

Creating a lesson on unit tests for the HSF Analysis Preservation Training



Yarelis D. Acevedo Rios
University of Puerto Rico Mayaguez
April, 2023

1 Introduction

Testing the code is a crucial task that confirms that the code is working correctly and meets the quality standards for customers and, in this case, for analysis. Automated tests are important to confirm that code is working correctly. There is a wide amount of testing frameworks, such as unittest, Pytest, doctest, nose2, Testify, and Hypothesis. Unittest, for example, is a built-in unit testing on Python, and the other frameworks are third-party. For this, we are going to concentrate on Pytest.

Why not test using the built-in framework that Python offers (unittest)? Unittest requires developers to create classes derived from the TestCase module and then define the test cases as methods in the class. [1] This means that there is a lot of boilerware code required to execute the tests, in contrast to Pytest, which is more condensed and easier to write. Additionally, Pytest is more informative in case of a failure, which is useful for debugging.

Pytest is easier to learn than its counterparts, by using Python functions and appending them, or using classes for groups. You can also use it by filtering out tests to get rapid feedback and continue maybe fixing other problems or building more of the framework. And last, but not least, it offers parameterization, which uses a decorator which allows us to define a single test definition, and then Pytest will test the various parameters we specify for us.

2 Project Proposal

The HSF Training group, together with IRIS-HEP started the training for analysis preservation, and part of the skills that are taught is Pytest. The aim of this proposal is to read and evaluate the carpentry incubator training, Python Testing and Continuous Integration [2] and use its tutorial format to contribute on the analysis preservation training. I've been familiarizing myself with Pytest to learn more tips and use my learning curve to in order to contribute and improve the analysis preservation training. That said, after improving the carpentry lesson, I will be proposing and discussing good ways to include training on Pytest into the analysis preservation training. Finally, creating a training module and recording videos for improving the experience of learning. I propose to work half-time for 7 - weeks.

3 Timeline

Week(s)	Tasks
1-2	Project Setup <ul style="list-style-type: none">• Familiarize myself with Pytest and existing training material. In particular, considering the training module from the carpentry incubator [2] and the TAC-HEP training module [5]• Read the existing workshop material on github actions [4] and gitlab CI [3].• Spot and correct mistakes if they come up. Learn how to locally builds carpentry-style training websites and learn how to create pull requests.• Learn how to locally builds carpentry-style training websites and learn how to create pull requests.
3	<ul style="list-style-type: none">• Put together concrete plan of how the content of a pytest training module should be structured and present it to training group. In particular, it needs to fit with the existing material on github actions and gitlab CI.• Help with maintenance work on several training websites (fixing broken links, standardize content, etc.) Put together.
4, 5, 6	Work on training module website
7	Finish first version of pytest training module

4 About myself

I started my undergraduate research on CMS Experiment (at the LHC, Switzerland) with Prof. Sudhir Malik in August 2018, where I grew my interest in

programming languages, so I trained myself in Python and C++. Additionally, I have teaching credits in physics. Went into an internship at Cornell University in the summer of 2019 to work on the construction of the CMS Phase-2 Tracker (Pixel) Project. There, under the supervision of Prof. Jim Alexander and Engineer Yadira Bordlemay, I designed and developed a tensile test machine to test it out on carbon fiber strips to see if they are good material for the Dees. The Dees are located on the inner tracker on the CMS and serve as a mechanical structure to populate several silicon pixel sensor modules. I am the co-founder (with Guillermo Fidalgo, another student at UPRM) of a student club named Scientific Software Club. I have given C++ workshops on campus and in high schools to show students the importance of coding languages and how to use them for problem-solving. I continued to give workshops online to high school students in the COVID-19 era. During the summer of 2020, I worked (remotely) with Sudhir Malik and the research group on a K-12 teacher's workshop to show them how to program Python and how they can implement the programming language in their classroom with different experiments. After the workshop, I participated in the Fermilab C++/STL class. During these intensive classes, I learned key functions that are used in Fermilab and the libraries made by Fermilab in C++ to be able to analyze data. After the Fermilab C++/STL class, I started working again remotely (due to COVID-19 and my semester at UPRM) with Cornell University, but this time building and programming a tube bending machine. After this, I went to Cornell University to work on the development of the deposition and characterization of TIM (thermal interface material) for the Dees. Later on, I was offered to work as a COOP at Cornell during the fall semester of the same year, working on the same project. Finally, in the summer of 2022, they selected me for the SUPER ATLAS program, where I got my hands-on analysis on VLQ+X (Search for pair production of a new heavy quark that decays into a W boson and a light quark in pp collisions at $\sqrt{s} = 8$ TeV with the ATLAS detector). I was working on a framework for the analysis using ROOT and Python for it, and I refreshed my knowledge in git and GitHub.

References

- [1] Datacamp. *How to Use Pytest for Unit Testing*. URL: <https://www.datacamp.com/tutorial/pytest-tutorial-a-hands-on-guide-to-unit-testing>.
- [2] HSF training group. *Python Testing and Continuous Integration*. URL: <https://carpentries-incubator.github.io/python-testing/>.
- [3] HSF Training group. *HSF Training CI Gitlab*. URL: <https://github.com/hsf-training/hsf-training-cicd>.
- [4] HSF Training group. *HSF Training cicd Github*. URL: <https://github.com/hsf-training/hsf-training-cicd-github>.

- [5] Romain Teyssier Henry Schreiner. *Intro to testing*. URL: <https://henryiii.github.io/se-for-sci/content/week2/testing.html>.