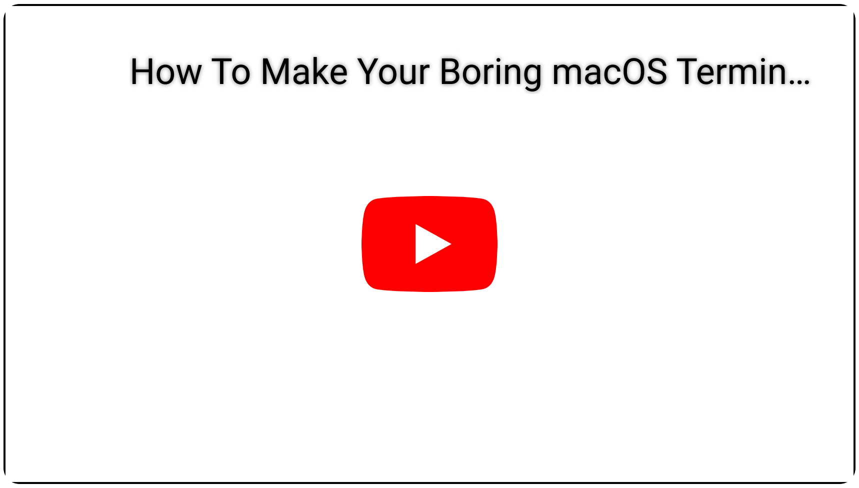


How To Make Your macOS Terminal Amazing With Alacritty

Published: May 1, 2024

After recommendations from you guys, I've swapped out Alacritty for WezTerm! Everything is the same except the Alacritty config file, you can swap it out with this ~/.wezterm.lua file and everything should look and work very similar to Alacritty!

Alacritty is an awesome, minimal and fast terminal setup that has worked great for me as an alternative to iTerm2. Let's set it up!

You can find my dotfiles here

Open The Terminal App

Open the default terminal app on macOs.

This setup is specifically for zsh (default) so make sure you are using that.

You can check by doing:

You can change to zsh if you have it installed by doing:

```
chsh -s /bin/zsh
```

Install homebrew

Run the following command:

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Hom   re
```

If necessary, when prompted, enter your password here and press enter. If you haven't installed the XCode Command Line Tools, when prompted, press enter and homebrew will install this as well.

Add To Path (Only Apple Silicon Macbooks)

After installing, add it to the path. This step shouldn't be necessary on Intel macs.

Run the following command to add the necessary line to `~/.zprofile` :

```
echo 'eval "$(/opt/homebrew/bin/brew shellenv)"' >> ~/.zprofile
```

Now source `~/.zprofile` by doing:

```
source ~/.zprofile
```

Install Alacritty

```
brew install --cask alacritty
```

Install git

Next install git:

```
brew install git
```

Install Meslo Nerd Font

Nerd Fonts are great for showing icons in the terminal. I personally use *Meslo Nerd Font*, but there are others.

You can install a nerd font with homebrew. Take a look at the fonts in the hombrew casks repo: homebrew-cask

It is no longer necessary to tap the homebrew-cask-fonts repo as the fonts have been added to homebrew-cask.

Then you can install the nerd font you'd like

```
brew install font-meslo-lg-nerd-font
```

Setup Alacritty Config File

Next we'll setup the `~/.config/alacritty/alacritty.toml` configuration file to configure Alacritty.

First create the necessary directory:

Then move into it:

```
cd ~/.config/alacritty
```

Now add the `alacritty.toml` file like so:

```
touch alacritty.toml
```

Then open it with your editor of choice. I use Neovim, but you can use whatever you prefer.

To open with Neovim do:

```
nvim alacritty.toml
```

You can also open with Vim:

```
vim alacritty.toml
```

Or TextEdit:

```
open -a TextEdit alacritty.toml
```

Or VSCode:

```
code alacritty.toml
```

Add the configuration to alacritty.toml

```toml
[env]
TERM = "xterm-256color"

[window]
padding.x = 10
padding.y = 10

decorations = "Buttonless"

opacity = 0.7
blur = true

option_as_alt = "Both"

[font]
normal.family = "MesloLGS Nerd Font Mono"

size = 18
```

Install powerlevel10k theme

Powerlevel10k is an awesome theme for zsh.

Install it like so:

```
brew install powerlevel10k
```

Then run the following:

This will add what you need to `~/.zshrc` to enable it.

Now source `~/.zshrc`:

```
source ~/.zshrc
```

The powerlevel10k configuration wizard should show up now.

If you want to open the wizard manually do: `p10k configure`.

Answer the prompts to make the theme look like you would like it to. For the colors of my coolnight theme to work use either lean (with the 8 colors option) or rainbow.

Setup the colorscheme for powerlevel10k and the terminal

Now we'll setup the colorscheme.

Navigate to `~/.config/alacritty`

```
cd ~/.config/alacritty
```

Then clone this repo which has a bunch of different alacritty themes like so:

```
git clone https://github.com/alacritty/alacritty-theme themes
```

Make sure you use the command as shown above to create a `themes` directory inside of `~/.config/alacritty`.

Add my coolnight theme to the themes folder

I've put together my own theme called coolnight, inspired by my previous iTerm2 theme.

```
curl https://raw.githubusercontent.com/josean-dev/dev-environme
```

Now open the `alacritty.toml` file with your editor of choice. With Neovim it would be:

```
nvim alacritty.toml
```

Now add the following to the top of this file:

```
import = [
    "~/.config/alacritty/themes/themes/coolnight.toml"
]
```

Save this file and go back to the command line.

Fix directory background color (only for p10k rainbow style)

If you're using the rainbow version of powerlevel10k, I recommend you change the directory background color from blue to black.

Open `~/.p10k.zsh` with your editor of choice. With Neovim it would be:

```
nvim ~/.p10k.zsh
```

And then look for `POWERLEVEL9K_DIR_BACKGROUND` and change the color from `4` to `0` like so:

```
typeset -g POWERLEVEL9K_DIR_BACKGROUND=0
```

Better zsh history completion with up, down arrows

```
# history setup
HISTFILE=$HOME/.zhistory
SAVEHIST=1000
HISTSIZE=999
setopt share_history
setopt hist_expire_dups_first
setopt hist_ignore_dups
setopt hist_verify
```

This will allow zsh to save the history to a file and configure how it should do so.

Then go back to the command line and run:

```
cat -v
```

Now press on your up and down arrow keys.

Copy the codes that you get as output.

Open the `~/.zshrc` file again and add the following to the bottom of this file:

```
# completion using arrow keys (based on history)
bindkey '^[[A' history-search-backward
bindkey '^[[B' history-search-forward
```

Replace `^[[A` and `^[[B` with the key codes you got for up and down arrow keys if they are different.

Setup zsh-autosuggestions plugin

```
brew install zsh-autosuggestions
```

Then run the following:

```
echo "source $(brew --prefix)/share/zsh-autosuggestions/zsh-auto    g
```

This will add what you need to `~/.zshrc` .

Now source it:

```
source ~/.zshrc
```

Now you can use the plugin! When you get a suggestion and want to use it, use the **right arrow** key.

Setup zsh-syntax-highlighting

This will provide some really nice syntax highlighting as you type out commands.

Install it like so:

```
brew install zsh-syntax-highlighting
```

Then run:

```
echo "source $(brew --prefix)/share/zsh-syntax-highlighting/zsh-    nt
```

This adds what you need to `~/.zshrc` to enable the plugin.

```
source ~/.zshrc
```

You should be getting the syntax highlighting now!

Install eza (better ls)

eza is a better version of ls with lots of different options.

Install it:

```
brew install eza
```

Now you can start using it!

You can create an alias for it in `~/.zshrc` like so:

```
# ---- Eza (better ls) -----

alias ls="eza --icons=always"
```

Install zoxide (better cd)

zoxide is an amazing alternative to cd.

It will remember the directories you've visited in the past and make it really easy to navigate back to them by just typing out a portion of the name of the directory you want to visit.

Install it like so:

Then add the following to `~/.zshrc` :

```
# ---- Zoxide (better cd) ----
eval "$(zoxide init zsh)"
```

If you want to keep using `cd` then create an alias in `~/.zshrc` :

```
# ---- Zoxide (better cd) ----
eval "$(zoxide init zsh)"

alias cd="z"
```

Save and then run:

```
source ~/.zshrc
```

Now you can use `z` as a much smarter replacement to `cd` .

Setup tmux

I have a dedicated video that goes in depth as to how I setup and use tmux.

It is very useful for managing sessions in the terminal as well as different windows and panes.

Let’s install tmux:

```
brew install tmux
```

Then you’ll want to add a config for it and it should be located in `~/.tmux.conf` .

```
curl https://raw.githubusercontent.com/josean-dev/dev-environme
```

Also install tpm (tmux plugin manager):

```
git clone https://github.com/tmux-plugins/tpm ~/.tmux/plugins/t
```

Now you can start tmux by running:

```
tmux
```

Then install the plugins I use with it by pressing `CTRL-A` (my prefix) followed by `Shift-I` .

If this doesn't work or tmux was already running, try reloading the tmux config first:

```
tmux source ~/.tmux.conf
```

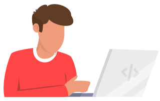
Then try pressing `CTRL-A` followed by `Shift-I` again.

For the tmux theme that I'm using to work properly, you'll probably need to install a newer version of bash:

```
brew install bash
```

Then reload the tmux configuration by doing `CTRL-A` (my prefix) followed by `r` .

You're Done!! 🚀



23 comments · 15 replies *– powered by giscus*

Oldest Newest