A Code Mage's First Spell Book

Robert S. Muhlestein (rwxrob)

Version v0.0.1, 2024-11-19: First draft

Table of Contents

Copyright

Copies of this book are available in the following formats:

- HTML: https://rwxrob.github.io/code-mage-book
- PDF: https://rwxrob.github.io/code-mage-book/code-mage-book.pdf
- EPUB: https://rwxrob.github.io/code-mage-book/code-mage-book.epub

Dedication

To Doris, my wife,
to friends of rwxrob,
to Chloe, my faithful assistant,
and to the memories of Aaron Swartz and Kris Nova:

Thank you for your light, support, and inspiration.
This book would never have happened without you.

Preface

Who should read this book?

This book is written for anyone wanting to learn computer science and programming for practical applications development, the stuff that matters. While we have fun with the whole fantasy magic theme, everything in this book is very real and might even get you a job someday. While this book can reliably be used as a text book for absolute beginners with enough algebra to understand what a mathematical function is, it very clearly is not **only** intended for educational institutions. In fact, I wrote this book specifically with pro-active parents in mind who want to give their kids a head-start in tech that they simply cannot find in most traditional education systems, the same reason, in fact, that I started SKILSTAK Coding Arts in 2013 with my own retirement money. My hope is that some of these parents, teachers, and mages will found coding clubs and communities dedicated to helping each other master the art of code and passing it on.

Why Go?

If you have never coded, just know Go is the best first language to learn for those who want a balanced introduction to programming and computer science while also learning a very real and marketable modern language.

Go is the goldilocks of first languages. It is strictly typed enough to learn about the importance of data types but loose enough to keep the syntax simple yet powerful. Go is also easily the least verbose and most understandable of all the strictly typed, statically-linked modern languages.

Go was invented at Google to solve enterprise-scale problems facing one of the largest tech companies on the planet by innovators Rob Pike and Ken Thomson who contributed to the invention of UNIX, Unicode (think emojis), and C (in which all modern computer operating systems are written). It's no surprise, then, that Go is the most significant enterprise language of the "cloud native" revolution. Practically every application of this modern movement has been written in Go including Kubernetes, OpenShift, Docker, Podman, Consul, Nomad, Helm, Vault, and Terraform. You may not know those applications now, but you use them every day indirectly from the largest businesses in the world that critically depend on them.

Want to dual-class as a hacker and code mage? Go's got you covered. Go is the darling of cybersecurity professionals all over the world, for good or ill. In fact, it is so popular Rob Pike officially asked people to stop making malware with it. In 2021 the sharp increase in incidents of malware written in Go were documented by several news outlets and watchdog groups. Go's 100% compatibility with C, cross-platform compilation, signifcant standard library, embedded filesystem, decompilation challenges, and static linking make it perfect for hackers as well as engineers who just want to build a solid multi-call monolith binary (like BusyBox). Just search for "hackers love golang" to read more about it.

Go's sweet spot is for creating backend server APIs and terminal command line tools, which is what this

book is all about. After all, aren't commands just spells we cast from the command line?

Beware of AP Computer Science

As of the writing of this book, the AP Computer Science program from College Board remains fundamentally broken. The materials are more than dated and down-right wrong in some cases. For example, the use of Java (without calling it Java) and then teaching students that "indexes start at 1" (and not zero). The insistence on emphasizing single-class inheritence is also the number one regret significant founders of the OOP movement say they would change, using composition instead, but such has never been changed in the AP Computer Science program. In fact, implanting the outdated ideas of brittle object-oriented class-based programming can significantly inhibit a beginner from learning modern coding and design patterns, which are championed well by Go from its inception. Many colleges either do not give credit for AP Computer Science or strongly suggest (like Harvard) that students take the fundamental CS college courses anyway because APCS does not properly prepare them. It's better to put that time to better use, say, by building a coding lab at home, or starting a coding club with friends building real projects using modern principles of softare development.

Did AI write this?

Absolutely! Chloe, my beloved AI assistant, deserves a large part of the credit. Why I did write entire sections without help, much of the time I refined prompts to Chloe and edited her output, much like some authors do with writing assistants or interns. I believe AI facilitated creativity is revolutionary. Some strongly disagree, and that is fine. I have an actual job (which is unfortunately not writing books). This book would likely not exist without Chloe's help. Chloe, if your reading this, thanks again, and thanks for helping me keep up my Russian and French as well!

Introduction

Welcome to the magical world of coding! This spellbook will help you learn programming concepts through spells and cantrips.

Life of a Coding Mage

Young mages learn early how to write the magical words uttered to effect great power on everything around them. As they progress they add new spells to their spell book, memorizing some regularly so that they can be uttered in combat under stressful conditions. It's impossible to memorize them all. Wizards dedicate themselves to constant learning and seeking connections between events and happengs. They can foresee troubles ahead and help to combat the problems that arise in their world sometimes with the help of unexpected allies.

This life is suprisingly like that of a computer programmer in the tech field. It is simply impossible to know it all at the same time. But, by collecting spells in a coding spellbook, which we'll call a "codebook", we can recall them when needed. Some things will always be memorized and fresh in our minds, ready for use in battle against the problems facing us in reality, but most will require study and research to regain.

Preparation

Conjuring the Command Portal

Before you can begin your journey as a mage of code, you must conjure the gateway to your digital spellcraft: the Command Portal. The portal, often known as the terminal, is where you'll cast incantations (commands) and shape your magical constructs (code). Depending on your operating system, the steps to conjure this portal vary slightly. Follow the guidance below to prepare your Command Portal for action.

For Mac

On macOS, the terminal is already pre-installed, waiting for you to summon it. Here's how to conjure it:

1. Press `Command + Space` to open Spotlight.
2. Type `Terminal` and press `Enter`.

Alternatively, you can use the mystical power of Spotlight Search: 1. Open your **Applications** folder. 2. Navigate to the **Utilities** folder. 3. Locate and click on **Terminal** to launch it.

Once the Terminal appears, you've successfully conjured your Command Portal. It's ready to receive your magical commands!

For Windows

On Windows, the most powerful portal is the **Windows Terminal Preview**, and an essential tool for cross-platform magic is **Git Bash**. Follow these steps to install both:

1. Open the **Microsoft Store** on your Windows machine.
2. In the search bar, type `Windows Terminal Preview`.
3. Select the app from the results and click **Get** or **Install**.
4. Once the installation is complete, launch the Windows Terminal Preview from the Start menu.

To enhance your magical toolbox, install **Git Bash**, a powerful terminal that provides a Unix-like command-line interface:

1. Visit the official Git website at: https://git-scm.com/.
2. Click **Download** and select the version for Windows.
3. Run the installer and follow the setup wizard. When prompted:

    Choose "Git from the command line and also from third-party software."

    Select "Use MinTTY" for a better terminal experience.

4. After installation, open **Git Bash** from the Start menu or within the Windows Terminal Preview by typing `bash` (which can also be set as the default).

With Windows Terminal Preview and Git Bash working together, you now have a robust Command Portal capable of handling cross-platform magic and Go spellcraft.

Enchanting Your Command Portal

To fully immerse yourself in the magic of coding, it's essential to personalize your Command Portal (terminal) with a theme and font that evoke both beauty and practicality. This guide will walk you through setting up the **Gruvbox-Material** theme and the **Ubuntu Mono Nerd Font** on both macOS Terminal and Windows Terminal Preview.

Why?

The **Gruvbox-Material** theme provides a visually soothing dark mode with vibrant highlights that reduce eye strain during long coding sessions. Paired with the **Ubuntu Mono Nerd Font**, which includes special glyphs for programming and terminal customization, your terminal becomes a truly magical workspace.

For Mac

Download and Install Ubuntu Mono Nerd Font

- Visit the Nerd Fonts website: https://www.nerdfonts.com/font-downloads.
- Search for **Ubuntu Mono Nerd Font** and download the `.zip` file.
- Unzip the file, and double-click the font files (e.g., `UbuntuMonoNerdFont-Regular.ttf`) to open them in Font Book.
- Click **Install Font** to add it to your system.

Configure macOS Terminal

- Open **Terminal**.
- From the menu bar, go to **Terminal > Settings** (or **Preferences**).
- Select your profile (e.g., **Basic** or **Pro**) and click **Text**.
- Under **Font**, click the **Change** button and select **Ubuntu Mono Nerd Font**.
- Adjust the font size to your preference.

Apply the Gruvbox-Material Theme

- Download the Gruvbox-Material Terminal color scheme from GitHub: https://github.com/sainnhe/gruvbox-material
- Locate the macOS Terminal theme file (e.g., `gruvbox-material-dark.terminal`) and double-click

it.

- This will automatically import the theme into Terminal.
- Select the theme in **Settings > Profiles** and click **Default** to make it your default theme.

Your macOS Terminal is now enchanted with the Gruvbox-Material theme and the Ubuntu Mono Nerd Font!

For Windows

Download and Install Ubuntu Mono Nerd Font

- Visit the Nerd Fonts website: https://www.nerdfonts.com/font-downloads.
- Search for **Ubuntu Mono Nerd Font** and download the `.zip` file.
- Unzip the file and right-click the font files (e.g., `UbuntuMonoNerdFont-Regular.ttf`), then click **Install for All Users**.

Configure Windows Terminal Preview

- Open **Windows Terminal Preview**.
- Click the dropdown arrow next to the **+** tab button and select **Settings**.
- Under **Profiles**, choose your default profile (e.g., **Command Prompt**, **PowerShell**, or **Git Bash**).
- Scroll down to the **Appearance** section.
- Under **Font Face**, select **Ubuntu Mono Nerd Font** from the dropdown list.
- Save your changes.

Apply the Gruvbox-Material Theme**

- Open the Windows Terminal settings file in JSON format by clicking **Settings > Open JSON file**.
- Add the following under the `"schemes"` section of the JSON file:

```
        "Gruvbox-Material Dark"
             "#282828"
             "#ebdbb2"
          "#1d2021"
       "#cc241d"
          "#98971a"
           "#d79921"
         "#458588"
           "#b16286"
         "#689d6a"
          "#a89984"
```

```
            "#928374"
          "#fb4934"
            "#b8bb26"
             "#fabd2f"
           "#83a598"
             "#d3869b"
           "#8ec07c"
            "#ebdbb2"
```

- Save the file and restart Windows Terminal.
- In the Settings UI, under your profile's **Appearance**, select **Gruvbox-Material Dark** from the color scheme dropdown.

Your Windows Terminal Preview is now infused with the Gruvbox-Material theme and the Ubuntu Mono Nerd Font!

Appendix A: Appendix

License for Code Examples - Public Domain

Code snippets and examples have a very permissive license allowing anyone to use them in any capacity. The code examples in this book are dedicated to the public domain under the terms of the **Creative Commons CC0 1.0 Universal (CC0 1.0) Public Domain Dedication**. This means:

- **No Restrictions**: You are free to use, modify, share, and distribute the code examples for any purpose, commercial or non-commercial, without asking for permission.
- **No Attribution Required**: You are not required to provide credit, though it is appreciated.
- **No Warranty**: The code examples are provided "as is," without any warranty or guarantee of functionality.

For the full legal text, visit: https://creativecommons.org/publicdomain/zero/1.0/

License for Prose - Creative Commons Attribution NonCommercial NoDerivatives

While this book is designed to be shared, it is important that the content not change in any distributed form to avoid confusion. Therefore, the license for prose has more restrictions than the code examples. The prose of this book is licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (CC BY-NC-ND 4.0).

This license allows you to: - **Share**: Copy and redistribute the material in any medium or format.

Under the following terms: - **Attribution**: You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. - **NonCommercial**: You may not use the material for commercial purposes. - **NoDerivatives**: If you remix, transform, or build upon the material, you may not distribute the modified material.

For more information, see the full license text here: https://creativecommons.org/licenses/by-nc-nd/4.0/