in your organization

The HSF Training Survey

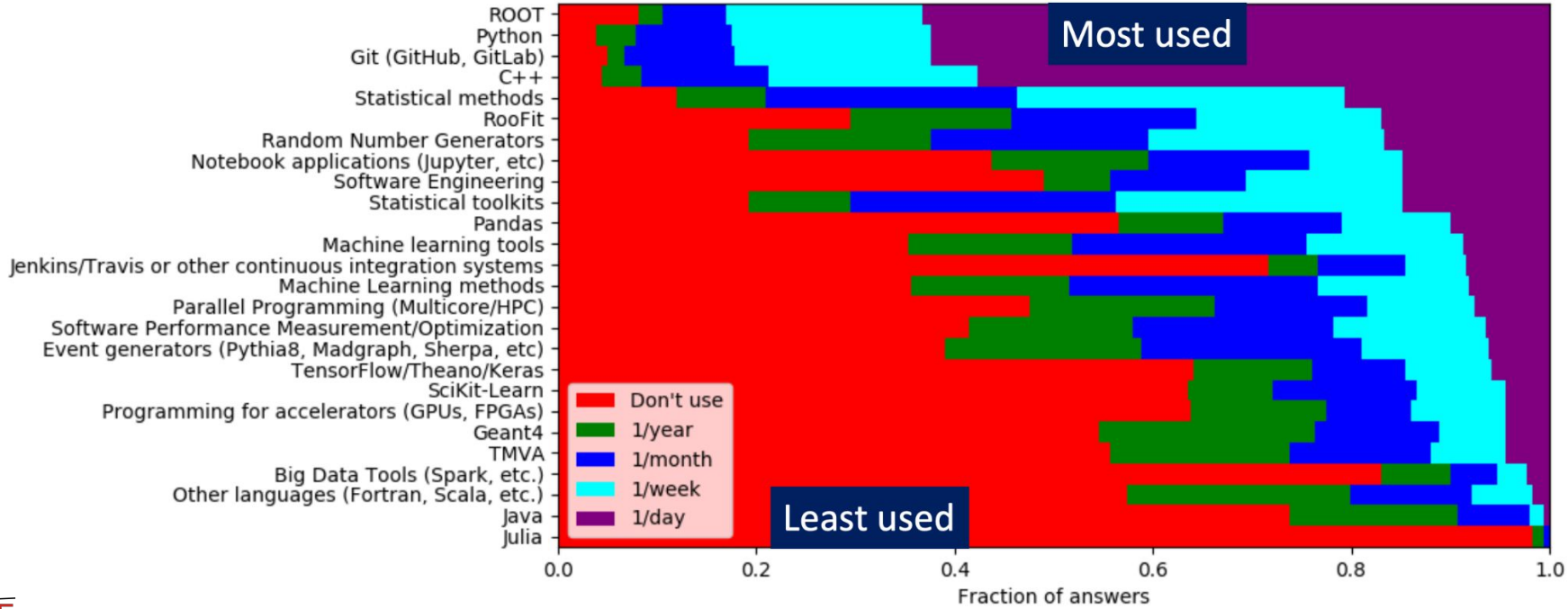
- Our own example: in 2019, a HSF training survey listed the needs of the community

  “Areas that need for more training materials/courses to help your research “



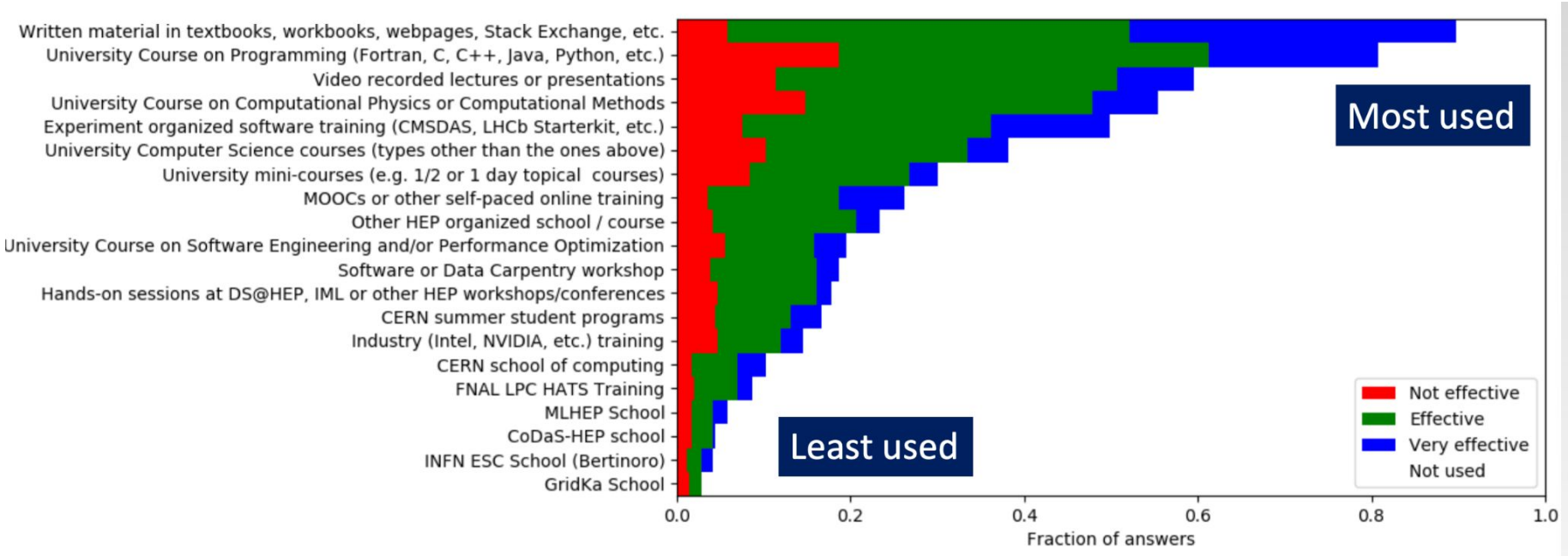
The HSF Training Survey

"Frequency of use of some scientific software/computing tools"



The HSF Training Survey

"Training sources that you have previously used for learning scientific software"



HSF Software Training Basics

- Since Aug 2021, in collaboration with The Carpentries, we have established training material for newcomers
- Fast-track competency with software fundamentals:
  - Bash, Git, Python
- Modules dedicated to HEP Software:
  - ROOT (data analysis framework),
  - Scikit-HEP (data analysis in Python)
- Hackathons have been organised to brainstorm, develop intermediate level material and discuss improvements
  - Naturally, the interest evolves as new technologies emerge (e.g. LLMs)

Designing Material for Adult Learners

- M. Knowles suggested 4 principles that are applied to adult learning

Adults need to be involved in the planning and evaluation

Experience (including mistakes) is the base for learning

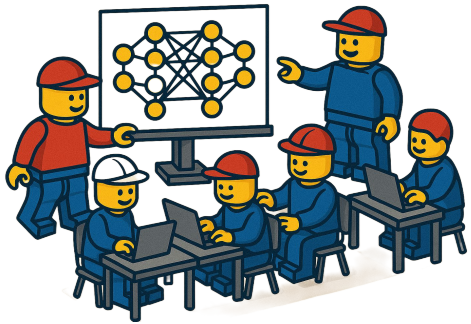
Problem-centered, rather than content-oriented

Subjects that have immediate impact in the work

https://elearningindustry.com/the-adult-learning-theory-andragogy-of-malcolm-knowles

The HSF-Training Design Philosophy

- **Hands-On**
  - Learning is achieved by doing
- **Experiment Agnostic**
  - Teach tools and techniques that are independent of a specific context
- **Student-Centric**
  - Acknowledging student voice as central to the learning experience
- **Reusable**
  - If resources already exist, use them
  - If they do not exist, develop them
- **Open and Accessible**



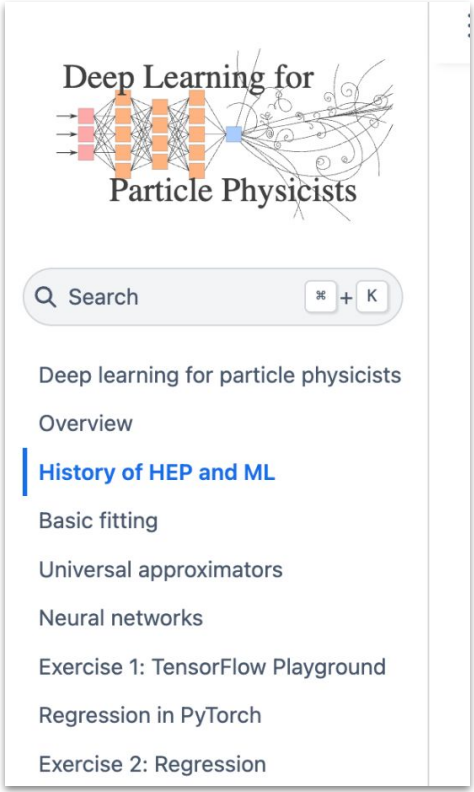
Designing Training Material

Key components to consider when creating materials for scientific software education

- Name and description
- Learning objectives
- Prerequisites
- Table of contents
- Instructional content, step by step
- Activities and exercises

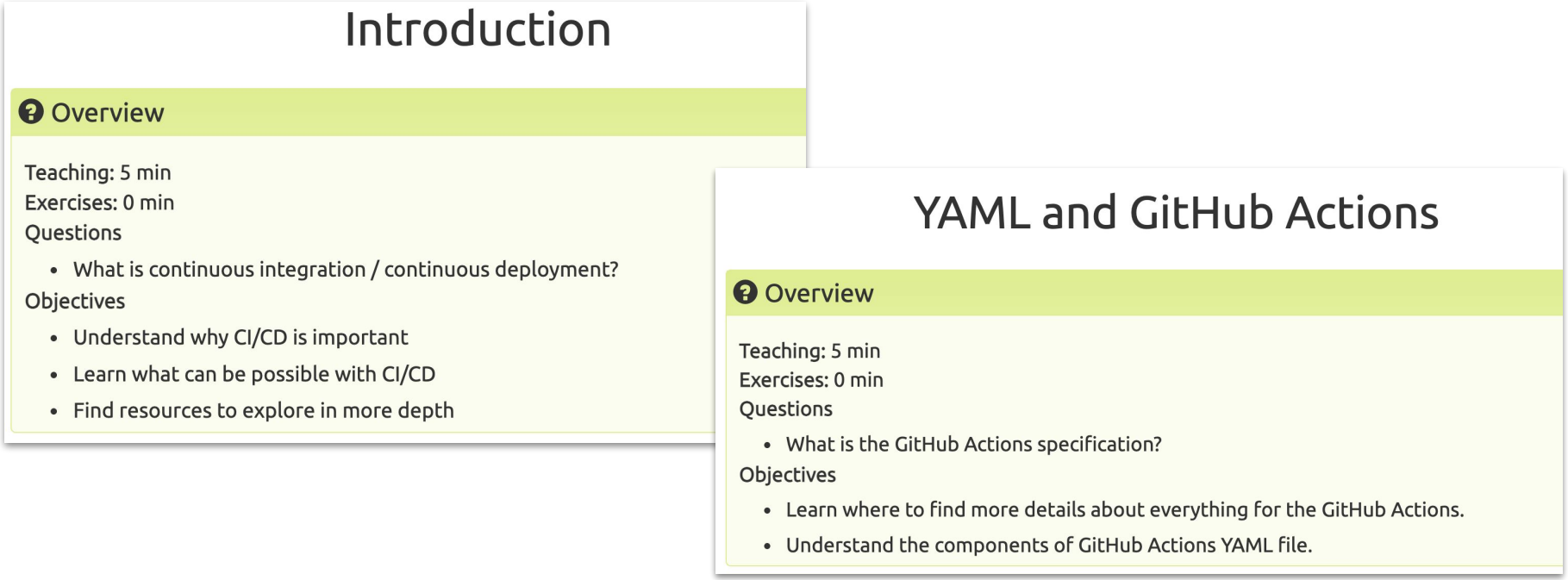
The HSF Training Modules

- The style and pedagogy is heavily inspired by The Carpentries
- Reusable study material that is open source and open access
  - Many licensed under CC BY 4.0: free to share, remix, transform, and build upon the material for any purpose
- Hosted in the HSF's Training repositories on GitHub
  - Using Carpentries styles and Jupyter Book
  - Anyone is welcome to open issues and pull requests



hsf-training.github.io/deep-learning-intro-for-hep/

The HSF Training Modules



Introduction

Overview

Teaching: 5 min
Exercises: 0 min
Questions

- What is continuous integration / continuous deployment?

Objectives

- Understand why CI/CD is important
- Learn what can be possible with CI/CD
- Find resources to explore in more depth

YAML and GitHub Actions

Overview

Teaching: 5 min
Exercises: 0 min
Questions

- What is the GitHub Actions specification?

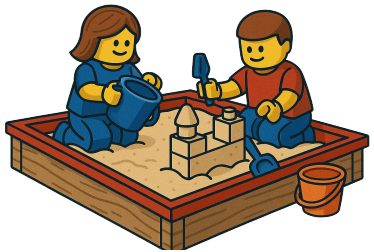
Objectives

- Learn where to find more details about everything for the GitHub Actions.
- Understand the components of GitHub Actions YAML file.

https://hsf-training.github.io/hsf-training-cicd-github/

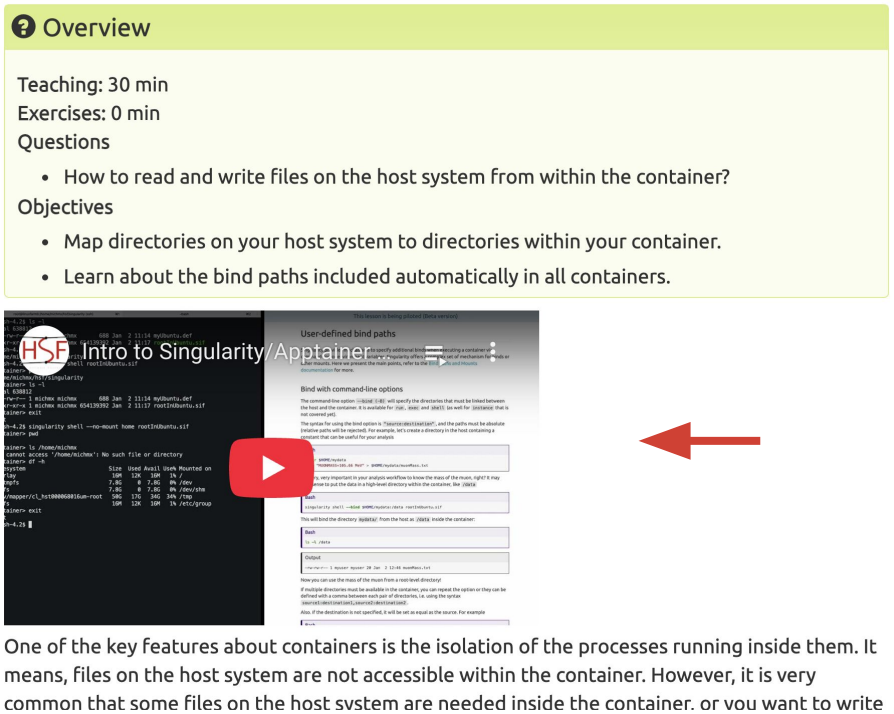
The Setup

- A proper setup provides the environment where participants feel **secure enough to experiment and make mistakes**
- **Minimize friction:** pre-configured environments reduce time spent on setup, keeping focus on the actual learning. A few useful tools
  - Virtual environments, etc.
  - Containers: provide an image with the required environment
  - GitHub Codespaces: full setup in a few clicks
    (example: GitHub codespace with MySQL)
- We found effective as well
  - Pre-recorded videos explaining how to setup (examples)
  - Booking an optional session on Indico to help with the setup



Videos / Recordings

- Complement the written material and contribute to higher information retention rates
- Learners can access video content anytime, anywhere, and on any device
    - Scalable, cost-effective in the long run
- We use screen recording, with edits that improve clarity, engagement, and pacing
    - Remove mistakes and long pauses
    - Invest in a good microphone! 🎙️

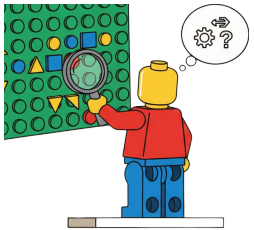


Examples and Exercises

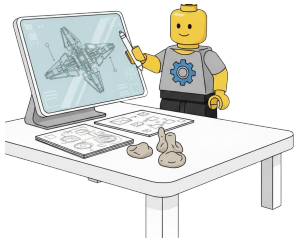
- Must be relevant and ideally have an immediate impact in the work of the trainee
  - Focus on (real!) problems in the HEP & Nuclear Physics context
  - Challenge on making them experiment-agnostic
- Some examples that we have found
  - Training on Apptainer:
    - A definition file for starting a Jupyter notebook with ROOT or Scikit-HEP tools available
  - Introduction to Databases (in development)
    - A metadata catalog with information of files like recorded in collisions
    - An introduction to Conditions Databases

Developing a Training Program

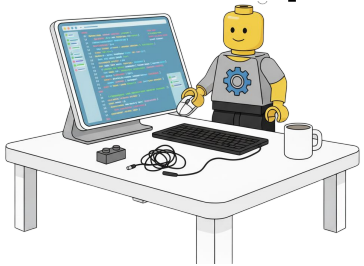
Analyze



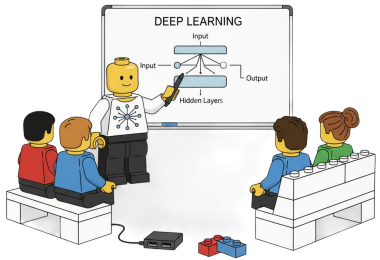
Design



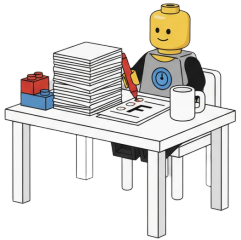
Develop



Implement



Evaluate



After the coffee break

A few thoughts about LLMs

- We are exploring LLMs for
  - Maintenance of our current training modules via AI agents
  - Updating outdated material, obsolete packages, etc
- The results out-of-the-box are really not that impressive. We are implementing Retrieval-augmented generation to provide more context

- In the long-term, it is likely that foundational models will enable generation of tailored training material, based on the current knowledge of the trainee and the training goals
- Probably the models will follow the same principles we just discussed

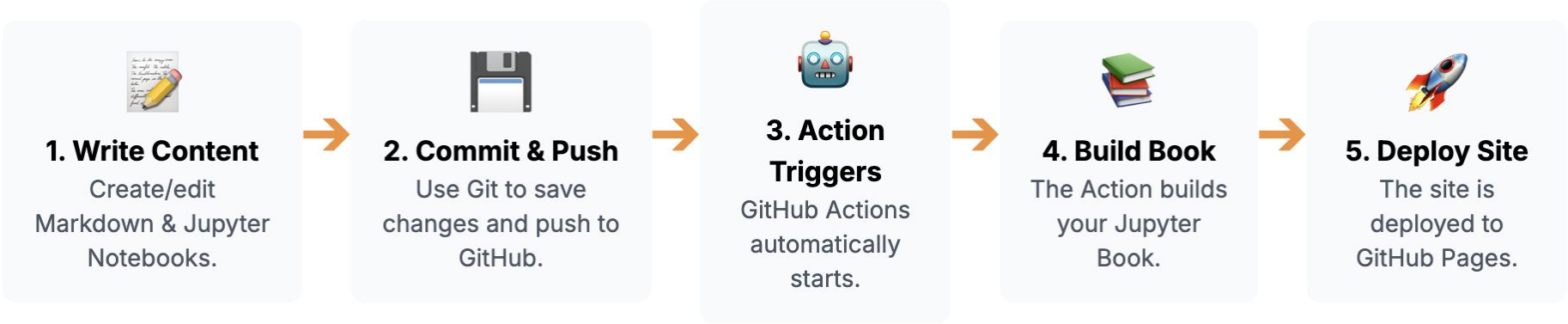
Summary

- The HSF shares its experience in training activities for curriculum development
- Training goals must be **SMART** and shared among all stakeholders
- The **ADDIE** model guides training program development: Analyze, Design, Develop, Implement, Evaluate
- The HSF-Training Design Philosophy emphasizes student-centric, hands-on, experiment-agnostic, reusable, and open content
- Proper setup and examples with **immediate impact** are crucial for effective training modules
- Creating training material and teaching requires a lot of commitment and time, it is therefore of great importance to **acknowledge the efforts** of everyone involved

Hands-on!

The Automated Development Workflow

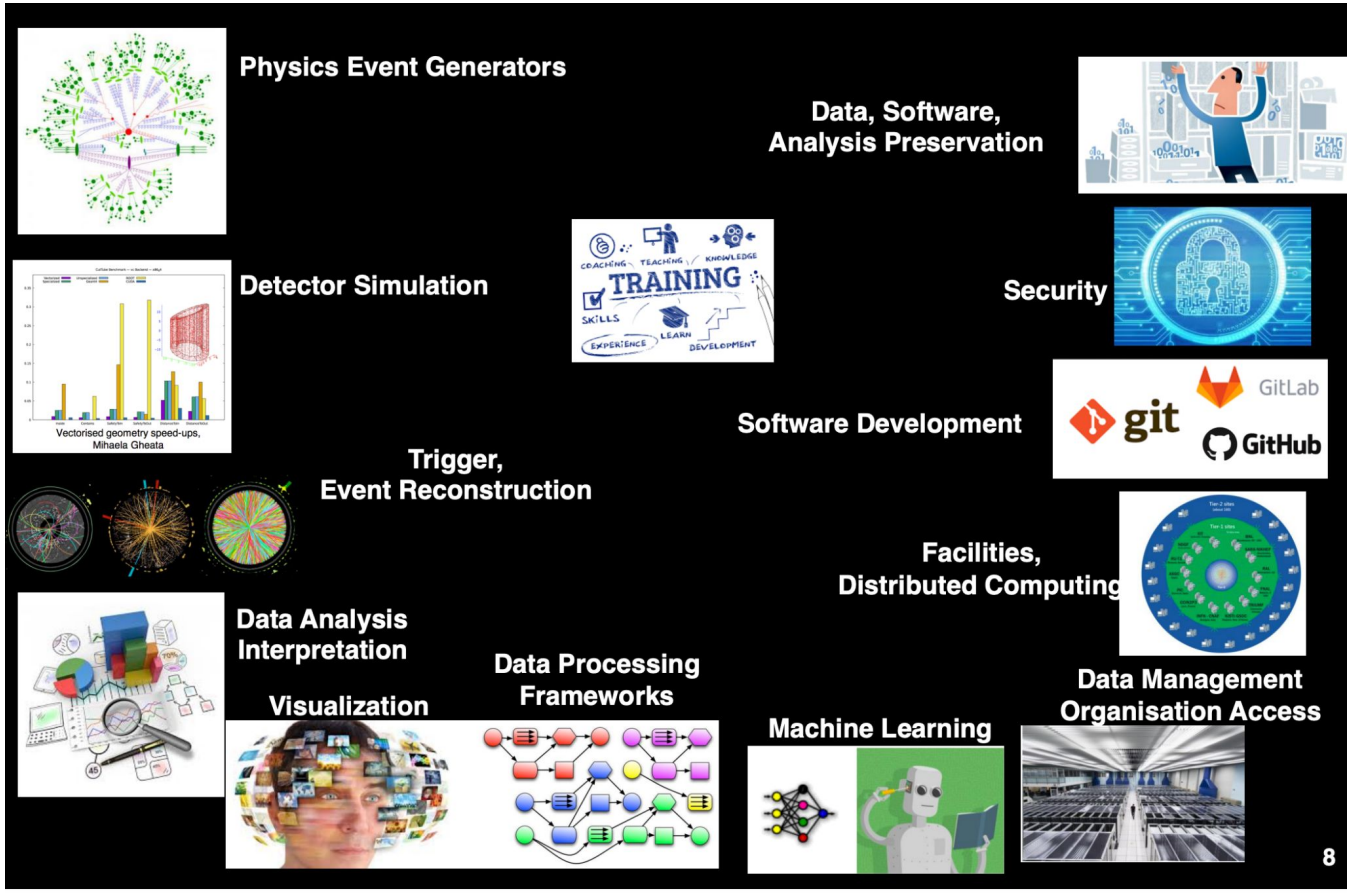
- We will automate the process of deploying training content, from local edits to a public website with minimal manual intervention
- Each push to the repository will trigger a build a deployment, ensuring the training is always up to date



https://michmx.github.io/2025-09-hsf-train-the-trainer/intro.html

Backup

The HEP Software Ecosystem

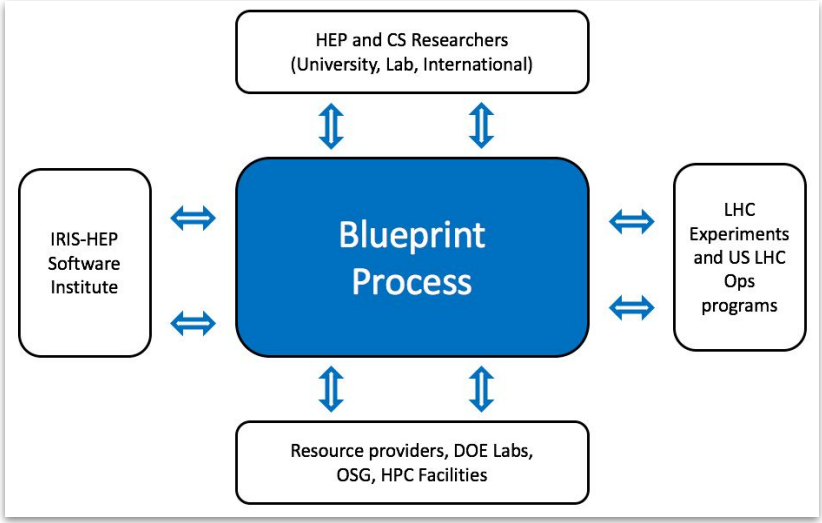


Blueprint Meetings

Blueprint Meetings

The Blueprint Activity is designed to inform the development and evolution of the IRIS-HEP strategic vision and build community consensus on the roadmap for the HL-LHC.

- Topics chosen by members of IRIS-HEP or external requests
- Most blueprints result in reports



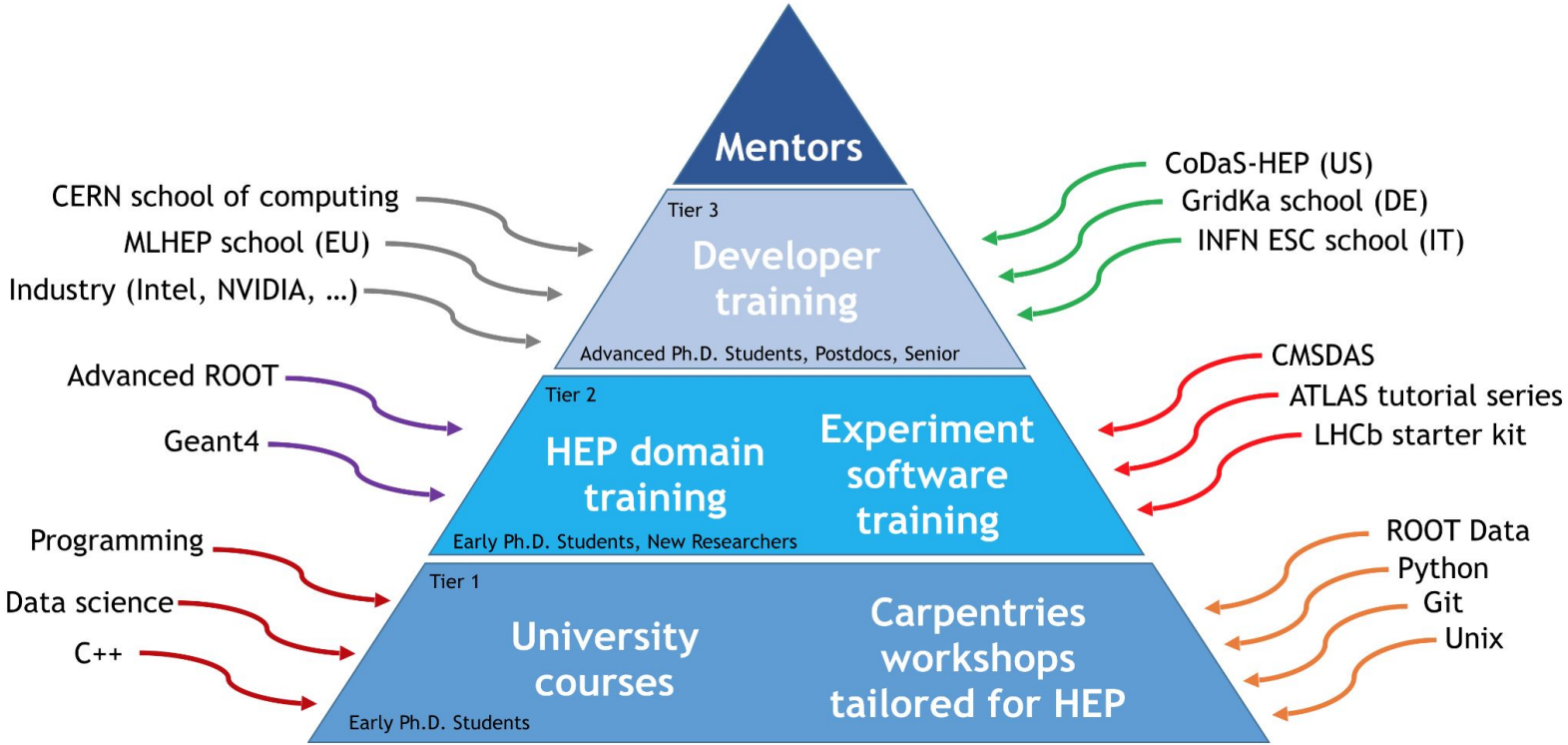
Completed (also see the top-level Indico page)				
2019				
Analysis Systems R&D on Scalable Platforms	AS, SSL	June 21-22	NYU	report
Fast Machine Learning & Inference	IA., SSL	Sept 10-13	FNAL	report
A Coordinated Ecosystem for HL-LHC Computing R&D	All	Oct 23-25	CUA	
2020				
Software Training	SSC	Feb 20	Virtual	report
Sustainable Software in HEP	SSC	July 22	Virtual	report
Future Analysis Systems and Facilities	AS, DOMA, SSL, OSG-LHC	Oct 26-27	Virtual	
Fast Machine Learning for Science	IA	Nov 30 - Dec 2	Virtual	report
Portable Inference	IA	Dec 4	Virtual	
2021				
Virtual Meeting on Virtual Meetings	All	May 5-6	Virtual	report
Differentiable Programming for the AS Grand Challenge	AS	Dec 1	Virtual	report
2022				
HSF/WLCG Analysis Facilities Forum Kick-off	AS	March 25	Virtual	AF Forum
A Coordinated Ecosystem for HL-LHC Computing R&D	All	Nov 7-9	DC	
Software Citation and Recognition in HEP	SSC	Nov 22-23	Virtual	

We've hosted 13 over the course of the institute's life

- Internal topics: Analysis Facilities, Differentiable Programming
- External Topics: FastML, Coordinated Eco-system in HEP with the funding agencies, Virtual Meetings, HSF Analysis Facilities,

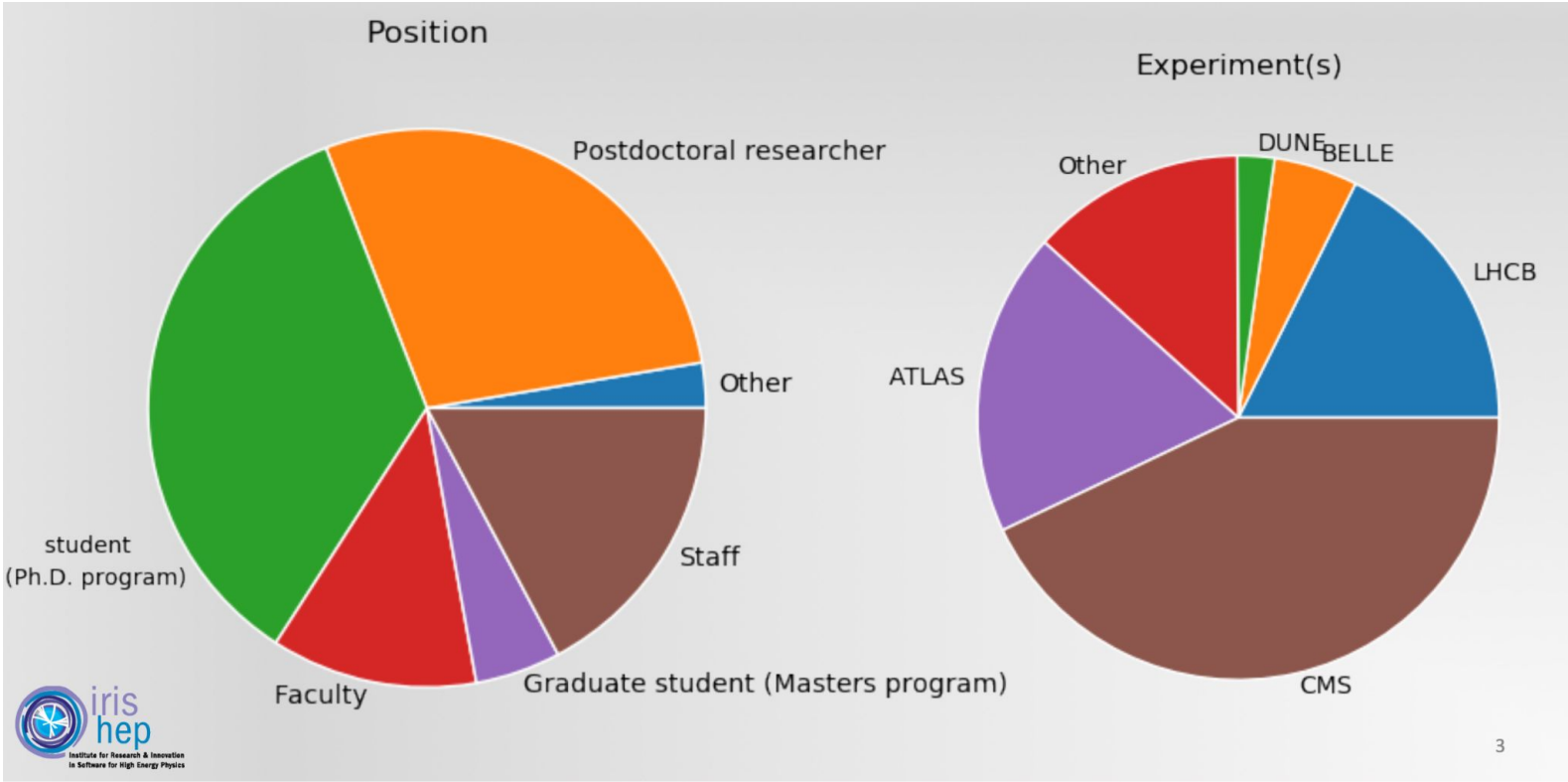
We will continue with our Coordinated EcoSystem Series and others we and the community need

The HSF Training Pyramid



https://iris-hep.org/ssc.html

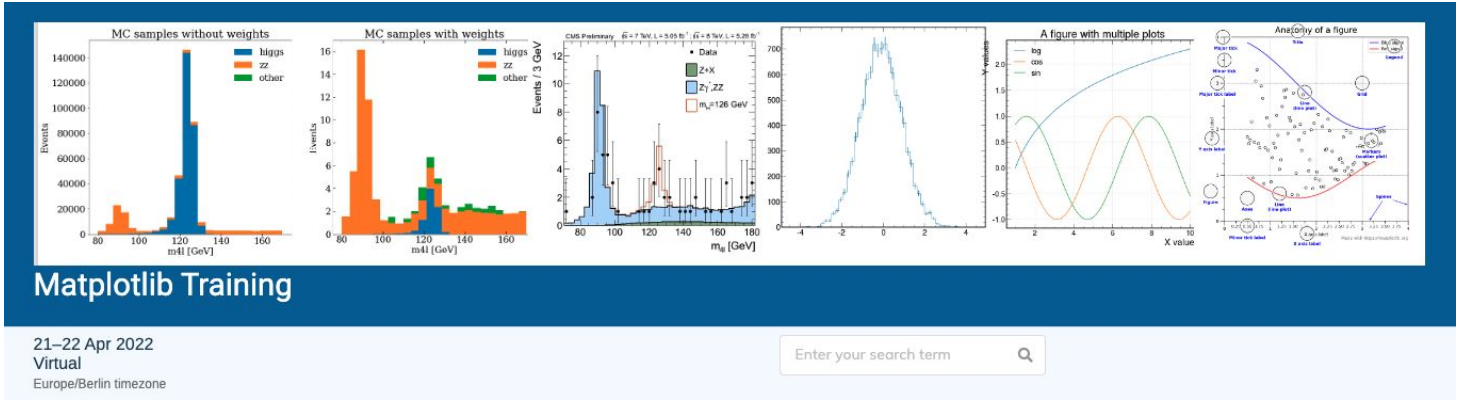
The Training Blueprint



Intermediate Training

Development and organization

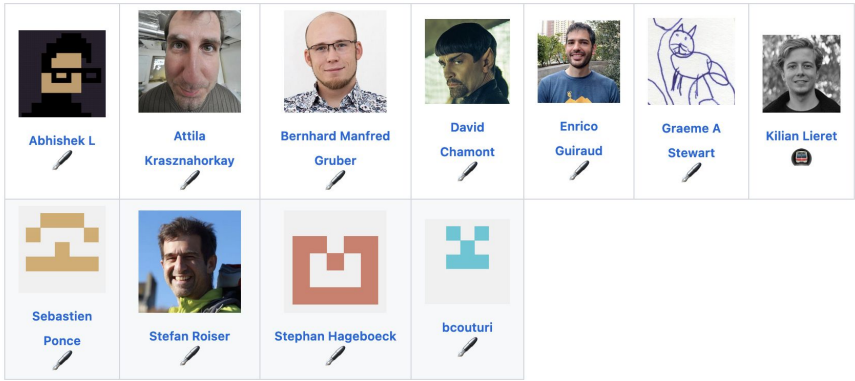
- Continuously organizing training meetings and hackathons for extending/improving material
- Established training events in C++ and Matplotlib
- Docker / Apptainer, CI/CD in GitHub and GitLab
  - Analysis Preservation Training events



The C++ Training

- Has been taught **in-person, virtual and hybrid**
- **Life lectures and exercise sessions**
- Full videos available
- Add link to Indico

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



SIDIS
Software Institute for Data-Intensive Sciences

CERN