Training 4 Programmers

To discuss this course and customizations:
Call: +1 434-509-5680 or Email: sales@training4programmers.com

High Performance Python with Rust

Duration

2 days

Description

This advanced course is designed for computer programming professionals seeking to enhance their skills in Python and Rust. The course covers a wide range of topics including debugging, profiling, interfacing Python with Rust, building Rust extensions, Python and Rust integration, and distributing Rust extensions. Participants will learn how to debug Python and Rust code using Visual Studio Code, profile Python applications, build Rust extensions, and distribute them. They will also gain a deep understanding of Python and Rust integration, exploring topics like Python classes in Rust, Python exceptions in Rust, Python GIL and releasing the GIL, threading, and async/await in both languages.

Objectives

- Master the art of debugging Python and Rust code using Visual Studio Code.
- Understand the differences between CPU-bound and IO-bound tasks and learn how to choose the best option for your application.
- Gain proficiency in profiling Python applications, detecting memory leaks, and benchmarking Python/Rust code.
- Learn how to interface Python with Rust using PyO3, and compare Rust and C extensions.
- Build Rust extensions using PyO3 Crate and Maturin Tool, and learn how to import Rust extensions into Python.
- Dive deep into Python and Rust integration, exploring topics like Python classes in Rust, Python exceptions in Rust, Python GIL and releasing the GIL, threading, and async/await in both languages.
- Understand how to distribute Rust extensions, including production build, building a wheel, and publishing to PyPi.
- Develop a comprehensive understanding of Python module & submodules programmed in Rust, function options, function signatures, type conversions, and declarative modules.

Prerequisites

All students should have significant experience with the Python and Rust languages. The class has no review of the Python or Rust programming languages. If you need to learn Rust programming, please consider the Rust for Python Programmers course.

Training Materials

All students receive comprehensive courseware covering all topics in the course. Courseware is distributed via GitHub in the form of documentation and extensive code samples. Students practice the topics covered through challenging hands-on lab exercises.

Software Requirements

Students will need a free, personal GitHub account to access the courseware. Students will need permission to install Python, Rust, and Visual Studio Code on their computers. Also, students will need permission to install Python Packages and Visual Studio Extensions. If students are unable to configure a local environment, a cloud-based environment can be provided.

Outline

- Getting Started
  - Debugging Python and Rust code with Visual Studio Code
  - Profiling Python Applications
  - Speed Gains of Rust Extensions
  - CPU-bound vs. IO-bound
  - How to Choose the Best Option
- Debugging
  - Compile the Rust Extension with the Debug Options
  - Debugging Tactics
- Profiling
  - Reference Counting and Memory Leaks
  - Build a Debug Version of Python
  - Detect Memory Leaks with Valgrind
  - Perform Sample Profiling with py-spy
  - Benchmarking Python/Rust Code
- Interfacing Python with Rust
  - PyO3
  - Python calling Rust
  - Rust calling Python

  - Compare Rust and C Extensions
- Building Rust Extensions
  - PyO3 Crate
  - Maturin Tool
  - Basic Structure of a Rust Extension
  - Compiling Rust Extensions
  - Import Rust Extensions into Python
- Python and Rust Integration
  - Python Module & Submodules programmed in Rust
  - Declarative Modules
  - Python Functions programmed in Rust
  - Function Options
  - Call Python Functions in Rust
  - Function Signatures
  - Type Conversions
  - Python Classes in Rust
  - Python Exceptions in Rust
  - Python GIL and Releasing the GIL
  - Threading in Python and Rust
  - Async/Await in Python and Rust
- Distributing Rust Extensions
  - Production Build
  - Building a Wheel
  - Publishing to PyPi