Updated on 27/06/2025

Sign up

Neovim training

2 days (14 hours)

Presentation

Master Neovim and its modern features with our immersive two-day training course, designed for developers wishing to take advantage of a lightweight, powerful and fully customizable editor. From getting started to advanced plugin integration.

You'll start by discovering Neovim's unique architecture, how it differs from Vim, and how to navigate efficiently between modes, buffers, windows and files. This is an essential first step in exploiting the productivity offered by this essential tool.

You'll then configure Neovim in Lua, the now-recommended native language, to build a modular, clear and maintainable configuration. You'll learn how to manage your plugins with modern managers such as Lazy.nvim or Packer, while discovering key everyday tools such as the file explorer and status bar.

dive into advanced features: intelligent autocompletion with nvim-cmp, native support for Language Servers (LSP), snippet integration and code navigation with Treesitter. You'll be able to seamlessly link analysis, editing and refactoring.

Like all our training courses, this one is based on the latest stable version of Neovim.

Objectives

- Understand Neovim's modern architecture and its advantages over Vim, including native integration of Lua and the LSP protocol.
- Know how to configure, customize and maintain a modular, high-performance Neovim environment development needs
- master the main plugins in the Neovim ecosystem and automate their management with tools such as Lazy.nvim or Packer
- Be able to work efficiently with language servers
- Adopt best practices for organizing, versioning and sharing your Neovim configuration in a robust and productive development environment.

Target audience

- Developers
- DevOps

Prerequisites

- good command of the terminal

Neovim training program

Introduction to Neovim

- Differences between Vim and Neovim
- Advantages for modern developers
- Modern architecture
- Installation on Windows / Linux / macOS
- Structure of config files : ~/.config/nvim/
- Difference between init.vim and init.lua

Neovim editing basics

- Normal, insert, visual and command line modes
- Switching between modes
- File navigation
- Word, line, block and file jumps
- Saving, closing, undoing/redoing
- Search and replace
- Efficient copying, pasting and deleting

Lua configuration

- Why Lua in Neovim
- Basic Lua syntax
- Structure of a modular Lua config
- Creating a clean init.lua file
- Basic options: numbering, colors, indentation
- Shortcut mapping
- Organization in separate files

Modern plugins with Packer or Lazy.nvim

- packer.nvim vs lazy.nvim
- Manager installation and configuration
- File explorer: nvim-tree.lua
- Status bar: lualine.nvim
- Quick search: telescope.nvim
- Use or lazy.setup
- Conditional loading, lazy loading

Using the Language Server Protocol (LSP)

- Role of Neovim's integrated LSP client
- nvim-lspconfig and mason.nvim
- Useful shortcuts: go to definition, hover, rename...
- Diagnostics and action code management
- Auto-formatting on save

Intelligent autocompletion with nvim-cmp

- Modular architecture
- Autocompletion sources
- Mapping of completion keys
- Customized completion window
- Integrating luasnip for code snippets
- Using LSP server snippets

Enhancing the Treesitter experience

- Modern syntax parsing
- Differences from traditional coloring
- Installing nvim-treesitter
- Choice of supported languages
- Intelligent syntax selections
- Textobjects, folding, rainbow brackets...

User interface and comfort

- Themes: tokyonight, gruvbox, etc.
- Configure lualine, bufferline, indentation guides...
- which-key.nvim: contextual shortcut menu
- telescope.nvim for fast navigation
- Using the integrated terminal
- Horizontal/vertical split, tabs, buffers

Best practices and organization

- Modular approach with Lua
- Files separated by responsibility
- Useful auto-commands
- Plugins for project management
- Using Git to version your config
- Managing dotfiles with chezmoi, yadm

Companies concerned

This course is aimed at both individuals and companies, large or small, wishing to train their teams in a new advanced IT technology, or to acquire specific business knowledge or modern methods.

Positioning on entry to training

Positioning at the start of training complies with Qualiopi quality criteria. As soon as registration is finalized, the learner receives a self-assessment questionnaire which enables us to assess his or her estimated level of proficiency in different types of technology, as well as his or her expectations and personal objectives for the training to come, within the limits imposed by the selected format. This questionnaire also enables us to anticipate any connection or security difficulties within the company (intra-company or virtual classroom) which could be problematic for the follow-up and smooth running of the training session.

Teaching methods

Practical training: 60% hands-on, 40% theory. Training material distributed in digital format to all participants.

Organization

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

Validation

At the end of the session, a multiple-choice questionnaire verifies the correct acquisition of skills.

Certification

A certificate will be awarded to each trainee who has completed the entire course.