

# Report from the FRO Lean Workshop @ ITP 2025

Joachim Breitner  
Principal Research Software Engineer

October 2nd, 2025, Reykjavik

**FRO what?**

# Lean FRO Mission

---

Our mission focuses on enhancing **Lean** in key areas: **scalability**, **usability**, **documentation**, and **proof automation**, while also broadening its application in various fields such as **education**, **research**, and **industry**. Over the next five years, we are dedicated to advancing Lean's capabilities and expanding its influence, ensuring it becomes an indispensable tool in these diverse domains. A pivotal aspect of our mission is to steer Lean towards **self-sustainability**, laying a strong foundation for its enduring growth and widespread utilization.

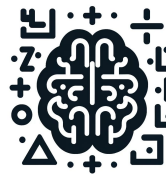
[lean-lang.org/fro/roadmap](https://lean-lang.org/fro/roadmap)



Formal mathematics



Software/Hardware  
verification



AI for math and  
code synthesis



Math and CS  
education

# Lean FRO Genesis

---

**Founded by**

Leonardo de Moura  
Sebastian Ullrich  
(August 2023)

**Funded by**

Simons Foundation International  
Alfred P. Sloan Foundation  
Richard Merkin Foundation  
Alex Gerko  
Convergent Research

# Lean FRO Team

---



# Lean FRO Team

---



Leo de Moura



Sebastian Ullrich



Ashley Blacquiere



Henrik Böving



Joachim Breitner



David Christiansen



Sebastian Graf



Markus Himmel



Marc Huisinga



Mac Malone



Joscha Menniken



Kyle Miller



Kim Morrison



Jason Reed



Paul Reichert



Sofia Rodrigues



Wojciech Różowski



Rob Simmons



Cameron Zwarich



Anne Baanen

Joseph Rotella

Johan Commelin

# Lean FRO Growth

---

## July 2025: New funding

Alex Gerko, CEO and founder of XTX Markets, donates \$5M to the Lean FRO.  
New team led by Jason Reed to focus on tooling and UX.

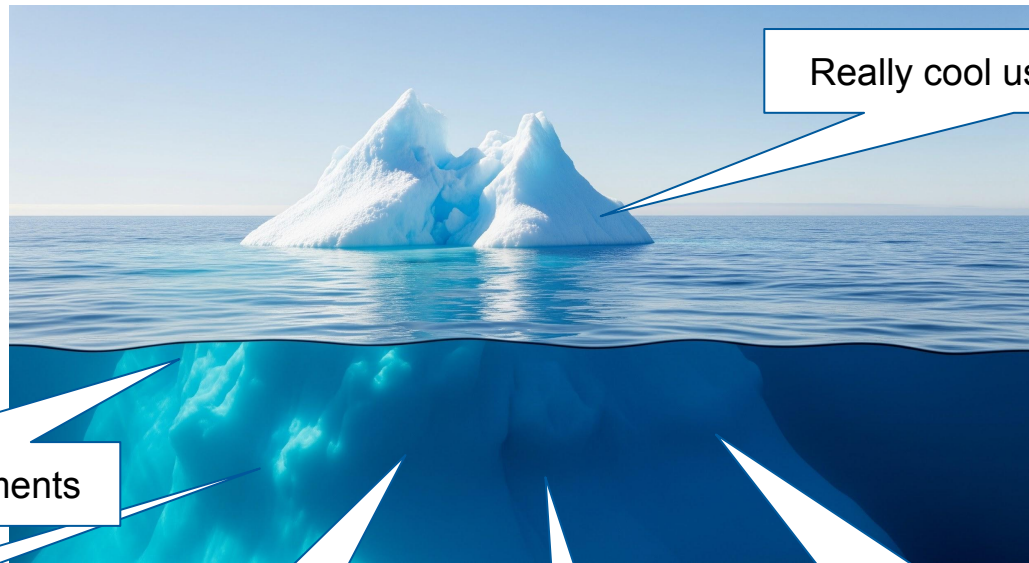
Alex Gerko, yes, the same one, donates \$5M to found the **Mathlib initiative**.  
Led by Johan Commelin, Adam Topaz and Olivier Nash.

# Lean highlights



# Lean highlights

---



Really cool user-visible feature

Performance improvements

New compiler

Release management

Bugfixes

That one missing `whnf`

# Demo time

---

<https://github.com/nomeata/lean-workshop-2025>

# New homepage

INTERNET ARCHIVE  
Wayback Machine  
102 captures  
18 Oct 2023 - 12 Jul 2025

https://lean-lang.org/about/

LEAN

ABOUT | DOWNLOAD | DOCUMENTATION | BLOG

LEAN

Install | Learn | Community | Use Cases | FRO | Playground | Reservoir


🌙 ↻

## About

Lean is a functional programming language that makes it easy to write can also use Lean as an interactive theorem prover. Lean programming and functions. This allows your focus to remain on the problem domain than the details of programming. Lean has numerous features, including:

- Type inference
- First-class functions
- Powerful data types
- Pattern matching
- Type classes
- Extensible syntax
- Hygienic macros
- Dependent types
- Metaprogramming framework
- Multithreading
- Verification: you can prove properties of your functions using Lean

The Lean project was launched by Leonardo de Moura when he was at open source project, hosted on GitHub. Lean 4 is the latest version. For Download. Information about the community of Lean users and the mathlib can be found at the Lean Community website. The Lean FRO r 2023. Its mission is to tackle the challenges of scalability, usability, and theorem prover.



Lean is an **open-source programming language** and **proof assistant** that enables correct, maintainable, and formally verified code

[→ Install](#)[Learn](#)

Powerful automation

Mathematics


```
-- 'Grind' efficiently manages complex pattern matching and
-- case analysis beyond standard tactics.

example (x : Nat) : 0 < match x with
| 0 => 1
| n+1 => x + n := by
  grind


-- Automatically solves systems of linear inequalities.

example (x y : Int) :
  27 ≤ 11*x + 13*y → 11*x + 13*y ≤ 45
  → -10 ≤ 7*x - 9*y → 7*x - 9*y > 4 := by
  grind
```


Grind is a powerful tool that can help you prove theorems quickly and efficiently.

 Trustworthy

Lean's minimal trusted kernel guarantees absolute correctness in mathematical proof, software and hardware verification.

 Powerful

From elementary concepts to cutting-edge research, Lean's expressive language and extensive built-in tools let users focus on the big picture rather than routine details.

 Extensible

Lean's metaprogramming capabilities enable users to extend the language with domain-specific notations and new proof automation techniques.

# Lean highlights of the past year

---

New in Lean 4.12...4.24 (highly subjective selection)

- New tactics: `bv_decide`, `grind`, `fun_induction`, `try?`, `mvcgen`
- New definitions: `partial fixpoint`, coinductive predicates, nested well-founded recursion
- Incrementality and parallelism
- Language Server & Code plugin: Inlay hints, Gutter decorations, unknown identifier code action, import hierarchy, Signature Help, structure auto-completion, overloaded go-to-def
- Usability: suggestions in error messages, better error messages, `tactic` configuration syntax,
- Library: map containers, iterators, systematic library development
- Module system
- Lake: local artifact cache
- Reference manual (and verso), named error messages
- New homepage

# Year 3 Roadmap

## Year 3 roadmap (selection)

---

- Module system for the masses
- Universal **rw** tactic
- Counter-example generation
- **do**-notation refactoring and integration with **mvcgen**
- Async/await and HTTP server functionality
- Cloud cache and version resolution in lake
- ...and performance and usability improvements throughout

[Full roadmap online](#)

**Questions?**