AMBIENT-IT

+33 (0)1 48 10 09 46
formation@ambient-it.net
140 Avenue Jean Lolive 93500 Pantin - France

Updated on 05/12/2026

Sign up

Lean Training: Proof Assistant

3 days (21 hours)

Overview

Lean is a proof assistant and programming language that allows you to formalize mathematics and verify logical properties with high reliability. It is used to secure proofs, document invariants, and prototype formal models.

This training aims to make Lean operational for writing readable, reusable, and maintainable proofs. You will learn how to structure a development project, utilize the standard library, and transform objectives into robust proof scripts.

The approach is resolutely practical: guided workshops, step-by-step demos, and formalization exercises (propositional logic, quantifiers, simple algebraic structures). Deliverables include a versioned Lean project, a collection of reusable lemmas, and a checklist of best practices (tactics, simplification, rewriting, file organization).

Like all our training courses, this one will introduce you to **the latest stable version** of the technology and its new features.

Objectives

- Install and configure a reproducible Lean environment.
- Write proofs using tactics and declarative style.
- Handle equalities, rewrites, and simplifications effectively.
- Organize files, namespaces, and project dependencies.
- Debug typing and proof inference errors.

Target Audience

- Developers looking to introduce formal guarantees.

- R&D engineers and data scientists working with critical models.
- Students/researchers in mathematics or theoretical computer science.

Prerequisites

- Basic logic (implication, quantifiers, case-by-case reasoning).
- Basic functional programming concepts (types, functions, immutability).
- Comfort with the command line.
- Basic experience with Git.

Technical prerequisites

- At least 8 GB of RAM (16 GB recommended).
- Linux, macOS, or Windows with WSL2.
- Lean 4 and the associated build tool (via the Lean version manager).
- Code editor with Lean support (dedicated extension) and terminal.

Our Lean Training Program

[Day 1 - Morning]

Getting started with Lean and the proof workflow

- Installing Lean 4 and the environment (VS Code, Lean extension, Lake)
- Understanding the loop: objectives, tactics, error messages, search
- Working with the basics: Prop, Type, variables, implicits, notations
- Writing simple proofs: by, intro, exact, apply
- Hands-on workshop: Setting up a Lean project and proving elementary implications.

[Day 1 - Afternoon]

Propositional logic: connectors and essential tactics

- Conjunction/disjunction: constructor, cases, left/right
- Negation and contradiction: by_contra, contradiction, exfalso
- Equality: rfl, simp, rewriting with rw
- Structuring a proof: have, let, subproofs, and readability
- Hands-on workshop: Prove a series of logic lemmas (de Morgan’s, contrapositive) using simp and rw.

[Day 2 - Morning]

Quantifiers and proofs on basic types

- Quantifiers: ? (intro), ? (exists), case-by-case extraction
- Functions and lambda: proofs on functions, extensionality (funext)
- Reasoning on Nat: calculus, simple inequalities, normalization using simp
- Using the library: finding a lemma, navigating Mathlib, #check, #find
- Hands-on workshop: Formalizing statements with ?/? and proving simple properties on Nat.

[Day 2 - Afternoon]

Induction and structured proofs

- Induction on Nat: induction, base case, induction hypothesis
- Recursion and definitions: writing functions and proving their properties
- Chaining rewrites: calc, ring (if available), targeted simplifications
- Goal management: simp?, aesop (if enabled), debugging strategies
- Hands-on workshop: Proving classical properties by induction (sum, double, associativity on Nat).

[Day 3 - Morning]

Structures, inductive types, and case-by-case proofs

- Inductive types: definition, constructors, elimination by cases
- Lists: nil/cons, proofs by induction on List
- Structures and records: fields, projections, structural equality
- Reasonable automation: simp rules, attributes, auxiliary lemmas
- Hands-on workshop: Defining a function on lists and proving a property (length, append, map).

[Day 3 - Afternoon]

Mini-project: formalize a statement and produce a robust proof

- Choose a realistic statement (logic + induction) and break it down into lemmas
- Write a maintainable proof: names, sections, comments, factorization
- Stabilize with simp/rw: avoid loops and fragile rewrites
- Review and improvement: readability, performance, minimal dependencies
- Hands-on workshop: Complete a full mini-project (specification, lemmas, final proof) and present it.

Relevant companies

This training program is designed for both individuals and businesses—large and small—that wish to train their teams in new, advanced IT technologies or to acquire specific industry knowledge or modern methodologies.

Placement upon enrollment

The pre-training assessment complies with Qualiopi quality standards. Upon final registration, the learner receives a self-assessment questionnaire that allows us to evaluate their estimated proficiency in various types of technologies, as well as their expectations and personal goals for the upcoming training, within the limits imposed by the selected format. This questionnaire also allows us to anticipate certain connection or internal security issues within the company (intra-company or virtual classroom) that could pose challenges for monitoring and ensuring the smooth running of the training session.

Teaching Methods

Practical Course: 60% Practical, 40% Theory. Training materials distributed in digital format to all participants.

Organization

The course alternates between theoretical input from the trainer, supported by examples and reflection sessions, and group work.

Assessment

At the end of the session, a multiple-choice questionnaire is used to verify that the skills have been properly acquired.

Certification

A certificate will be issued to each trainee who has completed the entire training program.