Introduction to Research Software Engineering with Python

Hello, and welcome to Research Software Engineering with Python!

The goal of this course is to help you improve how you write scientific software: not just writing the code itself, but the whole process around it, how you work and how you think about it.

Software is becoming more and more used throughout all research areas. To maintain the integrity of your work, it is important that the code you write is robust and sustainable. The course will teach you best practices for ensuring that your code is correct, easy to modify in the future, and useable by others.

The things you learn will be applicable to any scientific domain. Similarly, this course is open to everyone, whether you are a starting PhD student or a tenured academic. The one thing we do expect is that you have some coding experience, as we will not be teaching the basics of programming. The examples we use are in Python, although the lessons are equally applicable to other languages - if you are not familiar with the language, you may want to first look at some tutorials or our introductory online course, so that you can follow along.

This course touches on a number of different topics.

The first thing we will look at is how to use Git and GitHub to keep an archive of all the revisions you have made to your code, to easily undo mistakes that you make, switch between different versions of your code, and collaborate with others.

The next thing we will discuss is testing your code, to make sure that it actually does what you intend it to do. We will look at how to write tests, and how to use automated frameworks to stop you accidentally introducing bugs when making changes.

After that, we will discuss how to write a library and make it available for others to install and use. We will then look at structuring your code so that you or others can extend it more easily in the future, and how to think in an object-oriented way. We will also touch on some more advanced Python topics, and discuss how and when you can optimize your code to run faster.

We will be illustrating all of the above with code examples, which we encourage you to download and explore on your own, and there are also quizzes to check your understanding of the topics we discuss.

Like many things, learning to code is a journey, and this course should show you how to take your first steps beyond the basics.
After completing it, you should be ready to go and work on your own scientific software packages, whether from scratch or by contributing to existing work.

On behalf of everyone who has contributed to this course, we hope you find it useful. Enjoy!