

Developing Training

Hands-on!

Michel Hernandez Villanueva
Brookhaven National Laboratory

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



The HSF Hackathons

- An event where people come together for a short, fixed period to collaboratively develop new ideas, concepts, and prototypes
- The term hackathon combines
 - "Hack": solving problems or developing solutions
 - "Marathon": in the context of intense effort over a short timeframe

<https://en.wikipedia.org/wiki/Hackathon>



Some of the HSF “hacks”

Intro base OO More exp Tool conc py

HEP C++ course

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

CERN

March 10, 2023

1 / 542

HSF HEP Software Foundation iris hep docker A

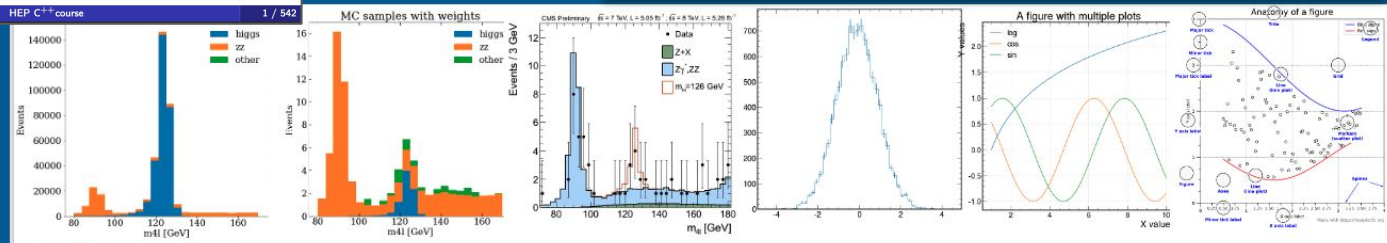
Training on Analysis Preservation (Virtual)

GitHub GitLab

16–21 Jan 2023
Virtual

Enter your search term

Learning the tools to make your analysis last to infinity and beyond!

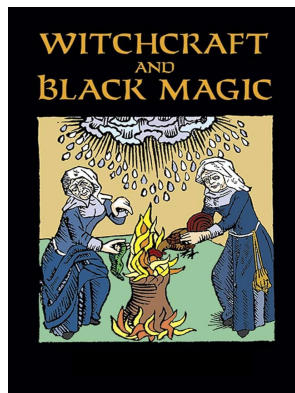
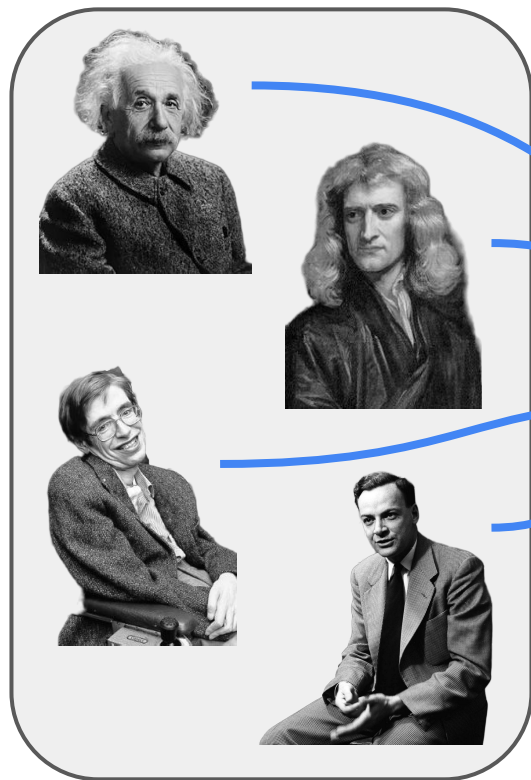


Matplotlib Training

21–22 Apr 2022
Virtual
Europe/Berlin timezone

Enter your search term

How the Hackathon works?



podman A reana

Introduction

Apptainer (formerly known as Singularity) is a free and open-source Apptainer called "containers" in a simple, portable, fast, and secure manner. container platforms are available, but Apptainer is designed to bring Computing (HPC) use cases. Using Apptainer, developers can work in environments that can be easily copied and executed on other platforms. This is an introduction to Singularity/Apptainer, its motivations and Based on the Apptainer user guide.

Prerequisites

- Basic knowledge of the Unix Shell, e.g., from the carpentry tool
- Access to a computing system with Apptainer/Singularity and access to CVMFS.

HSF Software Training

This training module is part of the HSF Software skills needed as they enter the field, and in parallel

Schedule

Time	Topic
00:00	Setup
00:00	1. Introduction

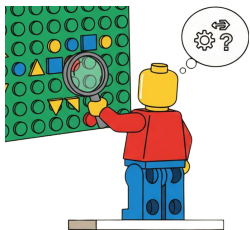
Intro to Singularity/Apptainer

HEP Software Foundation - 1/8

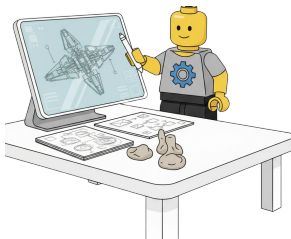
- Intro to Singularity/Apptainer #0 - Setup (5:48)
- Intro to Singularity/Apptainer #1 - Introduction (12:11)
- Intro to Singularity/Apptainer #2 - Containers and Images (13:58)
- Intro to Singularity/Apptainer #3 - Building Containers (27:24)
- Intro to Singularity/Apptainer #4 - Definition files (12:48)
- Intro to Singularity/Apptainer #5 - Sharing files between ho... (9:21)

How the Hackathon works?

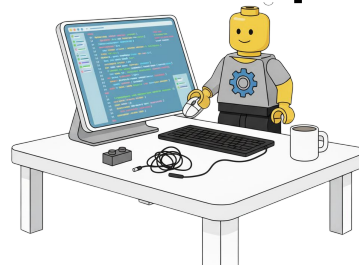
Analyze



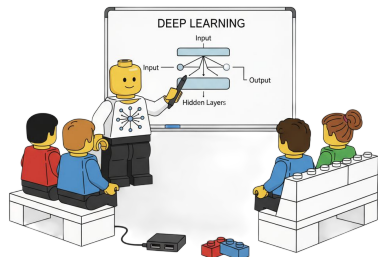
Design



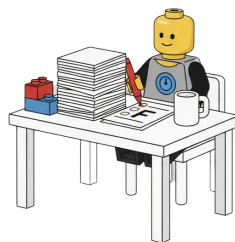
Develop



Implement

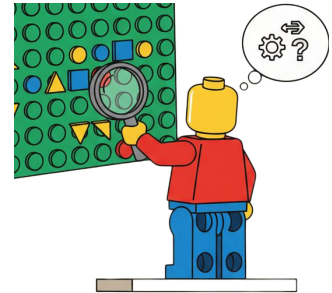


Evaluate

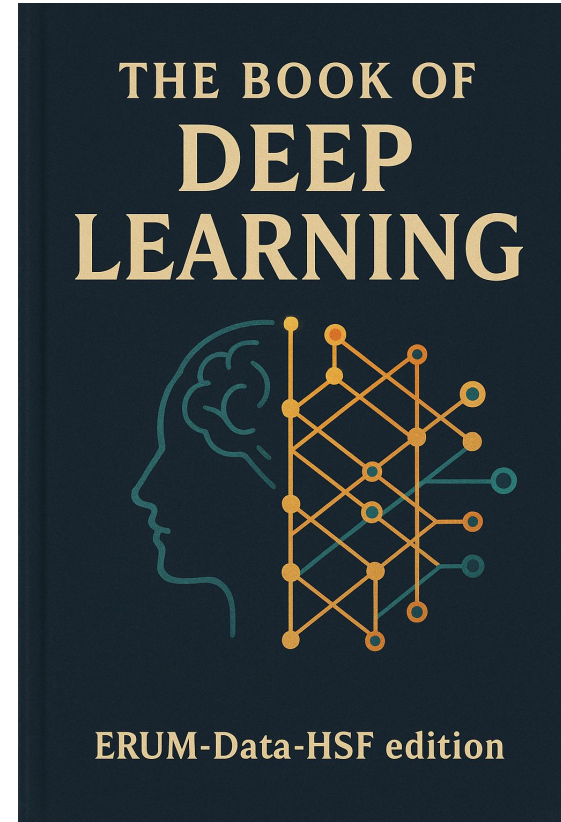


Analyze

- Usually we call for a Hackathon after we find the training need (topic)
 - Today you have a topic :)



The task for today



Analyze

Current knowledge

- Bash
- Scientific Python



Training!

Required

- Bash
- Scientific Python
- Training
- Inference
- ...

A few ideas for today

- K-Fold for training and validation
- Optuna for optimization
- Activation functions
- Inference with ONNX
- Your favorite Python package
- ...

Design

- Prerequisites
- Setup
- Questions and Objectives
- Chapters
 - Introduction
 - Main content
 - Summary / what's next
- Examples and exercises
 - Think about datasets (very important for Deep Learning)

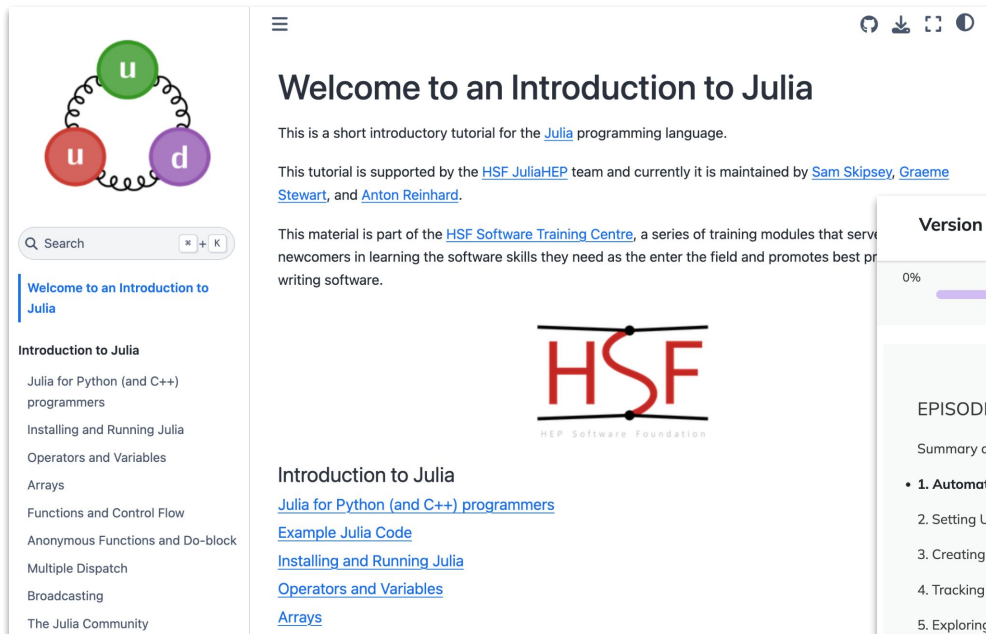
Questions

- What are the core SQL commands you need to know to interact with a MySQL database?
- How do you create, update, and delete records in a MySQL database?

Objectives

- Understand the basic SQL commands used in MySQL.
- Learn how to create a database and tables in MySQL.
- Practice inserting, updating, and deleting records in a MySQL table.

Develop



Welcome to an Introduction to Julia

This is a short introductory tutorial for the [Julia](#) programming language.

This tutorial is supported by the [HSF JuliaHEP](#) team and currently it is maintained by [Sam Skipsey](#), [Graeme Stewart](#), and [Anton Reinhard](#).

This material is part of the [HSF Software Training Centre](#), a series of training modules that serve newcomers in learning the software skills they need as they enter the field and promotes best practices in writing software.

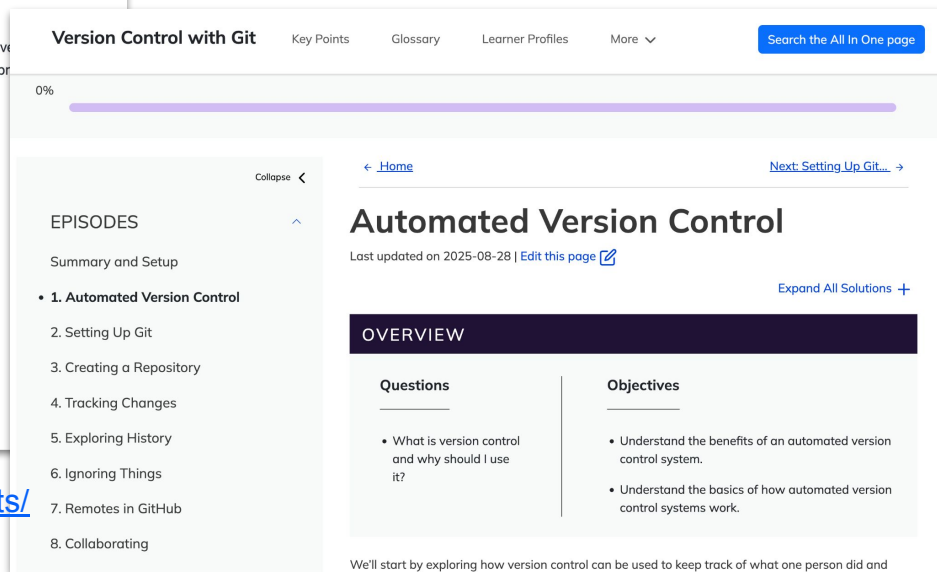
Introduction to Julia

- Julia for Python (and C++) programmers
- Installing and Running Julia
- Operators and Variables
- Arrays
- Functions and Control Flow
- Anonymous Functions and Do-block
- Multiple Dispatch
- Broadcasting
- The Julia Community

Introduction to Julia

- [Julia for Python \(and C++\) programmers](#)
- [Example Julia Code](#)
- [Installing and Running Julia](#)
- [Operators and Variables](#)
- [Arrays](#)

<https://juliahep.github.io/Hands-on-Julia-for-particle-physicists/>



Version Control with Git

Key Points Glossary Learner Profiles More

Search the All In One page

0%

EPISODES

- Summary and Setup
- 1. Automated Version Control
- 2. Setting Up Git
- 3. Creating a Repository
- 4. Tracking Changes
- 5. Exploring History
- 6. Ignoring Things
- 7. Remotes in GitHub
- 8. Collaborating

Automated Version Control

Last updated on 2025-08-28 | [Edit this page](#)

OVERVIEW

Questions	Objectives
<ul style="list-style-type: none">What is version control and why should I use it?	<ul style="list-style-type: none">Understand the benefits of an automated version control system.Understand the basics of how automated version control systems work.

We'll start by exploring how version control can be used to keep track of what one person did and

<https://swcarpentry.github.io/git-novice/01-basics.html>

Develop



Contents

[Purpose](#)

[Prerequisites](#)



My Jupyter Book

Q Search



[CNN Train-the-Trainer Workshop](#)

Introduction to CNNs (classic lecture)

Teaching Convolutional Neural Networks

Under the hood of CNNs (flipped classroom)

Flipped Classroom

Under the hood of CNNs

Manual Convolution Exercise

Exercise 1: Learn a Gaussian Filter with a Single Conv Layer (PyTorch)

U-Net (pair programming)

Pair programming

U-Net for Nuclei Segmentation (BBBC039)

Hyperparameter Tuning with Optuna



CNN Train-the-Trainer Workshop

Welcome! Use the left sidebar to navigate. Each notebook installs its own dependencies and can run on Colab (GPU).

Total course time: ~3 hours.

Purpose

The purpose of this jupyter book is twofold:

- A hands-on workshop to teach how convolutional neural networks (CNNs) work under the hood, and how to train them for image segmentation tasks.
- A set of materials for instructors to use in their own teaching, giving examples of classical lecture slides, flipped classroom lessons, and pair programming exercises.

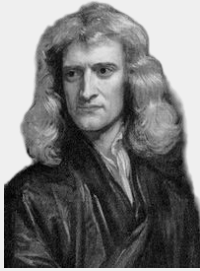
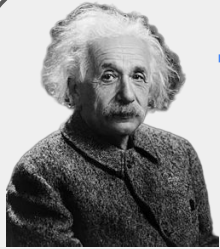
Prerequisites

- Basic Python experience, including Jupyter notebooks.
- Basic machine learning experience is helpful but not required.

Next

[Teaching Convolutional Neural Networks](#) >

Develop



Setup

1. Introduction

2. MySQL Basics

3. 03 Break

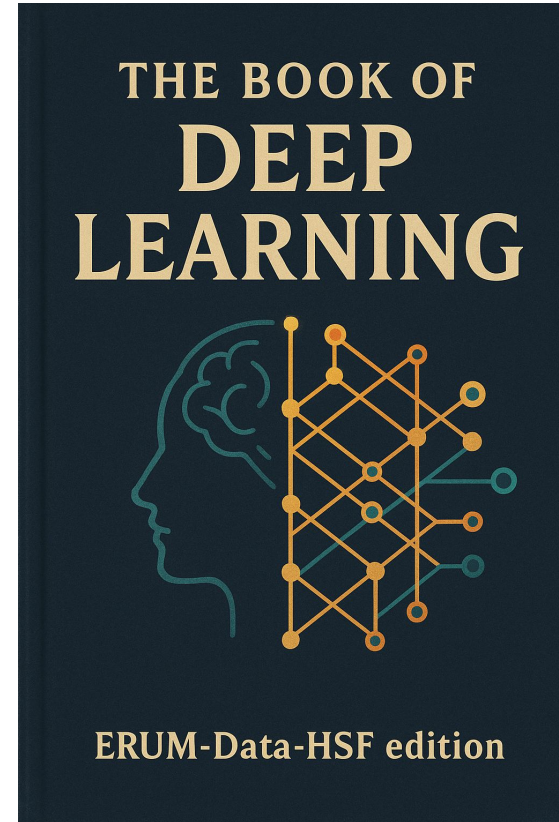
4. SQLAlchemy and MySQL: Exercises

5. Relations between Tables

Communication is key!

The teams

- Let's assign the task in 4 teams
 - Overlaps in content are OK!
- “Divide and conquer”
 - Easier to collaborate in small groups
- Each team will have a different approach on how to teach deep learning
 - Reduces the impact of the “curse of knowledge”
 - Adds value to the training



The teams



The teams

- Epoch Warriors
- ReLU Rebels
- Softmax Supremacy
- The Backpropagators

<https://github.com/orgs/erum-data-hsf/repositories>

Angela



The coaching staff

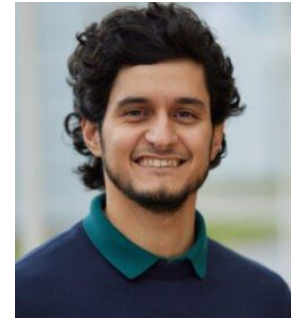
The “not that kind of training” team



The ErUM Data encyclopedia



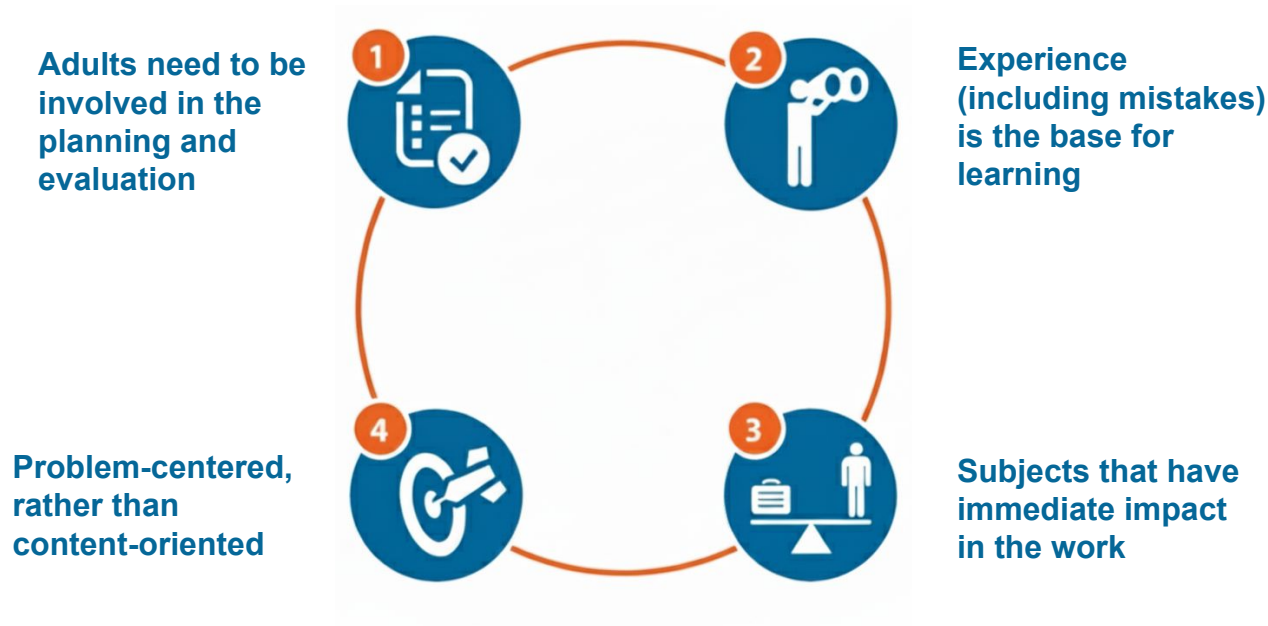
The deep learning wizards



Backup

Designing Material for Adult Learners

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



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

Required

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



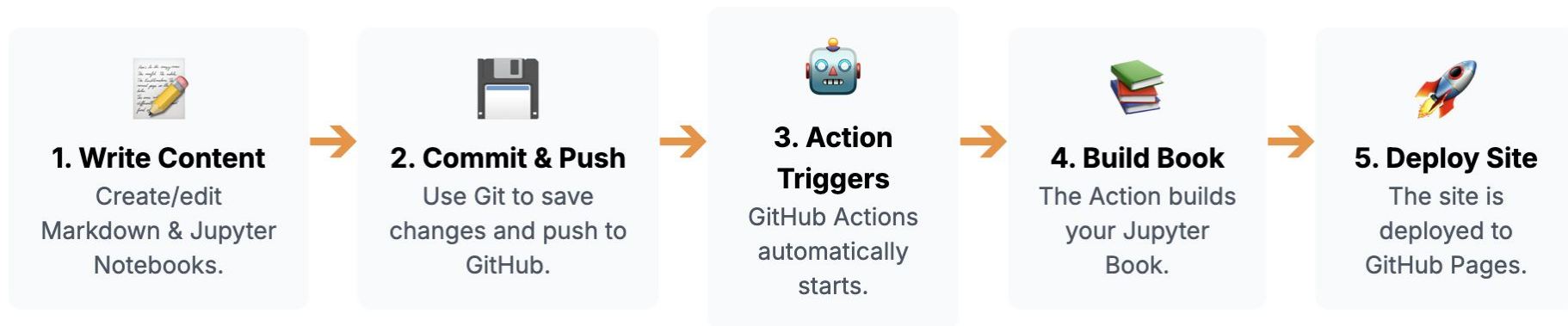
Training!

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")

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>

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**

