

Chapter 1

Introduction

You probably spend most of your day reading and writing text in one form or another. Programmers write text in arcane languages with unusual rules. Web developers produce HTML and CSS files by the pound, and those are just another flavor of plain old text. Even Ph.D. candidates tend to generate massive amounts of text that can be typeset using L^AT_EX or similar tools.

Now think about it: If you take the time to learn enough to trim even a few steps off your daily workflow, that's going to add up to be a big win in very little time. The great news is that you can probably save a lot of work with just a little knowledge. This book is full of timesaving techniques that pay off immediately.

So, which text editor should you learn? This is a heated topic in computer geek circles. Those on each side of the famous *vi* versus *emacs* debate are religious about their choice. You can use what you want, but I've worked with a few editors now, and TextMate is the right fit for me. In this short book, I'll show you how to automate much of your daily workflow and try to make a true TextMate believer out of you too.

1.1 A Brief History of TextMate

TextMate was created by Allan Odgaard in 2004 while he was waiting to start a new job. After a couple of months of work, he announced the first release of the editor. It was a little slim.

There were no preferences to set, and you couldn't print with it. Less than two years later, TextMate was awarded the Apple Design Award

for Best Developer Tool. As I write this introduction, TextMate has registered users in the tens of thousands. People have certainly started to take note of the little editor. It even has preferences and can print these days. Allan never quite made it to that other job, but instead spends his time improving TextMate for us users.

What has made TextMate so successful in such a short time is that it combines the power of Unix with the accessibility of Mac OS X. The many Unix tools are a powerful addition to the user's tool belt. Unix editors such as `vi` and `emacs` have been enormously successful because they tie in so well to these tools. For Mac users, though, these editors have painfully foreign interfaces.

TextMate really makes a great effort to expose these helpers to the user from a Mac-friendly interface. The automations really shine for this. You can wrap a few operations useful for a common task together and package them in a single menu item or keystroke. This helps users create tailored enhancements to TextMate for each domain with which they need to work.

Even better, TextMate makes it easy to share these specialized automations with others. That has given birth to a vibrant open source-like community sharing their TextMate enhancements, though the application itself remains proprietary. Today the official TextMate Subversion repository holds more than 125 groupings of automations and languages maintained by dozens of developers.

You can even download tools to work with spreadsheet data¹ or screenplays² in the editor. Odds are good TextMate already covers at least some of your needs, and for the situations where it doesn't, you can train it yourself.

That's what this book is all about. I'll show you how to use the built-in automations and teach you how to take your editing to the next level by building your own. You'll get the raw power of Unix, filtered for Mac user consumption.

1. <http://skiadas.dcostanet.net/afterthought/2006/04/20/more-spreadsheets-in-textmate/>

2. <http://ollieman.net/code/screenwriting/textmate/>

1.2 In These Pages

Going forward, I will cover a subset of TextMate capabilities that can turn anyone into a power user. Different users need to learn different features, though, so I've tried to lay out the information so you can conveniently get to what you need. This book details editing tips, automations, and TextMate's knowledge of languages:

Editing

In this part of the book, I will cover basic text editing with TextMate. Chapters include information for working with TextMate projects, editing text with an emphasis on moving around and making selections, and using *regular expressions* for super find-and-replace operations.

Though these are the basics, they build a foundation for working with TextMate's automations, and I'll sneak in enough tricks to help you impress your geek friends.

Automations

Here we will delve into the heart of TextMate. I'll show you how to use the most popular automations included with the application. You will get a lot of mileage out of these tools, I promise.

From there, this part turns to building your own automations so you can customize how TextMate helps you work. I'll show you how to build *snippets* to insert bits of customizable content into any document, how to record and replay your way past repetitive changes with *macros*, and how to introduce intelligent behaviors into TextMate with *commands*. This part closes with some tips from the automation pros.

Languages, preferences, and themes

The last leg of our journey provides detailed coverage of how to teach TextMate to read new languages and how to build themes to support them. I will also cover how to tweak TextMate's smart editing features.

This is your chance to fit your pet dialects into the TextMate mold, allowing you to build custom automations targeted specifically at the needs of the language.

Along the way you will also find the following:

Menu References

When describing TextMate's features I will often need to refer to commands selected from the menus. I will use a common shorthand for this, so when you see Edit → Find → Find, that means that you should select Find from the Find submenu in the Edit menu. My hope is that the shorthand is actually more natural, since it displays the items in the order you need to select them.

Live code

Most of the examples used in this book come from a bundle of automations I built while writing the book. You can download³ this bundle and play with the examples inside TextMate. If you see a marker line like the one that follows at the top of a listing in the book, that means you can find the code in the download:

```
Download language_grammars/json.textmate
constant = {
  name = 'constant.language.json';
  match = '\b(?:true|false|null)\b';
};
```

If you're reading the PDF version of this book and if your PDF viewer supports hyperlinks, you can click the marker, and the code should appear in a browser window. Some browsers (such as Safari) might mistakenly try to interpret some of the code as HTML. If this happens, view the source of the page to see the real source code.

Joe Asks...

Joe, the mythical developer, sometimes pops up to ask questions about stuff I talk about in the text. I answer these questions as I go along.

Before you jump right into playing with the editor, I need to cover one last convention used in this book.

1.3 The Mac Keyboard and Mouse

To really learn TextMate, you must break down and learn a set of keyboard shortcuts. It will change the way you work, I promise. TextMate

3. From <http://www.pragmaticprogrammer.com/titles/textmate/code.html>

will show you the keystrokes in several parts of the interface. Here's your cheat sheet:⁴

- ^ The Control or Ctrl key.
- ⌥ The Option or Alt key.
- ⇧ The Shift key.
- ⌘ The Command key. This always has an Apple logo on the key.
- The Tab key.
- ⇧← Not an actual key. This back-tab is produced when pressing Tab while holding the Shift key.
- ↵ The Return key.
- ⌘ The Enter key—Fn plus Return on a laptop without a dedicated key.
- ⌫ The Delete or Backspace key.
- ⌘ The Del key—Fn plus Delete on my laptop keyboard. This is often referred to as *forward delete*.
- ⌘ The Escape or Esc key.
- ← The left arrow key.
- The right arrow key.
- ↑ The up arrow key.
- ↓ The down arrow key.
- ⌘ The Page Up key—Fn plus the up arrow on my laptop keyboard.
- ⌘ The Page Down key—Fn plus the down arrow on my laptop keyboard.
- ⌘ The Home key—Fn plus the left arrow on my laptop keyboard.
- ⌘ The End key—Fn plus the right arrow on my laptop keyboard.
- ⌘ The Help key. This key is not available on laptop keyboards.

You should know three other facts about Mac keyboard shortcuts as they are displayed both in TextMate and in this book. First, keyboard shortcuts with letters are always shown with the capital letter for easier identification, but you only need to add a ⇧ to the keystroke if the symbol is actually displayed in the keyboard shortcut. For example, ^R doesn't require a ⇧ even though the R shown is capitalized, but ^⇧R would because you see the symbol.

Along similar lines, symbol-based keystrokes such as ^< may not display a needed ⇧. On some keyboards, that action involves two keys, but on my U.S. key layout I need a ⇧ to type a < character. Thus, for me the keystroke is actually ^⇧<.

4. If you would like to be able to type these glyphs yourself, refer to the documentation at <http://macromates.com/blog/archives/2006/07/10/multi-stroke-key-bindings/>.

A third tip to make note of is that laptop users trying to use keyboard shortcuts involving the function keys may actually adjust their screen brightness instead. When this happens, you can add the Fn key to the keystroke to get it to activate. However, the operating system seems to have a glitch regarding these keystrokes that sometimes requires you to press the Fn key after any other modifier keys. So if the TextMate keystroke is ⌘F2, you might need to press ⌘, then Fn, and then F2. If you get tired of this finger dance and don't mind relegating brightness adjustment to the Fn-enhanced keystroke, there is a checkbox for this in the Keyboard tab of the Keyboard & Mouse panel of System Preferences, available from the Apple menu.

I will also cover how to use the mouse with TextMate. Most Apple machines still ship with a one-button mouse. You could always right-click on Mac OS X with a one-button mouse by holding down the ^ key and clicking. In this book, two-button-mouse users should translate a click into left-click and ^-click into right-click.

1.4 Installing TextMate and Tools

If you don't already have TextMate installed, drop by the TextMate website <http://macromates.com/>, click the Download link in the top-right position of the sidebar, and drag the application out of the archive you downloaded and onto your Mac's Applications directory. Installing TextMate is really that easy, but keep reading for details on a few more helpers you can add.

When you first launch a recent version of TextMate, it should open an Enhanced Terminal Usage dialog box, offering to install the *mate* command-line tool. If you missed that dialog box, you can always get back to it by choosing Help → Terminal Usage. Either way, I highly recommend doing the install. The *mate* tool is described in the sidebar on the following page; I'll show you more tricks with it as we move along.

You don't want to miss one other install. If you haven't already, select Bundles → TextMate → Install "Edit in TextMate". This installs an input manager, which will allow you to edit the content of other applications, such as Mail and Safari,⁵ using TextMate. Try it: Open Safari, browse to <http://google.com/>, place your caret in the search box, trigger Edit → Edit in TextMate (^⌘E), type a search topic, and save (⌘S) and close (⌘W)

5. This works only for Cocoa applications.

Command-Line TextMate

Though TextMate is a modern Mac application with a nice user interface, it can interact flawlessly with Unix tools accessed from the Terminal. The bridge is the `mate` command-line tool.

Adding `mate` to commands will cause TextMate to activate and allow you to create or edit some content, which can be further processed by other commands. To edit a file in TextMate, just invoke `mate file_name_here`. You can also pipe content into a TextMate window for you to work with. For example, this command shows a directory listing in TextMate: `ls | mate`.

Another great feature of `mate` is its ability to wait on you to edit and return the content to the calling process. This allows you to use it as a standard Unix editor by adding the `-w` flag. For example, my bash `.profile` file (loaded each time I start a new session in the Terminal) contains the following lines that allow me to use TextMate as my editor for version control commit messages, and much more:

```
export EDITOR="mate -w"
export CVSEDITOR="mate -w"
export SVN_EDITOR="mate -w"
```

There's more to the `mate` command, so be sure to read the built-in Help by typing `mate -h` in the Terminal.

the TextMate document to push the content back to Safari. TextMate isn't a lot of help for content this size, but imagine editing wiki pages or long emails. It's worth the effort to memorize that keyboard shortcut for these occasions.

A final step I recommend taking to set up TextMate is to "live on the edge." Many pieces of software offer "edge" releases nowadays for those who like being the first to explore new features. Often these releases are only lightly tested and still pretty fragile. TextMate is evolving fast now, though, with a decent focus on refinement. This means the cutting-edge versions usually have more bug fixes and are actually safer. I think that's worth being hassled for a few seconds every couple of days by the automatic upgrade system. If you agree, you can turn it on by selecting TextMate → Preferences (⌘.), clicking the Software Update icon, and changing the Watch For menu to Cutting-Edge.

If You Already Had TextMate Installed

It is possible to download extras for TextMate, and it's common for people to do this. Unfortunately, this sometimes hides some built-in enhancements if you forget to keep the add-ons up-to-date. Therefore, I recommend you do not install any extras you don't need.

This book deals only with items included in a regular TextMate install, and I don't begin to cover all of it. TextMate itself has a lot of functionality, so be sure it in fact doesn't meet your needs before you decide to add on to it.

If you have downloaded any extras and you have trouble with the steps in this book, I recommend returning to a default TextMate install.⁶ If you still cannot find some item I refer to in the Bundles menu, you probably have it disabled. You can check what TextMate is currently loading by choosing Bundles → Bundle Editor → Show Bundle Editor (^ ⌘ B) and clicking the Filter List button at the bottom of that dialog box. Feel free to disable any languages you don't usually work with,⁷ but see Chapter 5, *Built-in Automations*, on page 59 for some details about what I find useful.

If you are in need of something missing from the default install, Get-Bundle is the safest and easiest tool to use to add on to TextMate. See <http://macromates.com/blog/archives/2006/08/21/getting-more-bundles/> for more information.

1.5 The Editing Window

When you are past all the installing, you should be greeted by the editing window when you launch TextMate. You will spend plenty of time interacting with this window, so it's worth recognizing the various parts of it. Use Figure 1.1, on the following page, as a reference.

Aside from the typical Mac window elements such as the Close, Minimize, and Maximize buttons in the title bar and the scroll bar down the right side, a TextMate window has two unique elements. First, the bar that runs down the left side of the window is known as the *gutter*. In the

6. See <http://macromates.com/wiki/Troubleshooting/RevertToDefaultBundles>.

7. Do mind the warning at the bottom of the dialog box about the bundles that should not be disabled, though.

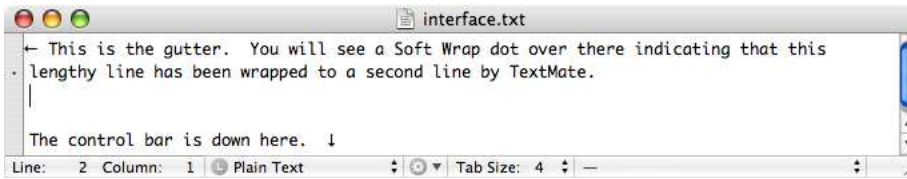


Figure 1.1: The editing window

figure you can see the Soft Wrap indicator dots there. You can configure exactly what shows up in the gutter through View → Gutter.

Second, the bottom of the editing window contains a control bar with various items. The first panel gives a readout of the current caret position in the document (line and column position). Following that, you have a set of four pop-up menus.

The first is the language menu that you can use to set the language for the current document. This influences TextMate's syntax highlighting, among other features. The second menu is the automation menu, which is just a shortcut for reaching most of the items in the Bundles menu. Users tend to access this menu via $\wedge \textcircled{S}$, use the arrow keys to navigate, and press \hookrightarrow to make a selection. The third menu is for controlling TextMate's tab behaviors described in Section 3.3, *Working with Tabs*, on page 43, and the fourth is the symbol menu described in Section 3.1, *Moving to a Line, Symbol, or Bookmark*, on page 31.

Now that you know what these elements are called, it's time to find out what you can accomplish with them!