

LENN

THEOREM PROVER

# What 's LEAN?

Lean is a functional programming language that makes it easy to write correct and maintainable code.

You can also use Lean as an interactive theorem prover. Lean programming primarily involves defining types and functions.

This allows your focus to remain on the problem domain and manipulating its data, rather than the details of programming.

# Features

Lean has numerous features, including:

- 1.Type inference
- 2.First-class functions
- 3.Powerful data types
- 4.Pattern matching
- 5.Type classes
- 6.Extensible syntax
- 7.Hygienic macros
- 8.Dependent types
- 9.Metaprogramming framework
- 10.Multithreading
- 11.Verification

(you can prove properties of your functions using Lean itself)

# Installation

OPTIONAL: RECOMMENDED TO INSTALL DEVIA 2 EXTENDED CONFIGURATION ASSETION:

THE 187625 ASSETION IS DEVIA 4. HOWEVER, IT DOESN'T SUPPORT LIBRARY \_\_\_\_\_ ONE OF THE MOST  
IMPORTANT FEATURES TO DEVIA:

WE CAN EDIT SUPPORT, THIS IS THE FIRST AND MOST IMPORTANT CODE EXTENSION, EACH OF WHICH  
INCLUDES EXTENSIVE SUPPORT:

DEVIA DOES PROVIDE WITH THE TOOL, IDENTIFIERS, WHICH KEYS ADD CODE UNDER ASSETION  
CONTROL WITH OUR

# VsCode VS Emacs

I like VsCode better than Emacs, because of the following reasons:

1.VsCode is the official supported editor of LEAN.

2.VsCode has a better performance in auto-complete and program analysis when we use LEAN.

3.VsCode is more out-of-box when Emacs spending a lot of time to build your own configuration(Maybe it's not an advantage for some groups of people, but it's newbie friendly).

**However, Emacs is much more lightweight.**

# Talk is cheap, show me the code!

Begin with LEAN studying using VsCode!