



Tweak Your Mac Terminal

Command Line macOS

Daniel Platt

Apress®

Tweak Your Mac Terminal

Command Line macOS

Daniel Platt

Apress®

Tweak Your Mac Terminal: Command Line macOS

Daniel Platt

Andover, Hampshire, UK

ISBN-13 (pbk): 978-1-4842-6170-5

ISBN-13 (electronic): 978-1-4842-6171-2

<https://doi.org/10.1007/978-1-4842-6171-2>

Copyright © 2021 by Daniel Platt

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr

Acquisitions Editor: Aaron Black

Development Editor: James Markham

Coordinating Editor: Jessica Vakili

Distributed to the book trade worldwide by Springer Science+Business Media New York, 1 NY Plaza, New York, NY 10014. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail booktranslations@springernature.com; for reprint, paperback, or audio rights, please e-mail bookpermissions@springernature.com.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/978-1-4842-6170-5. For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

*To my dad, for getting me interested in computers.
I still remember the book called "Computer Fun" which
inspired me to learn more about computers and
programming while I was still in primary school.
Thank you for having such an influence on my interests
and my path in life.*

Table of Contents

About the Author	xvii
About the Technical Reviewer	xix
Acknowledgments	xxi
Chapter 1: Getting Started	1
Terminal at a Glance	1
The Need to Know	4
Book Conventions	6
Book Requirements.....	6
Minimum Requirements	6
Installing Xcode	8
Using Homebrew	10
Why Use Brew?	10
Installing Brew.....	11
Finding Applications in Brew	13
Installing Applications from Brew.....	14
Upgrading Installed Applications	15
Reinstalling Applications.....	17
Summary.....	18
Chapter 2: Terminal Basics.....	19
The Terminal Application.....	19
Customizing Terminal.....	20

TABLE OF CONTENTS

What Is a Shell?	25
Zsh.....	25
Upgrading from a Previous Version of macOS.....	26
Shell Shortcuts.....	27
The Filesystem.....	28
Meta Directories	28
User Directories.....	29
Catalina's New Layout	30
Directory Tree	32
Moving Around the Filesystem.....	33
Inspecting Directories.....	34
Dragging and Dropping.....	35
Different Types of Paths.....	36
Directory Operations	38
Creating Directories.....	38
Deleting Directories	39
File Operations	42
Creating Files.....	43
Writing Files.....	43
Reading Files	44
Copying Files	46
Moving Files	47
Deleting Files.....	47
Editing Text in Terminal	47
Changing the Default Text Editor	51
Piping	51
Pipe.....	52

Redirecting Output.....	52
Redirecting Input.....	54
Hidden Files and Folders.....	55
Hiding Files the UNIX Way	55
Hiding Files the macOS Way.....	58
Globbering.....	60
Single Asterisk.....	60
Question Mark	61
Double Asterisk.....	62
Range Matching	62
Links	63
Hard Links	64
Symlinks.....	64
Naming Things	65
Getting Help with Commands	68
Searching the Manuals.....	69
The “See Also” Section.....	70
Permissions	71
File Permissions	71
File Permissions Breakdown	73
Changing File Permissions	80
Escalating Your System Privilege	82
Aliases.....	84
Customizing the Feel	86
Customizing the Welcome Message	88
Environment Variables	91
Why Use Variables?	93

TABLE OF CONTENTS

The Default Text Editor.....	93
How Can You Use Variables?	93
Customizing the Shell Prompt.....	94
State Symbols	94
Date and Time Symbols	95
Summary.....	99
Chapter 3: macOS Built-in Commands	101
Getting Started.....	101
Compression	102
Zip Compression.....	102
Gzip Compression.....	108
Working with Files	110
Finding Content in Files.....	110
Finding a Filename	114
Finding a Filename with locate	115
Where Is Your Command?.....	116
Inspecting File Contents	118
Working with Text.....	121
Printing Text.....	121
Formatting Output	122
Viewing the Contents.....	124
Transforming Text.....	125
Search and Replace.....	126
Spotting the Difference in Text Files.....	128
Viewing Content At the Beginning of a File	130
Viewing Content At the End of a File	131
Scanning and Processing with AWK.....	134
Working with Compressed Text.....	145

Printing Compressed Text.....	145
Searching Compressed Text	145
Troubleshooting	146
Running Processes	146
Runaway Processes	148
Removing a Drive with an Open File.....	151
Sysctl.....	152
Integration with macOS	153
Open in macOS	153
Copy and Paste.....	156
Using QuickLook on Files	157
Better man Pages	159
System Chime	160
Networking.....	162
Remote Shell	162
Downloading Files	165
Is the Web Server Down?	166
Determining the Path Your Data Takes	168
Pretending a Website Is Somewhere Else	171
Querying DNS	174
Networking Piping	180
Disk Management	184
How Big Is This Folder?	185
Free Space	188
Disk Management.....	191
Disk Images.....	197
Other Commands That I Love	202
The say Command	202

TABLE OF CONTENTS

Scheduling with Launchd	207
Top Commands	218
Summary.....	221
Chapter 4: Installing Commands Using Brew	223
Having Fun with Text.....	223
The fortune Command	223
The cowsay Command	225
The figlet Command	228
The lolcat Command.....	229
The Steam Locomotive Command	230
Media	232
ImageMagick.....	232
ffmpeg	253
Advanced Brew Concepts	266
Brew Casks.....	266
Brew Services	269
Utilities	272
The archey Utility.....	273
Linux Utilities for macOS	274
The ddrescue Command.....	277
The asciinema Command	279
Formatting XML and JSON Files.....	290
The watch Command.....	297
Networking.....	298
The wget Command	298
The curl Command	299
Checking Internet Speed	302
Summary.....	304

Chapter 5: Shell Scripting Basics	305
Your First Script	305
Running and Debugging Scripts	306
Running Scripts from the Current Directory	307
Text Modes	308
Clearing Text Modes	310
Combining Text Modes	311
Text Color	311
Variables	315
Special Variables	315
The If Statement	318
Arithmetic	320
Loops	321
Break and Continue	324
Exit Code	325
Exiting	326
Reading Input.....	326
Projects.....	327
Project 5-1: Print Arguments	328
Project 5-2: Quicker Say Command.....	329
Project 5-3: Higher or Lower	331
Project 5-4: Higher or Lower with Changeable Limits.....	334
Summary.....	339
Chapter 6: PHP.....	341
Determining Your PHP Version	341
Basic Usage.....	342
PHP Configuration.....	342

TABLE OF CONTENTS

PHP Configuration on Cli.....	342
PHP Configuration File.....	343
PHP Configuration in a Browser	344
PHP Web Server.....	344
PHP Basics	346
Language	347
Comments	347
Constants.....	348
Variables.....	350
Superglobals.....	366
Functions.....	371
If Statements	373
Boolean Expression	374
Comparison Operators.....	375
Does the Variable Exist?	376
Loops	377
The for Loop	377
The while Loop	378
The do while Loop	379
The foreach Loop.....	380
Exiting a Loop Early	380
Dependencies.....	383
Web Browser Projects.....	391
Project 6-1: Person API.....	391
Project 6-2: People API	394
Command-Line Projects.....	398

Project 6-3: Arguments.....	398
Project 6-4: Higher or Lower	400
Summary.....	403
Chapter 7: Version Control.....	405
Version Control at a Glance.....	405
Getting Started.....	407
Cloning Repositories	408
Branches.....	410
The HEAD Branch	410
Tags.....	410
Updating the Repository.....	411
Fetch.....	411
Status	412
Pulling.....	413
Your First Repository.....	414
Adding Files to a Repository.....	416
Removing Files from a Repository.....	416
Committing to a Repository	417
Your Identity.....	418
Amending a Commit	419
Repository Log.....	420
Publishing Your Repository	420
Signing Up	421
Creating a Repository	422
Authentication	425
Uploading to the Repository	434
Making Commits in GitHub.....	437

TABLE OF CONTENTS

Forking the Repository	440
Changing the Remote	442
Pull Requests.....	444
Going Further with Git.....	445
Summary.....	446
Chapter 8: Web Development.....	447
Installing a Web Server	447
Installing nginx	448
Starting nginx	449
Configuring nginx	450
Using PHP.....	453
Installing a Database	457
Creating the WordPress Database.....	458
Installing WordPress	460
Using WordPress	467
The Dashboard	468
Settings	469
Viewing the Blog.....	470
Permalinks.....	471
Taking It Live.....	473
Exporting the Database	474
WordPress Config	475
Summary.....	478
Chapter 9: Going Further	479
Brew Clean Up	479
Opting Out.....	480
Manual Clean Up	480

When Things Go Wrong.....	481
Broken Applications.....	483
Internet Services from Terminal.....	487
Weather	487
Stock Ticker	489
Star Wars	490
Nyan Cat	491
Oh My Zsh	494
Installation	495
Customizing Oh My Zsh	496
Ideas for Terminal	515
Mistyping	516
Identify Other Computers	516
Filtering Log Files	519
Programming Languages.....	522
Python	522
C and C++	523
Swift	523
Summary.....	524
Index.....	525

About the Author



Daniel Platt is a Senior Software Engineer at Comparison Technologies Ltd in the UK. He has also produced various online courses around web servers and web development on macOS. Daniel has been a macOS user since the very first MacOS X Public Beta in 2000 and has been building Linux servers since the late 1990s. He is a lifelong computer nerd and spent a bit too much time on his computer

working on his latest project. He found himself spending a lot of time using Terminal for his work and wanted to share his improvement “tweaks” with everybody. The result is this book.

While building web applications, Daniel also sets up web servers from scratch because he has yet to find the perfect hosting solution. His philosophy is “Why settle, when you can build it better yourself?” He even has a course on that topic.

If you’d like to reach out to Daniel, you can do so via his website:
<https://www.ofdan.com/>

About the Technical Reviewer

Ahmed Bakir is an iOS author, teacher, and entrepreneur. He has worked on over 30 mobile projects, ranging from advising startups to architecting apps for Fortune 500 companies. In 2014, he published his first book, *Beginning iOS Media App Development*, followed by the first edition of *Program the Internet of Things with Swift* in 2016 and the second edition in 2018. In 2015, he was invited to develop courses and teach iOS development at UCSD-Extension. He is currently building cool stuff in Tokyo! You can find him online at devatelier.com.

Acknowledgements

With lots of thanks to my wife, Sarah and my two children, Ellie and Livy, for being patient with me while I wrote this book.

I would also like to thank my friend Simon for double-checking and improving the content.

I am very grateful to Apress for taking a chance with me.

Finally, thanks go to my colleagues at Comparison Tech. for suggesting additional ideas.

CHAPTER 1

Getting Started

All personal computers these days come with a graphical user interface (GUI), although this hasn't always been the case. Most computers prior to the release of the Apple Macintosh booted into a terminal, text-only environment. All you would have been greeted with was a flashing cursor.

If your computer didn't come with a user manual, you might have struggled with which commands you should be typing, as some command lines were not very intuitive. If you didn't grow up using a terminal then I can imagine the prospect to be quite daunting. This is where this book comes in! I want to help you to feel just at ease with a terminal as you do with a mouse and the GUI.

Terminal at a Glance

The macOS Operating System comes with lots of preinstalled applications. Some of them you know—Calendar, Mail, Photos, and Safari. Others, you won't be familiar with, and they are more commonly known as commands—this includes `cat`, `cp`, `df`, `echo`, and `rm`.

One application has been bundled with macOS since the first release, back in 2001. The application is normally relegated to the average user as a way to perform tips and tricks that can be found on the Internet. The application is called “Terminal,” and it's shown in Figure 1-1. In this chapter we take a high-level look at this application before digging deeper into it in subsequent chapters.



Terminal
Application - 10.1 MB

Figure 1-1. *The Terminal application*

This book is a journey into a world of Terminal and the hidden commands and utilities that you are unlikely to be aware of. These commands are not like the normal applications, where you use a mouse and GUI to interact with them. Commands do not have a traditional GUI and they operate entirely from the keyboard.

The way to use these commands is by using the Terminal application that is provided with every version of macOS, hidden within the Utilities folder, as shown in Figure 1-2. The Utilities folder can be found inside the Applications folder.



Figure 1-2. *The Terminal application in the Utilities folder*

The quickest way to access Terminal is by activating Spotlight. You do that by pressing $\text{⌘}+\text{Spacebar}$, as shown in Figure 1-3.



Figure 1-3. *The Spotlight search bar*

Then you can type “Terminal” in the search bar and Spotlight will display the application for you to launch, as shown in Figure 1-4.

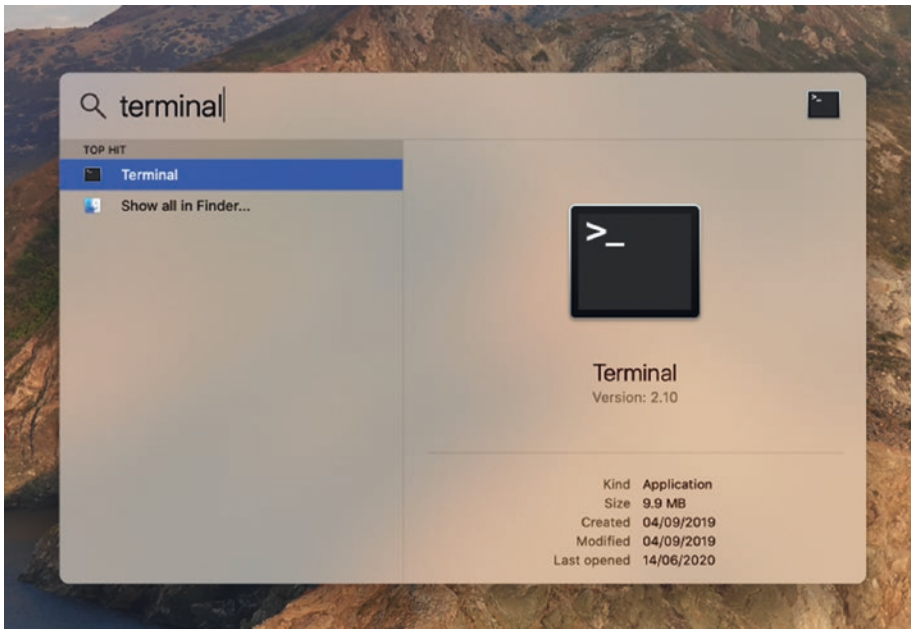


Figure 1-4. Searching for Terminal in Spotlight

The Need to Know

Why am I telling you about Terminal? It lets you type commands into the computer. Using a GUI, you could accidentally move a file to the wrong location. Compare that to when you are using Terminal. You can change the command as many times as you like until you press Enter. Only after you press Enter does the computer execute your command.

Another use of Terminal is to enable hidden features within applications. For example, Safari has a hidden Debug menu that you can only turn on using Terminal, as shown in Figure 1-5.

```
% defaults write com.apple.Safari IncludeInternalDebugMenu 1
```

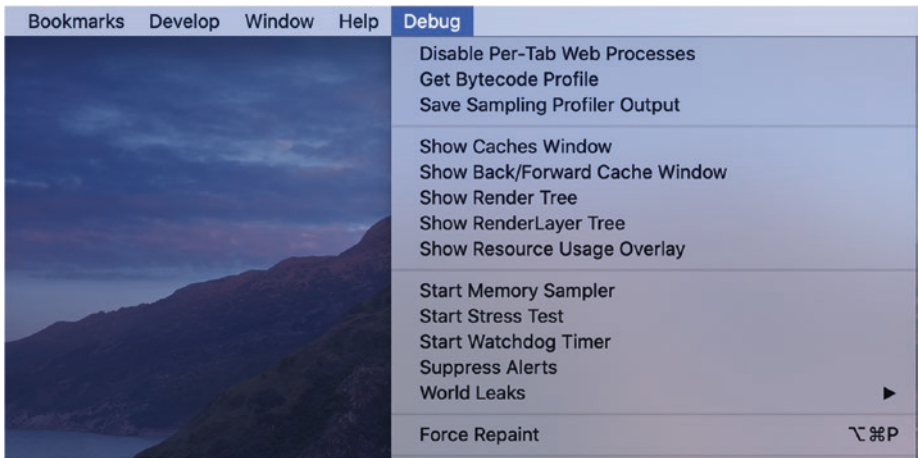


Figure 1-5. *The hidden Debug menu in Safari*

As a web developer, I use macOS to develop my web applications because the macOS Terminal is very similar to the Linux terminal.

If I wanted to copy a folder called `files` from my Desktop to my web server, it is as simple as using the `scp` command.

```
% scp ~/Desktop/files/* user@example.org:~/files/
```

You could download a SCP (Secure CoPy) GUI application, like CyberDuck, but using a GUI application can take longer than using this simple command.

Terminal isn't for everybody, but you should learn to feel comfortable with it. Many activities within the GUI can be performed far quicker within Terminal. This is what you will hopefully come to appreciate over the course of this book.

Book Conventions

Throughout this book, you will see commands you need to type into your terminal, as well as the output they produce.

We use % to refer to the command prompt and put the whole line you need to type in monospace **bold** font.

% **date**

The output appears beneath the command, like so:

% **date**

Tue 11 Oct 2019 10:57:34 BST

If a command is ever within with a block of text, it will be displayed in monospace font as well. For example, to print the current time to the terminal, you use the date command. Sometimes the output will be shown in screenshots to preserve the formatting for ease of viewing.

Book Requirements

To take full advantage of this book, you need your computer to be of a certain standard and have Xcode and Homebrew installed. Let's have a look at these requirements now.

Minimum Requirements

As stated by Apple, the following Apple computers can run Catalina.

- MacBook (Early 2015 or newer)
- MacBook Air (Mid 2012 or newer)
- MacBook Pro (Mid 2012 or newer)
- Mac mini (Late 2012 or newer)

- iMac (Late 2012 or newer)
- iMac Pro (2017)
- Mac Pro (Late 2013 or newer)

I cover the Catalina version of macOS (10.15), which was released on the October 7, 2019. All of Apple's older computers (since 2013) can run Catalina and some from 2012 can also do so. It's a good idea to check whether your Mac is compatible, if you are not already running Catalina.

If you are running an older version of macOS, you might have problems with some commands not existing or being older versions.

We will also be installing *Homebrew*, which is a package manager for the macOS Terminal. It needs to run on macOS High Sierra (10.13) or later.

We will be discussing Homebrew in greater detail later in this chapter and learning what we can install with it in Chapter 4.

If you are using an Apple computer within the last four years, you should have or will be able to upgrade your macOS version to Catalina.

You will need to install either Xcode or the Command Line Developer tools, which will be discussed in the next section, "Installing Xcode." These programs provide the necessary tools for Homebrew to be able to build and install new commands for your terminal.

It's also advantageous to have at least 20-30GB of free space on your Mac, so you can install all the requirements and the extra commands. You never want your Mac to run out of disk space, as many applications cannot function properly without free disk space.

The last requirement is that you need an Internet connection. Some tricks will require an Internet connection to work. Homebrew also requires the Internet to download new files, install new applications, and update existing ones.

Once you have all these things set up, you can proceed.

Installing Xcode

You need to use Xcode's Command Line Developer tools for Homebrew to be able to build new commands and keep them up to date. Thankfully, you can now easily install the Command Line Tools from Apple with a simple command.

```
% xcode-select --install
```

When you enter this command, it should look like Figure 1-6, before you press Enter.



Figure 1-6. *Entering the xcode-select command into Terminal*

You should then see the dialog in Figure 1-7, asking you to confirm installation of the tools.

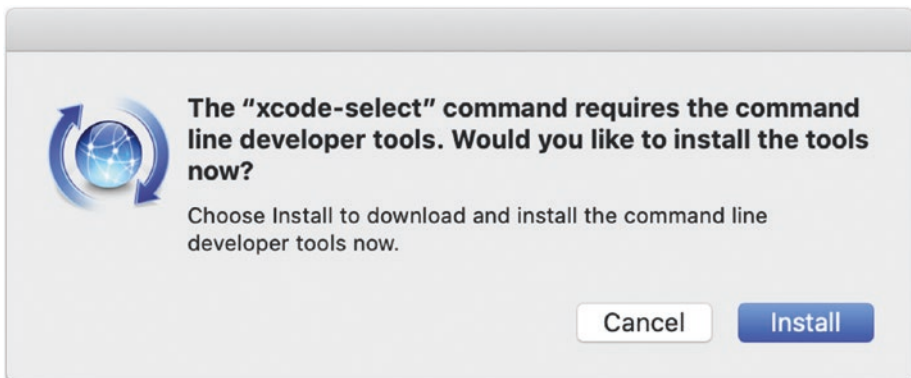


Figure 1-7. *The install dialog for xcode-select*

The Command Line Tools installation requires roughly 500MB of disk space and will install virtually everything Homebrew needs to compile the commands you install.

According to the Homebrew documentation, installing the full version of Xcode will negate the need to install these tools. However, in my experience, after installing Xcode, you still need to install the developer tools.

In some instances, Homebrew will require the full version of Xcode, so for completeness, I will include its installation instructions.

Installing Xcode is as simple as going to the App Store and searching for Xcode, as shown in Figure 1-8.

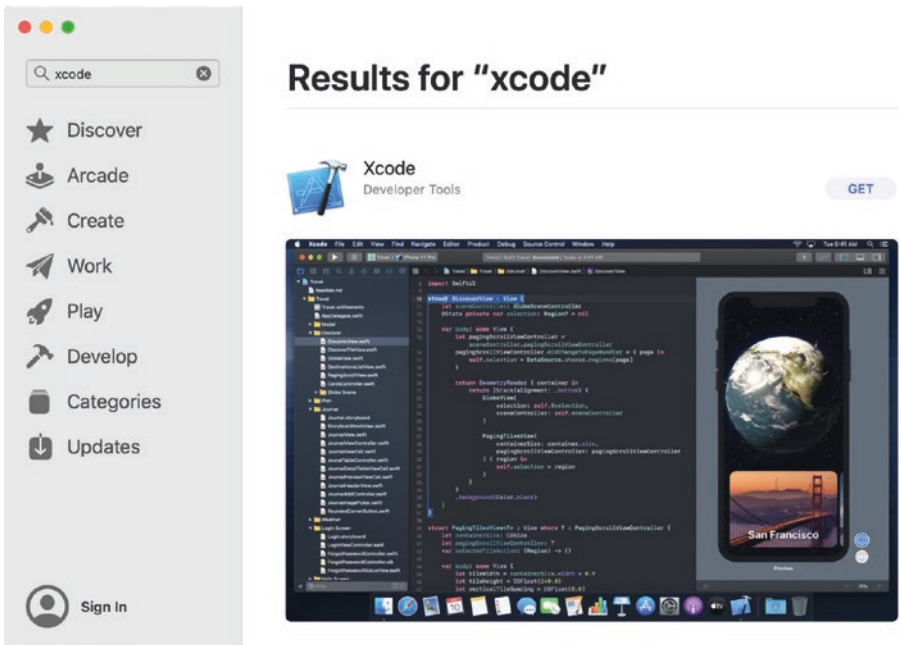


Figure 1-8. Searching for Xcode in the App Store

At present, Xcode is just under 11.2GB in size to download. The installation process will extract Xcode and uses about 30GB.

Using Homebrew

As a macOS user, you can download applications from the Internet with your web browser, or you can use the App Store. Occasionally you will also get new applications as part of the system updates to macOS. Other operating systems, such as Linux and UNIX, have package managers that help maintain the installed applications. Package managers let you perform tasks like searching, installing, upgrading, or removing applications from the system.

Examples of package managers for Linux include `apt`, `yum`, `pacman`, and `portage`. This list is not exhaustive.

The macOS comes with a package manager called the App Store. However, this only deals with software that is approved by Apple and they typically only use a graphical user interface (GUI). There are loads of useful open source programs that run on the command line (Terminal on macOS). You could download Xcode, the source code for the application and all its dependencies. After that, you would have to compile the source code manually yourself. This can get quite messy. So let me tell you about an easier way.

Homebrew is known as “The missing package manager for macOS.” There are other package managers, such as MacPorts, `pkgsrc`, `Nix`, and `Fink`, but I feel Homebrew gets the balance right. When you install Homebrew, you get a new command called `brew`. From now on, I refer to Homebrew as *brew*.

Why Use Brew?

There are thousands and thousands of different applications out there, and installing them all manually could be tedious.

You would first need to download the source code to the application that you wanted. Then you would need to determine all the application’s dependencies. These are in the form of libraries of functions that save developers time in not having to reinvent the wheel.

There are libraries for everything and sometimes there is more than one, as shown in Table 1-1. These libraries provide functionalities to programs, which save you time. For example, `libxml2` provides support for reading a file format known as XML.

Table 1-1. *Libraries and Their Functions*

Library	Purpose
<code>libcurl</code>	File transfer library
<code>libtls</code>	TLS library for certificate verification
<code>libreadline</code>	Allows users to edit commands as they typed them in
<code>libxml2</code>	XML parser

The point is, you need to install all the required libraries, most likely by compiling them from the source. Then you would be able to compile the application you originally wanted.

This is the reason package managers were created. They keep track of every piece of software and all the required libraries. When you install an application, the package manager compares the dependencies to everything that is currently installed and installs whatever is missing.

There is a huge community of developers producing and maintaining software, which can be installed on your computer using Brew. By leveraging Brew, you benefit from all the people before you.

Installing Brew

Installing Brew on your computer is a very straightforward process. By default, Brew will install itself into a shared folder, called `/usr/local`. This will allow you to share installed commands across all users on your Mac. However, only the user who installed Brew will be able to modify the commands.

If you want all users to be able to modify the installed commands, they will need their own installations of Brew.

Installing Brew Globally

The easier way to install Brew is to grab the installation command from the website, at <https://brew.sh/>:

```
% /bin/bash -c "$(curl -fsSL ␣  
https://raw.githubusercontent.com/Homebrew/install/master/  
install.sh)"
```

The installation command is simple but powerful. All you need to do is copy and paste that command into Terminal and press Return. The script will start running and you will get confirmation of where Brew will install itself.

If you are happy for the installation to proceed, then press Enter to continue. If you do not want to carry on, you can press any other button to abort the installation.

Note that you may be prompted for your password if the script requires extra permissions to start the installation.

Installing Brew for Each User

By default, Brew will install itself into `/usr/local`, which can be overwritten.

For example, if you wanted to install Brew into your home directory, you could use these commands.

```
% mkdir ~/homebrew  
% curl -L https://github.com/Homebrew/brew/tarball/master  
% tar xz --strip 1 -C ~/homebrew
```