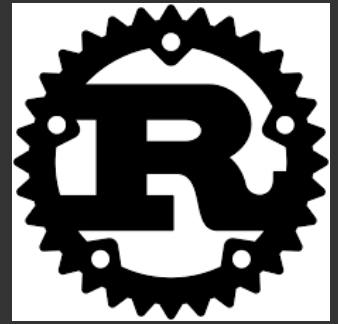


# ADVANCED RUST TRAINING



This advanced training is aimed at Rust developers who wish to discover the most advanced aspects of the language, such as advanced aspects of traits and ownership, which pave the way for discussing more advanced notions, such as encapsulation, concurrency, collections, etc. All sections include exercises that participants solve during the session, with the help of the trainers.

## TRAINERS

### Arthur Carcano (Rust)

Arthur is an R&D developer at OCamlPro. Arthur obtained his PhD in statistical methods at Institut Pasteur in 2022. Arthur has been a daily user of Rust for more than 5 years, whether for numerical computations during his doctorate as much as for system tools that he develops for our projects.

## TRAINING PROGRAM

### Warm-up

- Global state: const and static
- The Turbo fish

### Going further with Traits

- The Trait coherence rule
- Fundamental Traits of the standard library
- Traits and lifetimes (Add, Mul)
- Advanced trait bounds

### Advanced Ownership

- Diving into borrowing rules
- Lifetime subtyping
- Storing references
- Implementing traits for references

### Smart Pointers

- Reminder: the heap and the stack
- Box
- (A)Rc

### Fearless Concurrency

- Send and Sync
- Launching threads
- Scoped threads

### Dynamic Dispatch

- Trait objects
- Heterogeneous collections
- Trait objects and lifetime bounds

### Async Rust

- Introduction: asynchronous programming
- Under the hood: Future and Waker
- Pinning memory

### Unsafe Rust and FFI

- The powers of unsafe
- Raw pointers
- Unsafe traits and functions
- Basics of FFI
- Calling Rust from C and vice versa

## INFORMATION

**Price:** 2000 EUR/person. excl. taxes

**Duration:** 3 days

**Practice:** 50%

*Public*

**Developers**



*Pre-requisites*

**Functional Programming**



**Rust Language**



*Pedagogical Objectives*

- Use traits and master ownership in advanced cases
- Choose the most suitable types for your applications
- Write a concurrent multithreaded program
- Write asynchronous code
- Master higher-order programming
- Call Rust from C and vice versa

*Langages*

**French**



**English**



*Result Indicators*

- Mean satisfaction score: 18.5/20
- Taux de réussite: 100%

## CONTACT

✉ contact@ocamlpro.com

☎ +33 6 72 73 37 53

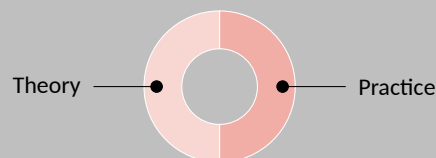
📍 21, rue de Chatillon, 75014, Paris, France

🏠 training.ocamlpro.com

🐦 @ocamlpro

Version: January 22, 2025

## THEORY VS PRACTICE





## **EVALUATING PROGRESS**

---

We make the progress of our trainees and its evaluation a core aspect of our courses. Indeed, guaranteeing the durable acquisition of the skills at hand is key, especially for the newer comers. To that extent, we will have trainees undergo tailored group works, exercises and hands-on practice which modalities can all be customised to your specific needs.

At the end of the course, you will have an opportunity for feedback to help us improve upon our methods. This is crucial as we believe there is always room for learning on both sides of the desk and no opinion other than yours matters more for us to do so.

## **CONSIDERING RQTH(RECOGNITION OF HANDICAPPED WORKER STATUS)**

---

If people with disabilities are part of the course, do reach out to us so we can adapt the training accordingly.

## **PEDAGOGICAL RESSOURCES**

---

The ressources are written by the OCamlPro team prior to the courses. Documents are generally written in english and can be translated to french if need be.

## **FUNDING RESORTS IN FRANCE: OPCO AND CPF**

---

The funding of the training by the OPCO is possible as OCamlPro has received the Qualiopi certification. The two following conditions must then be attained:

- The funding must cover all expenses relative to the course.
- The funding agreement must get to us five days prior to the training session at the latest.

## **INTER CORPORATION TRAININGS**

---

Les horaires pour nos formations inter-entreprises in-situ sont :  
Start - 9:30AM Lunch Break - 12:00PM to 01:00PM End - 05:30PM