

You have **2** free member-only stories left this month. [Sign up for Medium and get an extra one](#)

Customize the MacOS Terminal

Personalize the prompt for increased productivity using Z Shell



Julie Elise

Jul 9, 2020 · 4 min read ★

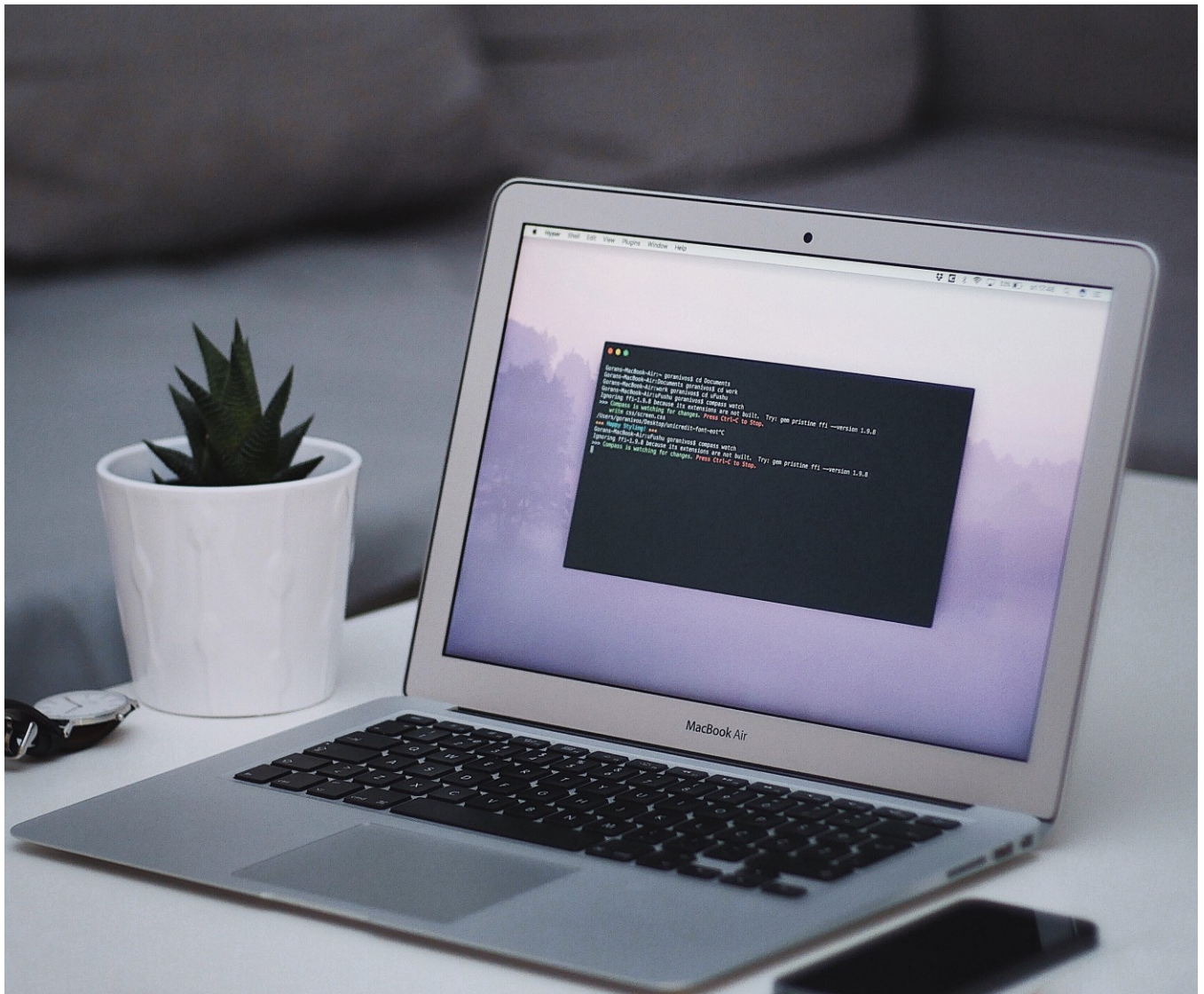
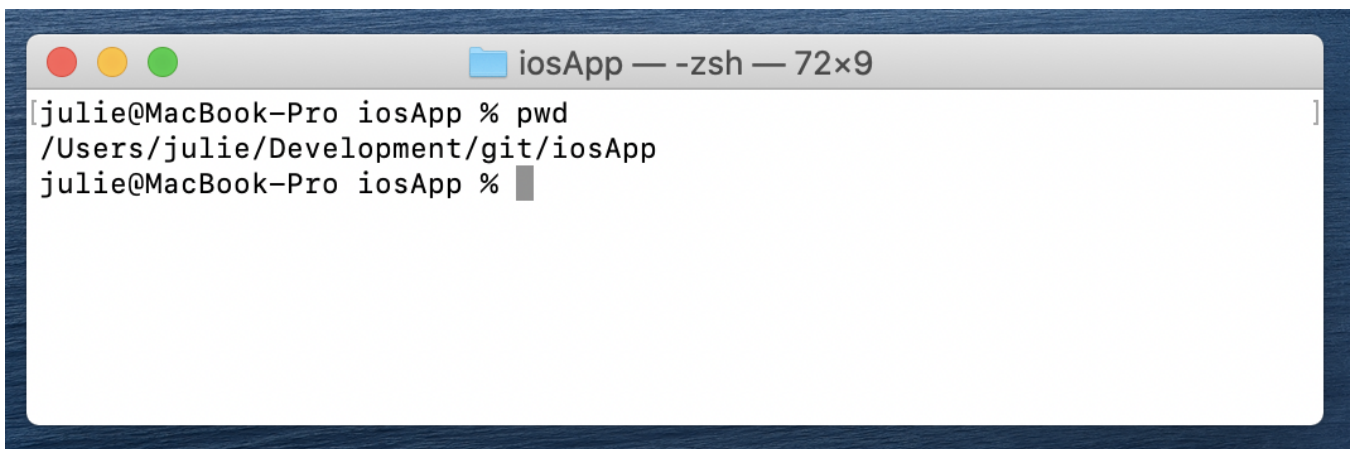


Photo by [Goran Ivos](#) on [Unsplash](#)

As of 2019, macOS Catalina has adopted Z Shell, or `zsh` for short, as the default login shell. Z Shell is a Unix shell that acts as an interactive login shell and command line interpreter for shell scripting.

If you're finding that the prompt isn't displaying enough or too much information, this article will act as a guide in customizing the prompt using a few simple steps. By default, when you open the terminal on any macOS device, the prompt will only display the current directory. Not being able to see the full current directory path can be inhibiting when navigating across many directories. For instance, you may find yourself frequently typing `pwd` to get the full path.



Below we'll go over how to customize the `zsh` prompt.

Create a `~/.zshrc` file

First, you'll need to create a configuration file, `~/.zshrc`, which will include the prompt variables and any other environment variables you may need.

1. Open the terminal
2. Type `touch ~/.zshrc` to create the file.
3. Open the `~/.zshrc` file using `vim`, `nedit`, or any other text editor. If you're accessing the file through Finder, use `Cmd + Shift + .` to show the hidden files. You'll want to add the `PROMPT` variable to this file, as described below.

Customizing the Prompt

The prompt can be overwritten by setting the `PROMPT` variable inside the `~/.zshrc` file.

The different placeholders are described in depth in the [zsh prompt documentation](#).

Below I will highlight some of the most common and useful prompt configurations.

The default terminal prompt shows the username, hostname, and the current directory. This is expressed as:

```
PROMPT='%n@m %. %% '
```

Explanation of components:

```
%n  $USERNAME
%m  The hostname up to the first '.'
%.  Current directory
%%  Character '%'
```

If you want to have the path refer to the full path, this could be configured as:

```
PROMPT='%n@m %~ %% '
```

The only difference here is that `%.` has been replaced by `%~` .

```
%~  Refers to the current working directory, but if the current
working directory starts with $HOME, $HOME part is replaced by a
'~'.
```

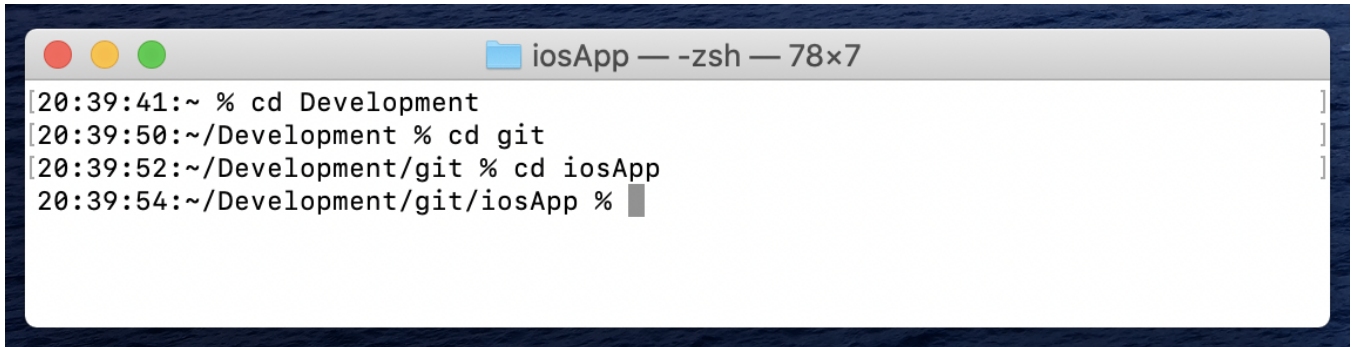
An alternative would be to use `%/`, which evaluates to the absolute current working directory path.

After applying these changes in `~/.zshrc` and opening a *new* terminal window, the full path now shows up in the prompt:

A screenshot of a macOS Terminal window. The title bar shows 'iosApp — -zsh — 80x8'. The terminal content shows a series of commands and prompts: '[julie@MacBook-Pro ~ % cd Development', '[julie@MacBook-Pro ~/Development % cd git', '[julie@MacBook-Pro ~/Development/git % cd iosApp', and 'julie@MacBook-Pro ~/Development/git/iosApp %'. The prompt shows the full path to the current directory.

If you're the only person using the machine and only accessing one machine, perhaps showing the username and machine name isn't important. Here's an example where we're just showing the time and path, where `%*` denotes the time of day, including seconds.

```
PROMPT='%*:%~ %% '
```

A screenshot of a macOS Terminal window titled 'iosApp — -zsh — 78x7'. The terminal shows a sequence of commands and their output: '[20:39:41:~ % cd Development', '[20:39:50:~/Development % cd git', '[20:39:52:~/Development/git % cd iosApp', and '20:39:54:~/Development/git/iosApp %' followed by a cursor. The prompt is not yet colored in this screenshot.

```
[20:39:41:~ % cd Development  
[20:39:50:~/Development % cd git  
[20:39:52:~/Development/git % cd iosApp  
20:39:54:~/Development/git/iosApp %
```

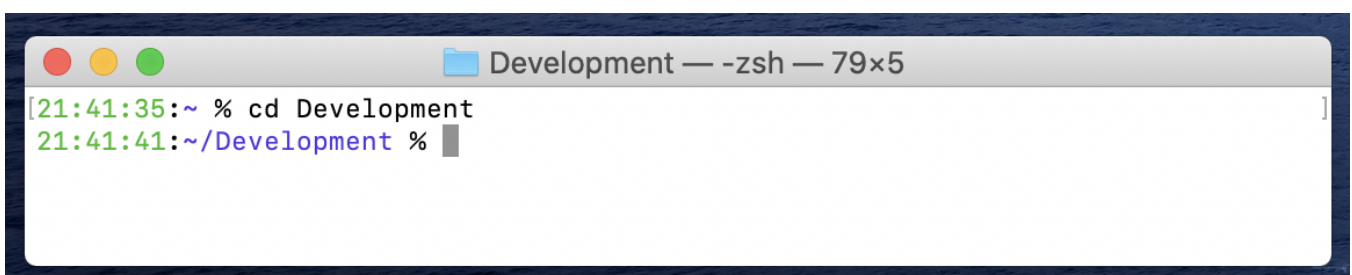
Adding Color

You can also assign a color to different components of the prompt. Zsh supports the color strings black, red, green, yellow, blue, magenta, cyan, white, or any number within the 256 color pallet from 0 to 255. Refer to this link to see a list of all 256 colors.

The color assignment must start with `%F` , meaning foreground color, and end with `%f` . For instance, `%F{red}Label%f` would print the word *Label* in red.

Continuing with our example above, let's say we want the time to be green, but the path to be blue.

```
PROMPT='%F{green}%*%f:%F{blue}%~%f %% '
```

A screenshot of a macOS Terminal window titled 'Development — -zsh — 79x5'. The terminal shows the same sequence of commands as the previous screenshot, but the prompt is now colored: '[21:41:35:~ % cd Development' (time is green, path is blue) and '21:41:41:~/Development %' (time is green, path is blue) followed by a cursor.

```
[21:41:35:~ % cd Development  
21:41:41:~/Development %
```


Additional Resources

- [Zsh prompt documentation](#)
- [Git integration](#)
- [Apple Shell Script Basics](#)
- [Oh My Zsh](#), an open-source framework for managing your Zsh configuration

Conclusion

Making simple changes to the terminal prompt has immensely helped my productivity when navigating across directories and projects using the terminal. This article has covered the basics, however further customization and fine-tuning that can be applied to cater to your exact needs. Thanks for reading!

Sign up for Best Stories

By Dev Genius

The best stories sent monthly to your email. [Take a look](#)

Your email

Get this newsletter

By signing up, you will create a Medium account if you don't already have one. Review our [Privacy Policy](#) for more information about our privacy practices.

Mac Zsh Programming Terminal Unix

[About](#) [Help](#) [Legal](#)

Get the Medium app

<https://medium.com/dev-genius/customize-the-macos-terminal-zsh-4cb387e4f447>

Get the medium app

