

High-Performance Python with C Programming

Download Whitepaper: Accelerate Your Modernization Efforts with a Cloud-Native Strategy
Get Your Free Copy Now

Course Number: PYTH-278

Duration: 3 days

Overview

Course Description

Ascendient Learning's High-Performance Python with C training course focuses on supporting legacy C code for Python Extensions and writing new highly performant code with Cython. This course also teaches attendees how to employ the NumPy C API to make NumPy arrays available in C routines.

Skills Gained

- Profile Python applications to identify performance bottlenecks
- Understand the speed gains of Cython and C extensions and how to choose the best option for a given application.
- Interface Python with C using ctypes FFI, C extensions, and Cython
- Debug Python and C code using VS Code
- Use the NumPy C API

Prerequisites

Attendees must have completed Ascendient Learning's [C Programming for Python Developers](#) course or have equivalent experience, including many years of continuous Python and C experience.

Training Materials

All Advanced C/Python training students receive comprehensive courseware.

Software Requirements

- A virtual machine (VM) with all tools pre-installed will be provided.
- Students will receive setup instructions for their local machine, but no support will be provided in class to get it working if it has problems.
- Students will need RDP or SSH to access the VM; even those planning to work locally need remote access if their local setup has problems.

Audience

Course Details

Outline

- Introduction
- Profiling and Debugging
 - Debugging Python and C code with VS Code
 - Profiling Python Applications
 - Speed Gains of Cython
 - Speed Gains of C Extensions
 - CPU-bound vs. IO-bound
 - How to Choose the Best Option
- Interfacing Python with C
 - CTypes FFI
 - C Extensions
 - Cython
- CTypes FFI
 - Compile C code to a Shared Object
 - Load Shared Object into Python
 - Call C Functions from Python
 - Python Structure Class

- Python Binary Formatted Strings
- C Extensions
 - Learn to use the Python C API documentation
 - Basic Structure of a C Extension
 - Compiling C Extensions
 - Import C Extensions in Python
 - Packaging C Extensions
 - Using PyObject
 - Managing Memory and Reference Counting
 - Challenges of Multithreaded Programming and the GIL
 - Challenges of Asynchronous Programming
- NumPy and C Extensions
 - Learn to use the NumPy C API documentation
 - Using NumPy Arrays with C Code
 - Passing NumPy Arrays to a C Extension
 - Creating NumPy Arrays in a C Extension
 - Using the NumPy C API
- Cython
 - What is Cython?
 - Installing Cython
 - Compile Python Code
 - Python Code Decorations with Cython
 - Create Python Extensions with Cython
 - Challenges of Multithreaded Programming and the GIL
 - Challenges of Asynchronous Programming
- Conclusion