

# with-gemini

give me principles of "suckless neovim propaganda"

elaborate further in two parts, start with the first

then the rest of them

---

Ah, you seek the **Suckless Neovim Propaganda** — not merely a configuration, but a *philosophy*. A creed. A quiet rebellion against bloat, complexity, and the cult of "more features."

Here are the sacred principles — written not in plugins, but in spirit:

---

## The Suckless Neovim Propaganda: 7 Commandments

### 1. Do Not Install Plugins You Cannot Explain in One Sentence

"If you can't describe what your plugin does without using the word 'magic,' you don't need it."

*Every plugin is a dependency. Every dependency is a potential break. Every break is a moment stolen from your focus.*

*Rule of thumb: If it doesn't fundamentally change how you think about editing, delete it.*

### 2. Your Config Is Not a Portfolio — It's a Tool

No screenshots of 200-line `init.lua` with 17 color schemes and animated status lines.

Your config should be:

- **Readable** by a stranger on a Friday night.
- **Small enough** to type from memory after a coffee spill.
- **Functional** — not fashionable.

*The best config is the one you didn't write — it's the one you inherited from your fingers.*

### 3. Use Native Features. Always.

Neovim ships with a *powerful* editor.

- `:grep + :copen > fzf`
  - `:terminal > tmux pane hopping`
  - `:vimscript > Lua abstractions you don't understand`
- If a native feature does 90% of the job — do 90% of the job.  
The other 10%? You'll learn it. And you'll own it.

### 4. No Mouse. Ever.

The mouse is the first step toward becoming a GUI zombie.

Your hands belong on the home row.

Your eyes belong on the text.

Your brain belongs in the flow.

*If you reach for the trackpad, you've already lost.*

### 5. No Themes. No Icons. No Animations.

Color schemes are vanity.

Icons are distractions.

Animations are performance tax paid in focus.

Use `:set background=dark` or `:set background=light`.

That's it.

The *content* is the art. Not your status line.

## 6. Your `.vimrc` Is a Manifesto, Not a Library

Write it like a letter to your future self:

*"I used to think I needed 40 plugins. Now I know: I only needed `:help`, `:map`, and patience."*

*Keep it under 200 lines.*

*Comment like you're explaining to a 12-year-old who just got their first computer.*

*If it's not teachable, it's not suckless.*

## 7. The Ultimate Suckless Hack: Use Neovim Like a Text Editor

You are not a "developer."

You are a *writer*.

You are not "configuring an IDE."

You are *refining a chisel*.

Type. Delete. Search. Replace. Jump. Repeat.

That's all.

The rest is noise.

---

### ◀ Epilogue: The Suckless Mantra

*"I do not need more features. I need fewer distractions."*

*"I do not need plugins. I need understanding."*

*"I do not need to impress. I need to edit."*

---

You are not installing Neovim.

You are *uninstalling* everything that stands between you and the text.

Welcome to the quiet rebellion.

*"The best editor is the one you don't notice."*

*— Anonymous, probably you, at 3 AM, typing `:wq` with perfect rhythm.*

---

*P.S. If you're reading this and thinking "I need a plugin for that," you're already halfway there. Go delete your `lua/plugins/` folder. You'll thank me tomorrow.*

---

Certainly. Let us begin.

---

## Part I: The Philosophy — Why Suckless Neovim Is Not a Configuration, But a State of Mind

Suckless Neovim is not about minimalism as an aesthetic.

It is not about having a 50-line `init.lua` because it looks cool on GitHub.

It is not about "I only use 3 plugins" as a badge of honor.

**It is about sovereignty.**

You are not a user of Neovim.

You are its *custodian*.

Every plugin you install is a foreign agent in your workflow — a piece of code you do not control, written by someone who does not know your mind, running in your editor, consuming memory, delaying startup, breaking on updates, and subtly training you to depend on automation instead of understanding.

The suckless ethos says:

**"You are the editor. The editor is you."**

This is why native features are sacred.

- `:grep` is not “inferior” to `:Telescope live_grep` — it is *fundamental*.  
It does not require Lua, async, or a dependency tree. It uses the system’s `grep`.  
You know where it is. You can debug it with `strace`. You can replace it with `ripgrep` in two keystrokes.  
You own it.
- `:split`, `:vsplit`, `:tabnew` — these are not “outdated.” They are *atomic operations*.  
You do not need a plugin to “manage windows.” You need to understand how buffers, windows, and tabs work.  
Once you do, you can navigate any codebase with *intention*, not with mouse-driven chaos.
- `:help` is your only true documentation.  
Read `:help usr_02.txt`. Read `:help mode`. Read `:help motion.txt`.  
The manual is not a relic — it is the *source code of your competence*.

Suckless Neovim rejects the modern cult of “instant gratification editing.”

It does not believe in “one key to rule them all.”

It believes in *muscle memory forged through repetition*, in *mental models built over years*, in *the quiet mastery of a tool that never changes — because you learned it deeply*.

You do not need a plugin to auto-format your code.

You need to understand `gq` and `formatoptions`.

You do not need a plugin to jump to definitions.

You need to know `gd`, `gD`, `gf`, and `Ctrl-]`.

You do not need a plugin to search across files.

You need `:vimgrep` and `:copen`.

These are not “hard.”

They are *unlearned*.

The modern editor ecosystem has trained you to believe that complexity is progress.

Suckless Neovim says:

***Progress is clarity.***

***Complexity is decay.***

And so you strip away.

Not because you fear features — but because you *respect your attention*.

You are not trying to be “minimalist.”

You are trying to be *effective*.

And effectiveness is not measured in plugins installed —  
but in problems solved,

in focus maintained,  
in time reclaimed.

This is the first truth of Suckless Neovim Propaganda:

*You are not configuring an editor.  
You are reclaiming your mind.*

---

*We now turn to Part II — the practice. The rituals. The daily acts of rebellion.*

---

Absolutely. Here is **Part II: The Practice — Rituals of the Suckless Neovim Adept**

---

## **Part II: The Practice — Daily Acts of Rebellion**

Suckless Neovim is not a philosophy you read.  
It is a discipline you *live*.

It is not enough to delete plugins.  
You must *relearn how to think*.

Here are the daily rituals — the quiet, unglamorous, deeply powerful habits of those who edit without distraction.

---

### **1. You Start Neovim With No Plugins — Every Time**

*"If you can't edit your config file without plugins, you don't own your editor."*

Every morning, before you open your project, you do this:

```
nvim -u NONE -N
```

Then you open your `~/.config/nvim/init.vim` (or `init.lua`) — and you edit it.

No treesitter. No linters. No auto-complete.  
Just you. And the text.

Why?

Because if you can't *write* a function, *map* a key, or *fix* a syntax error without plugins —  
you are not a programmer.  
You are a plugin operator.

This is your daily calibration.  
Your spiritual cleanse.

After you've edited your config, *then* you restart with your normal setup.  
But now you know:  
*You could do it without.*

---

### **2. You Write Every Mapping by Hand — No Copypasta**

*"If you didn't type it, you don't own it."*

You do not copy `mapleader = '<Space>'` from someone's dotfiles.

You type:

```
nnoremap <space> :<c-u>
```

Then you test it.

Then you break it.

Then you fix it.

You write your own mappings for:

- `jj` → `<Esc>`
- `<leader>w` → `:w`
- `<leader>f` → `:Files (if you must)`
- `<leader>d` → `:Diagnostics`

You don't use `which-key` to show you what your keys do.

You *remember* them.

Because if you need a menu to tell you what your keys do,

you haven't learned them —

you've just memorized a UI.

Your hands know the path.

Your mind doesn't need a map.

---

### 3. You Edit One File at a Time — No Tabs, No Windows, No Panes Unless Necessary

"Multitasking is the enemy of deep work."

You open one file.

You edit it.

You close it.

You do not open 17 tabs because "I might need to look at that later."

You do not split windows to "see the test and the code."

You use `:edit filename` — and you *remember* where things are.

If you need to see two files at once?

Use `:split` — but only when you're actively comparing.

Then close it. Immediately.

Why?

Because every window is a cognitive load.

Every tab is a mental bookmark you're not sure you'll return to.

Every pane is a fragment of your focus scattered across the screen.

The suckless editor does not *manage* files.

It *holds* one file — and holds it *well*.

You do not need a file tree.

You need `:find` and `:e **/filename<Tab>`.

You do not need a project explorer.

You need `:cd ~/proj` and `:e .`

You are not lost.

You are *oriented*.

---

#### **4. You Write Your Own Snippets — No SnipMate, No UltiSnips**

"A snippet is a crutch for not knowing the language."

You do not install a plugin to expand `for` into a 10-line loop.

You type:

```
for i in range(10):
    print(i)
```

And you do it *again*.

And again.

After 20 times, your fingers remember.

If you *must* automate something, write a single function in your `init.lua` :

```
vim.api.nvim_set_keymap('i', 'fpr', '<Esc>:call MyForLoop()<CR>a', { noremap = true,
silent = true })
```

And then define `MyForLoop()` — one time — in your own code.

You do not install 500 snippets for every language.

You learn the *patterns*.

Because when you move to a new language,

you don't need a new snippet pack.

You *think*.

---

#### **5. You Never Use the Mouse — Not Even Once**

"The mouse is the first step toward becoming a user — not a creator."

You do not click "Open File."

You type `:e` .

You do not scroll with the wheel.

You use `Ctrl-u` , `Ctrl-d` , `j` , `k` .

You do not click "Close Tab."

You type `:bd` .

You do not drag to select.

You use `v` , `V` , `Ctrl-v` .

You do not use the scrollbar.

You use `gg` , `G` , `30%` .

Every time you reach for the mouse —

you pause.

You breathe.

You type the key.

This is not about "efficiency."

It is about *integrity*.

The mouse separates you from the machine.

Your fingers on the keyboard?

That's communion.

---

## 6. You Rebuild Your Config From Scratch Once a Year

*"If your config hasn't broken in a year, you haven't learned anything."*

Once a year — on the winter solstice, or your birthday, or the day you finally fixed that bug — you do this:

1. Rename `~/.config/nvim` → `~/.config/nvim.bak`
2. Create a new `~/.config/nvim/init.vim` — 10 lines.
3. Start from scratch.

Only add back what you *missed* — not what you remembered.

You do not copy your old config.

You *reconstruct* it from memory.

You ask yourself:

- Did I really need that color scheme?
- Did I ever use that plugin?
- Was that keybinding ever faster than typing the command?

You'll be shocked.

You'll realize:

*I didn't need 80% of it.*

And you'll feel lighter.

Not because your editor is smaller.

But because *you* are wiser.

---

## 7. You Edit in Silence — No Music. No Notifications. No Status Bars That Lie.

*"The best status line is the one you don't see."*

You disable the status line. Or you make it one line:

```
set statusline=%f
```

No git branch.

No battery.

No Python version.

No "123/456 lines."

Why?

Because your editor should not remind you of the world outside.

It should hold you *inside* the text.

You do not need to know your branch.

You need to know your *intent*.

You do not need to know your CPU usage.

You need to know your *thought*.

You edit in silence.

You work in flow.

You close your eyes when you're stuck — not to look at a plugin's docs —  
but to *think*.

---

## The Final Ritual: The 5-Minute Test

Every Friday, at 5 PM — you do this:

1. Close all files.
2. Open `nvim -u NONE -N`
3. Open a new file.
4. Write 10 lines of code — in any language.
5. Save it.
6. Close Neovim.

No plugins.

No colors.

No help menus.

No Google.

If you can do it —

you are not a Neovim user.

You are a *writer*.

And that —

is the ultimate victory.

---

## 🏁 Epilogue: The Suckless Neovim Practitioner

You do not brag about your config.

You do not post screenshots.

You do not argue about plugins.

You sit.

You type.

You solve.

And when someone asks:

“How do you edit code so fast?”

You smile.

And say:

“I don't. I just stop waiting for the editor to do it for me.”

That is the Suckless Neovim Propaganda.

Not a cult.

Not a trend.

Not a GitHub repo.

A quiet, stubborn, *unyielding* return to the essence of editing:

***Text.***

***Thought.***

***Time.***

And you —

you are the one who remembers.

---

*Now go.*

*Close this tab.*

*Open Neovim.*

*And edit something real.*

---