

Mis à jour le 30/06/2025

S'inscrire

## Formation Neovim

2 jours (14 heures)

### Présentation

Maîtrisez Neovim et ses fonctionnalités modernes grâce à notre formation immersive de deux jours, conçue pour les développeurs souhaitant tirer parti d'un éditeur léger, puissant et entièrement personnalisable. De la prise en main à l'intégration avancée de plugins.

Vous commencerez par découvrir l'architecture unique de Neovim, ses différences avec Vim, et vous apprendrez à naviguer efficacement entre les modes, buffers, fenêtres et fichiers. Une première étape essentielle pour exploiter la productivité offerte par cet outil incontournable.

Vous configurerez ensuite Neovim en Lua, le langage natif désormais recommandé, pour construire une configuration modulaire, claire et maintenable. Vous apprendrez à gérer vos plugins avec des gestionnaires modernes comme Lazy.nvim ou Packer, tout en découvrant les outils clés du quotidien : explorateur de fichiers, barre de statut.

vous plongerez dans les fonctionnalités avancées : autocomplétion intelligente, support natif des Language Servers (LSP), intégration de snippets et navigation dans le code avec Treesitter. Vous serez capable d'enchaîner analyse, édition et refactorisation avec fluidité.

Comme toutes nos formations, celle-ci repose sur la dernière version stable de [Neovim](#).

### Objectifs

- Comprendre l'architecture moderne de Neovim et ses apports par rapport à Vim, notamment l'intégration native de Lua et du protocole LSP
- Savoir configurer, personnaliser et maintenir un environnement Neovim modulaire, performant et adapté à ses besoins de développement
- Maîtriser les principaux plugins de l'écosystème Neovim et automatiser leur gestion avec des outils comme Lazy.nvim ou Packer
- Être capable de travailler efficacement avec des serveurs de langage
- Adopter les bonnes pratiques pour organiser, versionner et partager sa configuration Neovim dans un environnement de développement robuste et productif

## Public visé

- Développeurs
- DevOps

## Pré-requis

- bonne maîtrise de l'utilisation du terminal

## Programme de la formation Neovim

### Introduction à Neovim

- Différences entre Vim et Neovim
- Avantages pour les développeurs modernes
- Architecture moderne
- Installation sur Windows / Linux / macOS
- Structure des fichiers de config : ~/.config/nvim/
- Différence entre init.vim et init.lua

### Les bases de l'édition dans Neovim

- Mode normal, insertion, visuel, ligne de commande
- Passages entre les modes
- Navigation dans un fichier
- Sauts de mots, lignes, blocs, fichiers
- Enregistrement, fermeture, annulation/rétablissement
- Recherche et remplacement
- Copier, coller, supprimer efficacement

### Configuration en Lua

- Pourquoi Lua dans Neovim
- Syntaxe de base de Lua
- Structure d'une config Lua modulaire
- Créer un fichier init.lua propre
- Options de base : numérotation, couleurs, indentation
- Mapping de raccourcis
- Organisation en fichiers séparés

### Plugins modernes avec Packer ou Lazy.nvim

- packer.nvim vs lazy.nvim
- Installation et configuration du gestionnaire
- Explorateur de fichiers : nvim-tree.lua
- Barre de status : lualine.nvim
- Recherche rapide : telescope.nvim
- Utiliser use ou lazy.setup
- Chargement conditionnel, lazy loading

## Utiliser le Language Server Protocol (LSP)

- Rôle du client LSP intégré de Neovim
- nvim-lspconfig et mason.nvim
- Raccourcis utiles : go to definition, hover, rename...
- Gestion des diagnostics et code actions
- Auto-formatage à la sauvegarde

## Autocomplétion intelligente avec nvim-cmp

- Architecture modulaire
- Sources d'autocomplétion
- Mapping des touches de complétion
- Fenêtre de complétion personnalisée
- Intégrer luasnip pour les extraits de code
- Utiliser les snippets du serveur LSP

## Améliorer l'expérience avec Treesitter

- Parsing syntaxique moderne
- Différences avec la coloration traditionnelle
- Installer nvim-treesitter
- Choix des langages supportés
- Sélections syntaxiques intelligentes
- Textobjects, folding, rainbow brackets...

## Interface utilisateur et confort

- Thèmes : tokyonight, gruvbox, etc.
- Configurer lualine, bufferline, indentation guides...
- which-key.nvim : menu contextuel des raccourcis
- telescope.nvim pour la navigation rapide
- Utiliser le terminal intégré
- Split horizontal/vertical, tabs, buffers

## Bonnes pratiques et organisation

- Approche modulaire avec Lua
- Fichiers séparés par responsabilité
- Auto-commands utiles
- Plug-ins pour la gestion de projet
- Utiliser Git pour versionner sa config
- Gérer les dotfiles avec chezmoi, yadm

## Sociétés concernées

Cette formation s'adresse à la fois aux particuliers ainsi qu'aux entreprises, petites ou grandes, souhaitant former ses équipes à une nouvelle technologie informatique avancée ou bien à acquérir des connaissances métiers spécifiques ou des méthodes modernes.

## Positionnement à l'entrée en formation

Le positionnement à l'entrée en formation respecte les critères qualité Qualiopi. Dès son inscription définitive, l'apprenant reçoit un questionnaire d'auto-évaluation nous permettant d'apprécier son niveau estimé sur différents types de technologies, ses attentes et objectifs personnels quant à la formation à venir, dans les limites imposées par le format sélectionné. Ce questionnaire nous permet également d'anticiper certaines difficultés de connexion ou de sécurité interne en entreprise (intraentreprise ou classe virtuelle) qui pourraient être problématiques pour le suivi et le bon déroulement de la session de formation.

## Méthodes pédagogiques

Stage Pratique : 60% Pratique, 40% Théorie. Support de la formation distribué au format numérique à tous les participants.

## Organisation

Le cours alterne les apports théoriques du formateur soutenus par des exemples et des séances de réflexions, et de travail en groupe.

## Validation

À la fin de la session, un questionnaire à choix multiples permet de vérifier l'acquisition correcte des compétences.

## Sanction

Une attestation sera remise à chaque stagiaire qui aura suivi la totalité de la formation.