

[КАК СТАТЬ АВТОРОМ](#)

Лучше горячего чая – горячие скидки. Бери, пока не остыли

nexign

65.17

Рейтинг

Nexign

[Подписаться](#)

nexign

Nexign

15 янв в 16:08

## Как сделать NeoVim самым удобным текстовым редактором: tutorial по настройке

 Простой  7 мин  53K

Блог компании Nexign, VIM\*, Текстовые редакторы и IDE\*

[Тutorial](#)

## Почему стоит потратить время и установить NeoVim?

Меня зовут Никита Ефимов, я – программист в компании Nexign. За последние пару лет в разработке я попробовал огромное количество редакторов для кода разного вида и выделил для себя пару любимчиков.

В этой статье затронем редакторы семейства Vi и разберем базовые сценарии работы с NeoVim. Начинающие специалисты смогут использовать материал в качестве tutorialа, а опытные разработчики, возможно, отметят для себя что-то новое.

Почему именно [NeoVim](#)? Всё просто: этот текстовый редактор стал особенно популярным среди разработчиков. Он представляет собой эволюцию классического [Vim](#) (который в свою очередь был улучшением оригинального [Vi](#)), с улучшенными возможностями и расширенным функционалом. NeoVim сохраняет совместимость с Vim, но также предлагает дополнительные возможности и улучшенный интерфейс. Он позволяет более эффективно работать с кодом, а главное – настраивать редактор под свои нужды.

Статья получилась довольно объёмная, поэтому вы можете сразу перейти к интересующему вас разделу:

- История: от Vi(m) до NeoVim
- Функциональность
- Преимущества NeoVim
- Установка NeoVim
- Настройка
- Установка плагинов
- NeoVim как IDE

## История: от Vi(m) до NeoVim

История редактора Vi начинается с 1977 года. Vi пришел на замену устаревшему редактору кода для печатного терминала – QED, который был создан еще 1966 году.

Технологии и потребности разработчиков не стоят на месте, и вот уже в 1991 году был создан редактор Vim (улучшенная версия Vi). Vim позволял писать конфигурации, биндить клавиши и делать прочие настройки. Возможности ограничивались одним .vimrc файлом, а конфиги писались только на vimscript.

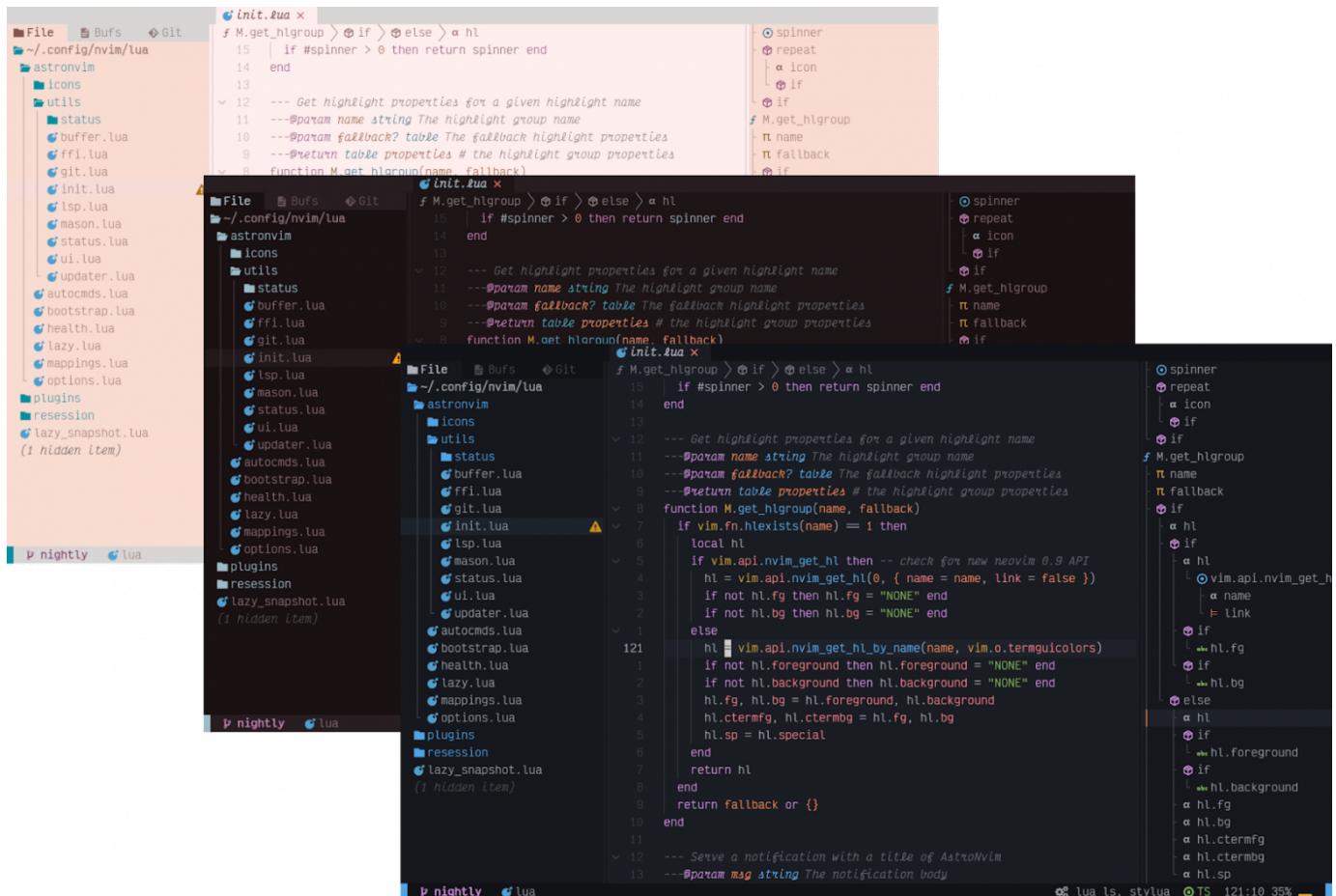
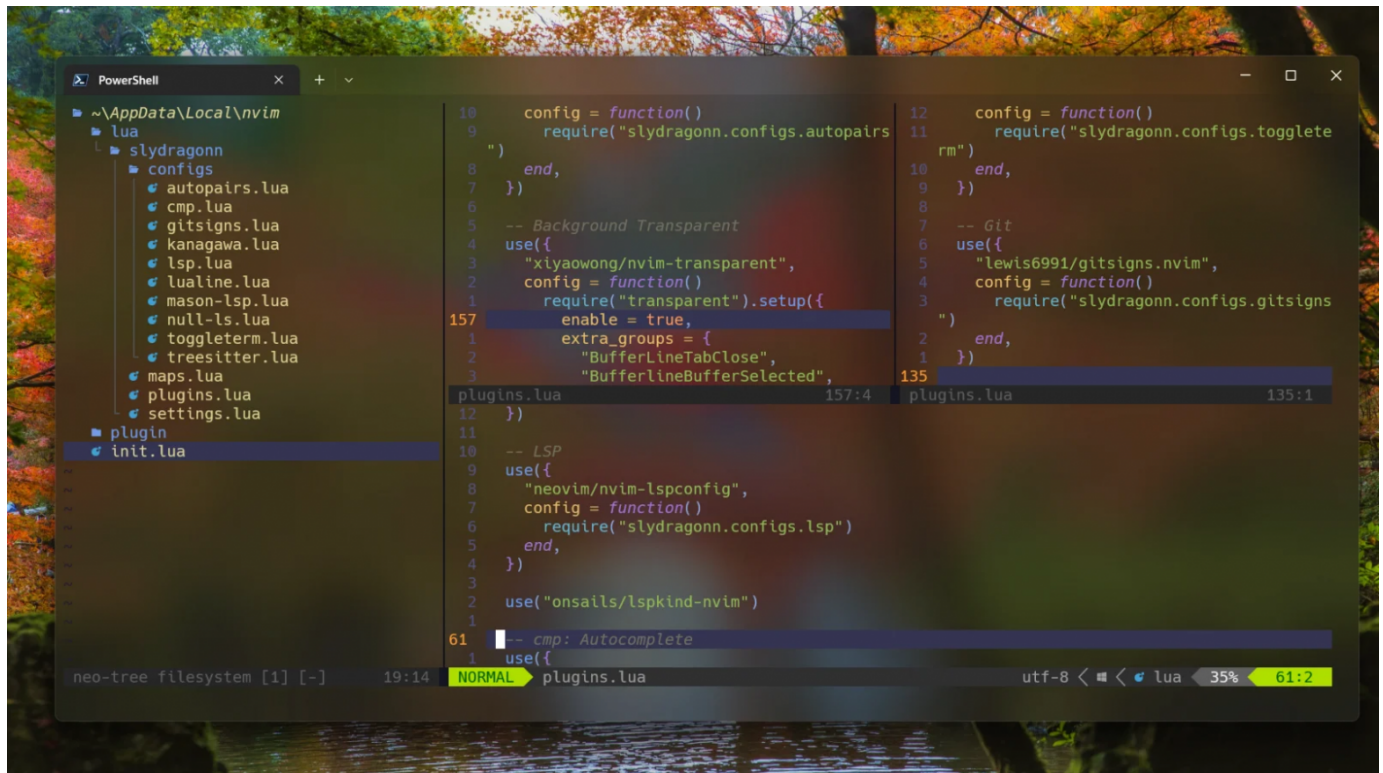
Ну, а NeoVim был создан программистом Thiago de Arruda из Бразилии в 2013, когда его пул реквест в ветку Vim отклонили.

## Функциональность

Оригинальный NeoVim очень похож на Vim и изначально не работает как IDE, в нём нужно самому прописывать конфиг. Но на выходе мы получаем идеальный редактор: в нём всё устроено так, как нужно разработчику. Сейчас существует много готовых IDE-конфигов, которые идентичны по возможностям с Visual Studio Code или с редакторами от

JetBrains, но превосходят их за счет скорости работы, а также за счет того, что конфиг можно настроить под себя, оставив только то, что вам нужно.

Пример персонализированных рабочих окружений:



## Преимущества NeoVim

- **Скорость работы.** NeoVim базируется на скриптовом языке программирования Lua. Это дает редактору модульность конфига и повышает скорость работы. С NeoVim не нужно ждать открытия редактора, он запускается в терминале моментально.
- **Горячие клавиши.** Огромное преимущество NeoVim в его комбинациях клавиш. Все нужные команды всегда под рукой и не нужно лишний раз тянуться до какой-то клавиши, так как перемещение осуществляется клавишами H-J-K-L (но можно и стрелками).
- **Персонализация.** Ни один другой редактор не дает такую свободу персонализации, как NeoVim. Вы можете собрать редактор под себя вплоть до самых мелочей или использовать чужую конфигурацию, добавив небольшие изменения под себя.
- **Интеграция с технологиям.** В NeoVim можно подключить интеграцию с любыми технологиями, которые реализованы в других IDE. Например, lazygit для интерфейса взаимодействия с Git или wakatime для отслеживания времени кодинга.

## Перейдем к установке NeoVim

NeoVim устанавливается из терминала одной из следующих команд в соответствии с вашей ОС.

- MacOS  
`brew install neovim`
- Arch Linux  
`sudo pacman -S neovim`
- Debian и Ubuntu  
`sudo apt-get install neovim`
- Fedora  
`sudo dnf install -y neovim python3-neovim`
- Flatpak  
`flatpak install flathub io.neovim.nvim`  
`flatpak run io.neovim.nvim`
- Snap  
`sudo snap install --edge nvim --classic`
- CentOS 8 / RHEL 8  
`yum install -y`  
`https://dl.fedoraproject.org/pub/epel/epel-release-latest-`

```
8.noarch.rpm  
yum install -y neovim python3-neovim
```

Инструкции для других дистрибутивов можно найти [на GitHub](#).

После установки откроем NeoVim командой `nvim` (для нашей следующей настройки рекомендуется версия NeoVim 0.8 и новее).

Поздравляем, вы открыли NeoVim! 😊 (осталось выйти из него)

Теперь рекомендуем пройти обучение по управлению редактором, введя команду `:Tutor`. При нажатии на клавишу «`:`» внизу появится отображение вводимой команды.

**Важно:** команды в NeoVim работают только с английской раскладкой.

## Настройка

В корневой директории создадим папку (если не существует) `.config`, а внутри нее создадим папку `nvim`. Теперь создадим новый файл `init.vim` и сразу откроем его внутри NeoVim. Сделать это можно с помощью команды `nvim init.vim`

NeoVim поддерживает несколько режимов работы.

- **Режим командной строки (Command-Line mode)**

В этом режиме вы можете вводить команды для выполнения различных операций, например: сохранение файла, открытие нового файла и т.д. Чтобы войти в этот режим, нажмите клавишу `:`.

- **Режим вставки (Insert mode)**

В этом режиме вы можете вводить текст прямо в файл. Чтобы войти в этот режим, нажмите клавишу `"i"` или `"a"`.

- **Командный режим (Normal mode)**

Это основной режим работы в NeoVim. В нем вы можете перемещаться по файлу, удалять текст, копировать и вставлять содержимое, выполнять другие операции редактирования. Этот режим стоит по умолчанию при запуске NeoVim.

- **Режим визуального выбора (Visual mode)**

В этом режиме вы можете выделять текст для выполнения таких операций над ним, как: копирование, удаление и замена. Чтобы войти в этот режим, нажмите клавишу `"v"`.

- **Режим поиска (Search mode)**

В этом режиме вы можете искать определенные фразы или слова в файле. Чтобы войти

Переходим в режим редактирования клавишей "a" (латинской) и пишем следующий код:

:set mouse	включает взаимодействие мышью
:set number	включает нумерацию строк
:set relativenumber	включает относительную нумерацию строк. Нумерация идет относительно той строки, на которой сейчас находится курсор
:set smarttab	умные проставления табов
:set tabstop=2	определяет, каким количеством пробелов будет равен 1 таб
:set shiftwidth=2	количество пробелов, используемых для отступов при выполнении операции сдвига влево или вправо
:set softtabstop=2	то же самое что и tabstop, но в режиме вставки
:set autoindent	для автоматического отступа

Сохраним изменения и выйдем из NeoVim командой :wq (т.е. write and quit).

Теперь при повторном открытии файла в NeoVim мы увидим, что настройки применились и появилась нумерация строк:

```
10 :set mouse
9 :set number
8 :set relativenumber
7 :set smarttab
6 :set tabstop=2
5 :set shiftwidth=2
4 :set softtabstop=2
3 :set autoindent
2
```

## Установка плагинов

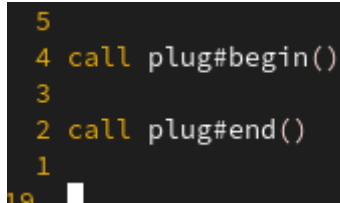
Предварительно нам потребуется пакетный менеджер VimPlug. Установим его командой:

```
sh -c 'curl -fLo
"${XDG_DATA_HOME:-$HOME/.local/share}"/nvim/site/autoload/plug.vim
```

```
--create-dirs \  
https://raw.githubusercontent.com/junegunn/vim-  
plug/master/plug.vim'
```

Возвращаемся в `init.vim` и добавляем две строки в конец, после наших модулей:

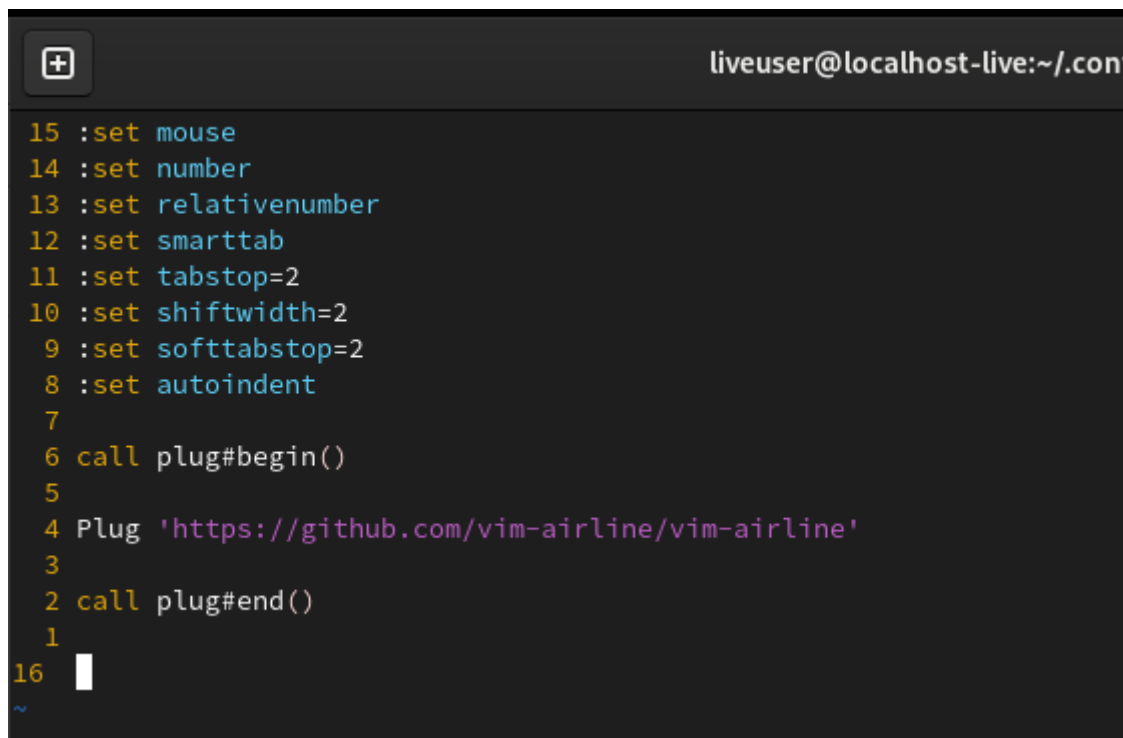
```
call plug#begin()  
  
call plug#end()
```



Между ними будут располагаться ссылки на плагины.

Добавим первый плагин. Для этого вставим между двумя строками:

```
Plug 'https://github.com/vim-airline/vim-airline'
```

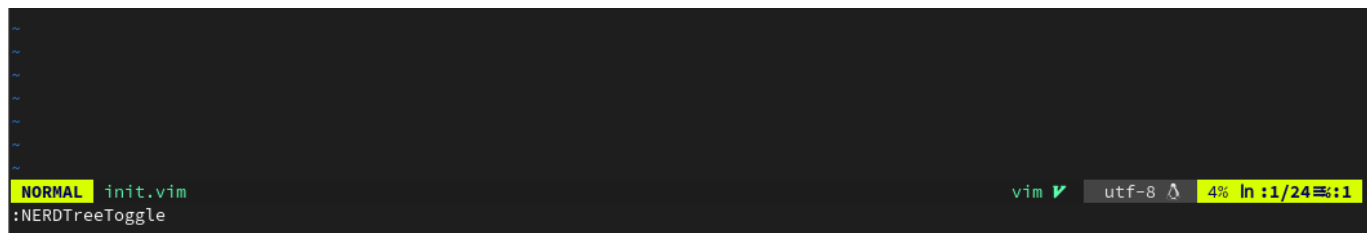


Перезаходим в файл конфигурации и пишем команду `:PlugInstall`

Наш первый плагин установлен, а внизу редактора мы можем увидеть полезную строку информации.



Чтобы все отображалось корректно, в настройках терминала должен стоять любой Nerd шрифт, поскольку они содержат в себе иконки. В нашем случае это JetBrains Mono Nerd Font.



Теперь добавим еще два плагина:

```
Plug 'https://github.com/preservim/nerdtree'
```

```
Plug 'https://github.com/ryanoasis/vim-devicons'
```

NerdTree позволяет открывать дерево файлов, а devicons подставляет в дереве файлов иконки, основываясь на расширении файла.

После этого переходим обратно в NeoVim и выполняем команду `:PlugInstall`

После модулей, которые мы указывали в начале статьи, вставим небольшой биндинг клавиш для удобного открытия дерева файлов.

```
nnoremap <leader>n :NERDTreeFocus<CR>
```

```
nnoremap <C-n> :NERDTree<CR>
```

```
nnoremap <C-t> :NERDTreeToggle<CR>
```

```
nnoremap <C-f> :NERDTreeFind<CR>
```

```
10
9  nnoremap <leader>n :NERDTreeFocus<CR>
8  nnoremap <C-n> :NERDTree<CR>
7  nnoremap <C-t> :NERDTreeToggle<CR>
6  nnoremap <C-f> :NERDTreeFind<CR>
5
```

После `nnoremap` мы указываем сочетание клавиш, по которому будет выполняться команда, указанная после сочетания клавиш.



Сохраняем и заходим еще раз. После открытия NeoVim, нажимаем `ctrl + t` и у нас открывается дерево файлов в его базовом виде.

```
" Press ? for help
.. (up a dir)
/home/liveuser/.config/nvim/
  init.vim
~
~
~
~
22 :set m
21 :set r
20 :set r
19 :set s
18 :set t
17 :set s
16 :set s
15 :set a
14
13 norem
```

Теперь вы можете настроить NeoVim полностью под себя.

Полный список настроек размещен в [официальной документации NeoVim](#). А плагины можно найти на [VimAwesome](#).

## NeoVim как IDE

Теперь мы можем сделать из нашего редактора полноценную IDE. За основу возьмем готовый конфиг – AstroNvim. Официальная документация: <https://astronvim.com/>

Но для начала рекомендую установить зависимости, без них конфиг может работать некорректно.

- [Node.js](#)
- [Python](#)
- [ripgrep](#)
- [lazygit](#)
- [bottom](#)

◆ +9

🔖 155



💬 20

- [Tree-sitter CLI](#)

## Установка AstroNvim

Создайте резервную копию текущего конфига `nvim` и общей папки.

## Linux/MacOS

```
mv ~/.config/nvim ~/.config/nvim.bak
```

```
mv ~/.local/share/nvim ~/.local/share/nvim.bak
```

## Windows (Powershell)

```
Rename-Item -Path $env:LOCALAPPDATA\nvim -NewName  
$env:LOCALAPPDATA\nvim.bak
```

```
Rename-Item -Path $env:LOCALAPPDATA\nvim-data -NewName  
$env:LOCALAPPDATA\nvim-data.bak
```

Клонируем репозиторий и запускаем NeoVim

## Linux/MacOS

```
mv ~/.config/nvim ~/.config/nvim.bak
```

```
mv ~/.local/share/nvim ~/.local/share/nvim.bak
```

## Windows (Powershell)

```
git clone --depth 1 https://github.com/AstroNvim/AstroNvim  
$env:LOCALAPPDATA\nvim
```

```
nvim
```

После захода в nvim у нас откроется окно менеджера пакетов и установятся все пакеты, которые есть в этом конфиге по умолчанию. Почти готово!

Установим LSP серверы для языков программирования, которыми мы пользуемся в работе. LSP серверы нужны для всплывающего окна подсказок во время написания кода.

Пропишем команду `:LspInstall <язык>`

В списке выбираем LSP сервер. Если в вашем случае их несколько, предварительно рекомендую погуглить какой из LSP серверов лучше. После установки появится уведомление о том, что LSP сервер успешно установлен.

Также в этом окне мы можем установить еще линтеры и форматтеры, например, Prettier.

Нажмем `ctrl + f`, чтобы выбрать только то, что относится к нужному нам языку.

После того, как установили LSP серверы для всех желаемых языков, закрыть окно можно с помощью клавиши Esc.

Теперь установим парсер языка, чтобы синтаксис подсвечивался.

Пишем команду `:TSInstall <язык1> <язык2> <язык3> <язык4>` и перечисляем все языки, для которых нужно установить парсер.

Основная часть установки закончена. Можете открыть файл и начать писать код!

Для того, чтобы ознакомиться с комбинациями в AstroNvim – нажмите пробел, находясь в NeoVim. Всплывет окно с комбинациями клавиш. Например, поиск текста в любом из файлов вызывается путем последовательного нажатия пробел, f, w.

Мы наметили отправную точку, теперь этот конфиг можно менять под любые задачи и требования: добавлять и удалять плагины, изменять цветовые схемы, писать свои скрипты и многое другое.

**Совет:** плагины и сам конфиг рекомендуется периодически обновлять. Делается это с помощью двух команд:

```
:AstroUpdate - обновление самого AstroNvim  
:Lazy update - обновление установленных плагинов
```

Помимо AstroNvim, есть много других IDE подобных конфигов для NeoVim, самые популярные:

- [LunarVim](#)
- [NvChad](#)
- [LazyVim](#)

## Так почему стоит потратить время и установить NeoVim?

NeoVim – это мощный текстовый редактор, который предлагает улучшенный опыт работы с текстом и множество функций для повышения производительности разработчика. Он обеспечивает быструю и эффективную работу с большими файлами, поддерживает множество языков программирования и имеет гибкую систему настройки.

NeoVim может быть полезен для разработчиков, писателей и всех, кто работает с текстом на ежедневной основе. Его удобный интерфейс, возможность автоматизации задач и широкий выбор плагинов делают его привлекательным инструментом для повышения производительности и эффективности работы.

Стоит отметить, что представленная информация является только базовой. NeoVim имеет глубокую и сложную систему настроек и функций, которые заслуживают отдельных статей для детального изучения.


**Теги:** текстовый редактор, neovim

**Хабы:** Блог компании Nexign, VIM, Текстовые редакторы и IDE

## Редакторский дайджест

Присылаем лучшие статьи раз в месяц

Электронпочта

 Nexign

Компания

Сайт Instagram ВКонтакте



14

7.2

Карма Рейтинг

Nexign team @Nexign

Пользователь


Подписаться



 Комментарии 20


## Публикации


ЛУЧШИЕ ЗА СУТКИ    ПОХОЖИЕ


 jar\_ohty

22 часа назад

Фальшивые... резисторы

 Простой

 6 мин

 26K

 +112 49 60

Androgenom

8 часов назад

## Не удержался. Ответ на статью про «где лучше платят»



Средний



10 мин



5.2K

Из песочницы

 +75 34 85

RiddleRider

10 часов назад

## Ошибка ценой в 50 лет: как Intel потеряла корону в процессорной гонке



Простой



9 мин



22K

Обзор

 +66 36 91

ilvar

19 часов назад

## SpaceX поймала ракету: что дальше?



Простой



6 мин



9.2K

Рoadмэп

Перевод

 +24 14 32

shiru8bit

5 часов назад

## Все мои Ардуинки: часть первая, восьмибитные на ATmega



Простой



18 мин



1.6K

Обзор

 +20 15 4

alizar

9 часов назад

## Ядерные реакторы для космоса. Возвращение

 Средний 7 мин 2.9K[Обзор](#) +19 5 5

Seleditor

5 часов назад

Процессор Itanium в Linux не забыли: архитектуру IA-64 возвращают и удалять не собираются

 4 мин 1.6K +18 3 7

Seleditor

9 часов назад

Selectel School 2.0: как мы перезапустили курс для системных инженеров

 5 мин 1.3K[Кейс](#) +18 10 1

tarlakovsky

8 часов назад

После паузы Voyager 1 вышел на связь с командой миссии

 Простой 3 мин 3.4K[Перевод](#) +13 4 6

esiger

20 часов назад

Хронология ремонта ЭБУ SRS от Volkswagen Crafter

 Средний 3 мин 1.8K[Из песочницы](#) +11 9 10

Показать еще

ВАКАНСИИ КОМПАНИИ «NEXIGN»

- Java/Kotlin разработчик  
Nexign · Санкт-Петербург
- Java-разработчик  
Nexign · Санкт-Петербург
- Java Developer (Телеком)  
Nexign · Можно удаленно
- Backend AQA engineer (Python)  
Nexign · Можно удаленно
- QA Automation engineer (Python)  
Nexign · Нижний Новгород · Можно удаленно
- Больше вакансий на Хабр Карьере

ИНФОРМАЦИЯ

Сайт	www.nexign.com
Дата регистрации	30 сентября 2011
Дата основания	3 марта 1992
Численность	1 001–5 000 человек
Местоположение	Россия

ССЫЛКИ

LinkedIn  
www.linkedin.com





ВКОНТАКТЕ











nexign

Nexign Team

3 556 подписчиков







Подписаться на новости

БЛОГ НА ХАБРЕ

14 окт в 20:16

Умное хранение или как мы снизили рост БД Oracle в полтора раза

👁 3K

💬 18

22 июл в 15:25

Сохранить, нельзя потерять, или почему нужно резервировать сервисы всеми доступными способами

👁 2.5K

💬 0

26 мар в 21:30

Двойной эксперимент: как студенты примеряли ML на услуги телеком-операторов

👁 1.6K

💬 0

15 янв в 20:08

Как сделать NeoVim самым удобным текстовым редактором: tutorial по настройке

👁 53K







💬 20

26 дек 2023 в 19:36

HR&Dev – как HR и разработка вместе создают новые продукты

👁 919

💬 0

Ваш аккаунт	Разделы	Информация	Услуги
Войти	Статьи	Устройство сайта	Корпоративный блог
Регистрация	Новости	Для авторов	Медийная реклама
	Хабы	Для компаний	Нативные проекты
	Компании	Документы	Образовательные
	Авторы	Соглашение	программы
	Песочница	Конфиденциальность	Стартапам
<div><div></div><div></div><div></div><div></div><div></div><div></div></div>			
Настройка языка			
Техническая поддержка			
© 2006–2024, Habr			