參考資料 (References)

第 1 章

1. Rust 在 2023 年榮獲最愛不釋手及最想學的語言 (https://survey.stackoverflow.co/2023/)
2. Rust 維基百科 (https://zh.wikipedia.org/zh-tw/Rust)
3. Everybody is talking about Rust (https://levelup.gitconnected.com/everybody-is-talking-about-rust-elon-musk-microsoft-even-javascript-ecosystem-whats-the-127230449219)
4. Rust crate registry 官網 (https://crates.io)
5. Awesome Blockchain Rust (https://github.com/rust-in-blockchain/awesome-blockchain-rust)
6. 官網 Get started 網頁 (https://www.rust-lang.org/learn/get-started)
7. Microsoft Visual Studio (https://visualstudio.microsoft.com/zh-hant/downloads/)
8. Microsoft C++ Build Tools (https://visualstudio.microsoft.com/zh-hant/visual-cpp-build-tools/)
9. The rustup book 的 MSVC prerequisites (https://rust-lang.github.io/rustup/installation/windows-msvc.html)
10. VS code 官網 (https://code.visualstudio.com/)
11. VS code 官網文件 (https://code.visualstudio.com/docs/languages/rust)
12. Cargo book (https://doc.rust-lang.org/cargo/guide/index.html)
13. Cargo Build Commands (https://doc.rust-lang.org/cargo/commands/build-commands.html)
14. The rustup book (https://rust-lang.github.io/rustup/concepts/toolchains.html)
15. Rust 官方文件首頁 (https://www.rust-lang.org/learn)
16. Rust 官方文件首頁 (https://www.rust-lang.org/learn)
17. 中文 Rust 官方文件首頁 (https://www.rust-lang.org/zh-TW/learn)

18. Rust by Example (https://doc.rust-lang.org/rust-by-example/)

19. tutorialspoint Rust (https://www.tutorialspoint.com/rust/index.htm)

20. awesome-rust (https://github.com/rust-unofficial/awesome-rust)

第 2 章

1. What Happens If We Code the Same Algorithm in Python and Rust? (https://betterprogramming.pub/from-pythonic-to-rustacean-this-is-the-way-6ee46ee63033)

2. What Happens If We Code the Same Algorithm in Python and Rust? (https://betterprogramming.pub/from-pythonic-to-rustacean-this-is-the-way-6ee46ee63033)

3. reveal.js 官網 (https://revealjs.com/)

4. Tauri 官網 (https://tauri.app/)

5. Tauri 文件說明 (https://tauri.app/v1/guides/getting-started/setup)

6. Tauri Prerequisites (https://tauri.app/v1/guides/getting-started/prerequisites)

7. OpenCV-Rust 安裝網址 (https://github.com/twistedfall/opencv-rust/blob/master/INSTALL.md)

8. chocolatey (https://chocolatey.org/)

9. Visual Studio 2022 (https://visualstudio.microsoft.com/zh-hant/vs/whatsnew/)

10. Linfa 套件 (https://github.com/rust-ml/linfa)

11. tch-rs 套件 (https://github.com/LaurentMazare/tch-rs/tree/main)

12. Candle 套件 (https://github.com/huggingface/candle)

13. Burn (https://github.com/tracel-ai/burn)

14. Anaconda 套件 (https://www.anaconda.com/download)

15. PyTorch 套件 (https://pytorch.org/)

16. ImageNet classes (https://deeplearning.cms.waikato.ac.nz/user-guide/class-maps/IMAGENET/)

17. Working With SQL Databases in Rust (https://www.makeuseof.com/working-with-sql-databases-in-rust/)

18. Getting Started with Diesel (https://diesel.rs/guides/getting-started)

19. [1] Rust 配置 diesel 庫 Windows 上安裝配置 (https://blog.51cto.com/u_15060533/4112387)

20. Visual Studio 下載 (https://visualstudio.microsoft.com/vs/community/)

21. awesome-rust (https://github.com/rust-unofficial/awesome-rust)

22. Rust crate registry 官網 (https://crates.io)

第 3 章

1. The Rust Programming Language 的 3.2 Data Type (https://doc.rust-lang.org/book/ch03-02-data-types.html)

2. Rust 【遊樂場】(Playground) (https://play.rust-lang.org/?version=stable&mode=debug&edition=2021)

3. 官方文件 std::fmt 中文說明 (https://rustwiki.org/zh-CN/std/fmt/)

4. How do I print in Rust the type of a variable? (https://stackoverflow.com/questions/21747136/how-do-i-print-in-rust-the-type-of-a-variable)

5. The Rust Programming Language 的【Appendix B: Operators and Symbols】 (https://doc.rust-lang.org/book/appendix-02-operators.html)

6. 麻煩的浮點數 (https://zhuanlan.zhihu.com/p/21520083)

7. 最燒腦的智慧指標 (https://weihanglo.tw/slides/rust-smart-pointers.html#3)

8. The Rust Edition Guide (https://doc.rust-lang.org/nightly/edition-guide/rust-2021/c-string-literals.html)

9. A Comprehensive Guide to String Formatting in Rust (https://medium.com/@teamcode20233/a-comprehensive-guide-to-string-formatting-in-rust-c39a75af8ae6)

10. 官方文件 std::fmt (https://doc.rust-lang.org/std/fmt/)

11. std 函數庫文件說明 (https://doc.rust-lang.org/std/string/struct.String.html)

12. The Rust Programming Language 的【3.17 Strings 說明】 (https://web.mit.edu/rust-lang_v1.25/arch/amd64_ubuntu1404/share/doc/rust/html/book/first-edition/strings.html)

13. Pointers, References and Dynamic Memory Allocation (https://www3.ntu.edu.sg/home/ehchua/programming/cpp/cp4_PointerReference.html)

14. Supercharge Your Rust Code: Harness The Magic Of Smart Pointers (https://pinjarirehan.medium.com/supercharge-your-rust-code-harness-the-magic-of-smart-pointers-c2aa250c5e7c)

15. Stack vs Heap: What's the difference? (https://www.educative.io/blog/stack-vs-heap)

16. What Is Ownership? (https://doc.rust-lang.org/book/ch04-01-what-is-ownership.html)
中文 (https://rust-lang.tw/book-tw/ch04-01-what-is-ownership.html)

17. Box, stack and heap (https://doc.rust-lang.org/rust-by-example/std/box.html)

18. 【The Rust Programming Language】的【15.4 Rc<T> 參考計數智慧指標】(https://rust-lang.tw/book-tw/ch15-04-rc.html)

19. std::collections (https://doc.rust-lang.org/std/collections/index.html)

20. Tutorialspoint Rust - Array (https://www.tutorialspoint.com/rust/rust_array.htm)

21. Rust array 文件說明 (https://doc.rust-lang.org/std/primitive.array.html)

22. Struct std::vec::Vec (https://doc.rust-lang.org/std/vec/struct.Vec.html)

23. Rust: Vectors Explained (https://levelup.gitconnected.com/rust-vectors-explained-189b7e44b49)

24. Rust 程式語言 (https://askeing.github.io/rust-book/README.html)

25. 【std:: result::Result】說明 (https://doc.rust-lang.org/std/result/enum.Result.html)

26. 遊樂場(Playground) (https://play.rust-lang.org/?version=stable&mode=debug&edition=2021)

27. rustlings (https://github.com/rust-lang/rustlings/)

第 4 章

1. StackFlow, What are the alternatives to pattern-matching floating point numbers? (https://stackoverflow.com/questions/45875142/what-are-the-alternatives-to-pattern-matching-floating-point-numbers)

2. 【Rust By Example】的【8. Flow of Control】
(https://doc.rust-lang.org/rust-by-example/flow_control.html)

3. rust: Error handling
(https://blog.devgenius.io/rust-error-handling-61c18f611771)

4. Rust By Example 24.1 open
https://doc.rust-lang.org/rust-by-example/std_misc/file/open.html)

5. The Rust Programming Language，搜尋【unwrap_or_else】
(https://doc.rust-lang.org/core/?search=unwrap_or_else)

6. anyhow 套件 (https://crates.io/crates/anyhow)，文件說明
(https://docs.rs/anyhow/1.0.79/anyhow/)

7. 好用的 crates
(https://suibianxiedianer.github.io/rust-cli-book-zh_CN/tutorial/errors_zh.html
)

8. futility 套件
(https://docs.rs/futility-try-catch/latest/futility_try_catch/macro.try_.html)

9. std::io::ErrorKind (https://doc.rust-lang.org/std/io/enum.ErrorKind.html)

10. Building a Port Scanning Tool in Rust
(https://medium.com/rustaceans/building-a-port-scanning-tool-in-rust-f2667d19d2fc)

第 5 章

1. 命令列手冊
(https://suibianxiedianer.github.io/rust-cli-book-zh_CN/README_zh.html)

2. StructOpt 套件 GitHub (https://github.com/TeXitoi/structopt)

3. WalkDir 套件 (https://docs.rs/walkdir/latest/walkdir/)

4. log 套件說明 (https://docs.rs/log/latest/log/)

5. Complete Guide To Testing Code In Rust
(https://zerotomastery.io/blog/complete-guide-to-testing-code-in-rust/)

6. Rust By Example -- 21.3. Integration testing
(https://doc.rust-lang.org/rust-by-example/testing/integration_testing.html)

7. CommonMark Markdown specification (https://commonmark.org/)

8. Rust By Example -- 21.2. Documentation testing

(https://doc.rust-lang.org/rust-by-example/testing/doc_testing.html)

9. Learning Rust gitbook
(https://learning-rust.gitbook.io/book/lets-get-it-started/modules)

10. config 套件 (https://github.com/mehcode/config-rs/tree/master)

11. confy 套件 (https://github.com/rust-cli/confy)

第 6 章

1. Rust 遊樂場(Playground)
(https://play.rust-lang.org/?version=stable&mode=debug&edition=2021)

2. Rust 程式設計語言 4. 理解所有權
(https://rust-lang.tw/book-tw/ch04-00-understanding-ownership.html)

3. Rust Ownership — Explained for Beginners
(https://medium.com/@vennilapugazhenthi/rust-ownership-explained-for-beginners-de70de16b099)

4. Learning Rust 的 Lifetime
(https://learning-rust.gitbook.io/book/the-tough-part/lifetimes)

5. Rust 程式語言 4.9 生命週期
(https://askeing.github.io/rust-book/lifetimes.html)

第 7 章

1. The Rust Programming Language 的 10. Generic Types, Traits, and Lifetimes
(https://doc.rust-lang.org/book/ch10-00-generics.html)

2. The Rust Programming Language 3.19. Traits
(https://web.mit.edu/rust-lang_v1.25/arch/amd64_ubuntu1404/share/doc/rust/html/book/first-edition/traits.html)

3. std::collections::HashMap
(https://doc.rust-lang.org/std/collections/struct.HashMap.html)

4. CS109 Data Science GitHub
(https://github.com/cs109/2014_data/blob/master/countries.csv)

5. CSV 套件說明文件 (https://docs.rs/csv/latest/csv/)

6. Polars (https://github.com/pola-rs/polars/tree/main)

7. Polars user guide (https://docs.pola.rs/#key-features)

8. Rust Polars: Unlocking High-Performance Data Analysis — Part 2 (https://towardsdatascience.com/rust-polars-unlocking-high-performance-data-analysis-part-2-7c58a3cb7a1f)

9. Rust 語言聖經的 2.8.1. 泛型 (https://course.rs/basic/trait/generic.html)

10. Rust 程式設計語言的 10.1. 泛型資料型別 (https://rust-lang.tw/book-tw/ch10-01-syntax.html)

第 **8** 章

1. 為何說 composition 優於 inheritance？ (https://tw.twincl.com/programming/*662v)

2. Quora, What makes Rust not an object-oriented programming language (https://www.quora.com/What-makes-Rust-not-an-object-oriented-programming-language-even-though-it-has-classes-and-inheritance-support-in-its-syntax)

3. 遇到尾數是五時，單入偶不入 (https://en.wikipedia.org/wiki/Rounding)

4. Rust By Example 16.3. Returning Traits with dyn (https://doc.rust-lang.org/rust-by-example/trait/dyn.html)

5. Rust By Example 16.1. Derive (https://doc.rust-lang.org/rust-by-example/trait/derive.html)

6. lpxxn, rust-design-pattern 工廠模式 (https://github.com/lpxxn/rust-design-pattern/blob/master/creational/factory.rs)

7. Factory Method in Rust (https://refactoring.guru/design-patterns/factory-method/rust/example)

8. Design Patterns in Rust (https://refactoring.guru/design-patterns/rust)

9. lpxxn, rust-design-pattern (https://github.com/lpxxn/rust-design-pattern)

10. The Rust Programming Language 的 17.3. Object-Oriented Design Pattern (https://web.mit.edu/rust-lang_v1.25/arch/amd64_ubuntu1404/share/doc/rust/html/book/second-edition/ch17-03-oo-design-patterns.html)

11. Object-Orientation in Rust (https://stevedonovan.github.io/rust-gentle-intro/object-orientation.html)

第 **9** 章

1. std 參考手冊 (https://doc.rust-lang.org/std/#macros)

2. The Little Book of Rust Macros (https://veykril.github.io/tlborm/introduction.html)

3. Rust By Example 17.1.1. Designators (https://doc.rust-lang.org/rust-by-example/macros/designators.html#designators)

4. [1] The Rust Reference, Macros By Example (https://doc.rust-lang.org/reference/macros-by-example.html)

5. The Rust Programming Language 的 19.5. Macros (https://doc.rust-lang.org/book/ch19-06-macros.html)

6. 10 Python Decorators to Take Your Code to the Next Level (https://python.plainenglish.io/10-python-decorators-to-take-your-code-to-the-next-level-887eac41e2f4)

第 **10** 章

1. JavaScript Arrow Functions and Closures (https://vmarchesin.medium.com/javascript-arrow-functions-and-closures-4e53aa30b774)

2. Rust By Example 的 9.2.2. As input parameters (https://doc.rust-lang.org/rust-by-example/fn/closures/input_parameters.html)

3. 標準函數庫 Enum std:: option::Option (https://doc.rust-lang.org/stable/std/option/enum.Option.html)
   中文：https://rustwiki.org/zh-CN/std/option/enum.Option.html

4. 標準函數庫 Enum std::result::Result (https://doc.rust-lang.org/std/result/enum.Result.html)
   中文：https://rustwiki.org/zh-CN/std/result/enum.Result.html

5. Easy Rust (https://dhghomon.github.io/easy_rust/Chapter_1.html)

6. Easy Rust 的 38. Closures (https://dhghomon.github.io/easy_rust/Chapter_37.html)

7. Rust By Example 的 9.2. Closures (https://doc.rust-lang.org/rust-by-example/fn/closures.html)

8. Easy Rust 48. Closures in functions (https://dhghomon.github.io/easy_rust/Chapter_47.html)

9. The Rust Programming Language 的 13.1. Closures: Anonymous Functions that Capture Their Environment (https://doc.rust-lang.org/book/ch13-01-closures.html)

10. The Rust Programming Language 的 3.23. Closures (https://web.mit.edu/rust-lang_v1.25/arch/amd64_ubuntu1404/share/doc/rust/html/book/first-edition/closures.html)

11. Understanding Closures in Rust (https://medium.com/@ajml/understanding-closures-in-rust-2ca11c9683fd)

第 **11** 章

1. The Rust Programming Language 的 16. Fearless concurrency (https://rust-lang.tw/book-tw/ch16-00-concurrency.html)

2. The Rust Programming Language 的 16.1. Using Threads to Run Code Simultaneously (https://doc.rust-lang.org/book/ch16-01-threads.html)

3. The Rust Programming Language 的 16.2. Using Message Passing to Transfer Data Between Threads (https://doc.rust-lang.org/book/ch16-01-threads.html)

4. The Rust Programming Language 的 16.3. Shared-State Concurrency (https://doc.rust-lang.org/book/ch16-01-threads.html)

5. Rust 語言聖經，第五章 (https://course.rs/advance-practice1/intro.html)

6. Rust 語言聖經, 5.2. 多執行緒版本 (https://course.rs/advance-practice1/web-server.html)

7. Asynchronous Programming in Rust (https://rust-lang.github.io/async-book/01_getting_started/01_chapter.html)

8. 【Coroutine 停看聽】的【Day4：Coroutine 的四大特點】 (https://ithelp.ithome.com.tw/articles/10261501)

9. Asynchronous Programming in Rust 的 2. Under the Hood: Executing Futures and Tasks (https://rust-lang.github.io/async-book/02_execution/01_chapter.html)

10. Tokio 教學文件 (https://tokio.rs/tokio/tutorial)

11. Tokio GitHub 說明 (https://github.com/tokio-rs/tokio?tab=readme-ov-file#related-projects)

12. Rayon 套件 (https://docs.rs/rayon/latest/rayon/)

13. Tokio 參考文件 (https://docs.rs/tokio/latest/tokio/index.html)

第 **12** 章

1. Rust and WebAssembly (https://rustwasm.github.io/docs/book/introduction.html)
2. MDN WebAssembly (https://developer.mozilla.org/en-US/docs/WebAssembly/Rust_to_wasm)
3. Node 下載頁面 (https://nodejs.org/en/download)
4. npm 網站 (https://www.npmjs.com/)
5. 【Rust and WebAssembly】的【Hello, World!】(https://rustwasm.github.io/docs/book/game-of-life/hello-world.html)
6. wasm-bindgen GitHub (https://github.com/rustwasm/wasm-bindgen/tree/main)
7. The wasm-bindgen Guide (https://rustwasm.github.io/docs/wasm-bindgen/examples/index.html)
8. Creating a Small Game with WebAssembly and Rust (https://medium.com/comsystoreply/creating-a-small-game-with-webassembly-and-rust-20c6945efa1d)
9. webassembly-rust-snake GitHub (https://github.com/joern-kalz/webassembly-rust-snake)
10. devserver (https://github.com/kettle11/devserver)

第 13 章

1. 標準函數庫 std::fs (https://doc.rust-lang.org/std/fs/index.html)
2. Open file 的模式選項(OpenOptions) (https://doc.rust-lang.org/std/fs/struct.OpenOptions.html#method.open)
3. encoding_rs (https://docs.rs/encoding_rs/latest/encoding_rs/)
4. encoding_rs【Enums】頁籤 (https://docs.rs/encoding_rs/latest/encoding_rs/#enums)
5. rdpFX GitHub (https://github.com/RickyDane/rdpFX)

第 14 章

1. SQLx (https://github.com/launchbadge/sqlx)
2. SQLx 官方文件 (https://docs.rs/sqlx/latest/sqlx/)

3. SQLx CLI (https://github.com/launchbadge/sqlx/blob/main/sqlx-cli/README.md#enable-building-in-offline-mode-with-query)

4. SQLiteSpy (https://www.yunqa.de/delphi/apps/sqlitespy/index)

5. MariaDB (https://mariadb.org/download/?t=mariadb&p=mariadb&r=11.3.2&os=windows&cpu=x86_64&pkg=msi&mirror=blendbyte)

6. EDB Postgres 官網 (https://www.enterprisedb.com/downloads/postgres-postgresql-downloads)

7. Configuring an ODBC Driver Manager on Windows, macOS, and Linux (https://blog.devart.com/configuring-an-odbc-driver-manager-on-windows-macos-and-linux.html)

8. MIR 18-1 ODBC 與 DSN 簡介 (http://mirlab.org/jang/books/asp/odbc&dsn.asp?title=18-1%20ODBC%20%BBP%20DSN%20%C2%B2%A4%B6)

9. The unixODBC Project home page (https://www.unixodbc.org/)

10. StackOverflow, what is the difference between OLE DB and ODBC data sources? (https://stackoverflow.com/questions/103167/what-is-the-difference-between-ole-db-and-odbc-data-sources)

11. ODBC-API 套件 (https://crates.io/crates/odbc-api)

12. crate.io ORM 統計 (https://crates.io/keywords/orm)

13. A Guide to Rust ORMs in 2024 (https://www.shuttle.rs/blog/2024/01/16/best-orm-rust)

14. Choosing a Rust Database Crate in 2023 (https://rust-trends.com/posts/database-crates-diesel-sqlx-tokio-postgress/)

15. SeaORM GitHub 的 Community (https://github.com/SeaQL/sea-orm/blob/master/COMMUNITY.md#built-with-seaorm)

16. northwind_psql (https://github.com/pthom/northwind_psql/tree/master)

17. MySQL 官網 (https://dev.mysql.com/downloads/installer/)

18. SQLite 官網 (https://www.sqlite.org/download.html)

19. Diesel , All About Inserts (https://diesel.rs/guides/all-about-inserts.html)

20. Diesel , All About Updates (https://diesel.rs/guides/all-about-updates.html)

21. Diesel, Relations (https://diesel.rs/guides/relations.html)

22. Diesel GitHub (https://github.com/diesel-rs/diesel/tree/2.1.x)

23. SeaORM 官方文件 (https://www.sea-ql.org/SeaORM/docs/introduction/orm/)

24. SeaORM Tutorials (https://www.sea-ql.org/sea-orm-tutorial/ch01-00-build-backend-getting-started.html)

25. SeaORM Cookbook (https://www.sea-ql.org/sea-orm-cookbook/)

26. Migration (CLI) (https://www.sea-ql.org/sea-orm-tutorial/ch01-02-migration-cli.html)

27. Migration (API) (https://www.sea-ql.org/sea-orm-tutorial/ch01-03-migration-api.html)

28. SeaORM GitHub (https://github.com/SeaQL/sea-orm/tree/master)

29. MangoDB 官網文件 (https://www.mongodb.com/docs/drivers/rust/current/)

30. Rust Driver Quick Start (https://www.mongodb.com/docs/drivers/rust/current/quick-start/#std-label-rust-quickstart)

31. MangoDB 官網 (https://www.mongodb.com/try/download/community)

32. MongoDB Rust Driver (https://crates.io/crates/mongodb)

33. MongoDB 套件的文件說明 (https://docs.rs/mongodb/2.8.2/mongodb/)

第 15 章

1. After Abandoning C/C++, Microsoft Forms New Team to Rewrite C# Code in Rust! (https://blog.stackademic.com/after-abandoning-c-c-microsoft-forms-new-team-to-rewrite-c-code-in-rust-b90019c685ea)

2. Google 投百萬美元給 Rust 基金會，要強化 C++與 Rust 互通性 (https://www.ithome.com.tw/news/161222)

3. Rust Once Again Chosen for Cost Savings! Rust Replaces Python, Slashing Amazon Cloud Costs by 75%! (https://blog.stackademic.com/rust-once-again-chosen-for-cost-savings-rust-replaces-python-slashing-amazon-cloud-costs-by-75-65f3d1af171c)

4. Windows 套件說明 (https://microsoft.github.io/windows-docs-rs/doc/windows/)

5. Rust For Windows 套件首頁 (https://crates.io/crates/windows)

6. Rust For Windows 的 GitHub (https://github.com/microsoft/windows-rs)

7. 微軟 WIN32 API 的程式設計參考 (https://learn.microsoft.com/zh-tw/windows/win32/api/)

8. MessageBoxW 函式 (winuser.h) (https://learn.microsoft.com/zh-tw/windows/win32/api/winuser/nf-winuser-messageboxw)

9. 【Windows 應用程式開發】的【建立視窗】 (https://learn.microsoft.com/zh-tw/windows/win32/learnwin32/creating-a-window)

10. Native Windows GUI (https://github.com/gabdube/native-windows-gui)

11. Native Windows GUI 畫面截圖(Showcase) (https://github.com/gabdube/native-windows-gui/tree/master/showcase)

12. iced 套件 (https://iced.rs/)

13. iced GitHub (https://github.com/iced-rs/iced)

14. iced GitHub 的 examples 資料夾 (https://github.com/iced-rs/iced/tree/master/examples)

15. CXX-Qt GitHub (https://github.com/KDAB/cxx-qt/)

16. Qt 官網 (https://www.qt.io/download)

17. CXX-Qt 官網教學範例 (https://kdab.github.io/cxx-qt/book/getting-started/2-our-first-cxx-qt-module.html)

18. CXX-Qt GitHub 的 examples/qml_features 資料夾 (https://github.com/KDAB/cxx-qt/tree/main/examples/qml_features)

19. Tauri (https://tauri.app/)

20. egui (https://github.com/emilk/egui)

21. Dioxus (https://dioxuslabs.com/)

22. Node.js 官網 (https://nodejs.org/en)

23. Tauri Prerequisites (https://tauri.app/v1/guides/getting-started/prerequisites)

24. Yarn 下載與安裝 (https://classic.yarnpkg.com/lang/en/docs/install/#windows-stable)

25. Tauri GitHub (https://github.com/tauri-apps/tauri/tree/dev)

26. Tauri Vite (https://tauri.app/v1/guides/getting-started/setup/vite)

27. Tauri Application Debugging (https://tauri.app/v1/guides/debugging/application)

28. Tauri Debugging in VS Code (https://tauri.app/v1/guides/debugging/vs-code)

29. Hyper 套件 (https://github.com/hyperium/hyper)

30. Tiny-http (https://github.com/tiny-http/tiny-http)

31. Actix Web (https://github.com/actix/actix-web)

32. axum (https://github.com/tokio-rs/axum)

33. Rocket (https://github.com/rwf2/rocket)

34. Postman (https://www.postman.com/jp/downloads/)

35. datatables.js (https://datatables.net/)

36. dataTable 新增、讀取、更新、刪除 (https://cwcchannel.com/2024/02/15/新增、讀取、更新、刪除-crud-data-table-系統-javascript/)

第 16 章

1. The Rust Programming Language 的 Unsafe Rust (https://doc.rust-lang.org/book/ch19-01-unsafe-rust.html?highlight=FFI#calling-an-unsafe-function-or-method)

2. vanjacosic, rust-ffi-to-c (https://github.com/vanjacosic/rust-ffi-to-c/tree/main)

3. cc crate (https://crates.io/crates/cc)

4. Rust By Example 的 20.8 Foreign Function Interface (https://doc.rust-lang.org/rust-by-example/std_misc/ffi.html)

5. Effective Rust 的 Item 34: Control what crosses FFI boundaries (https://effective-rust.com/ffi.html)

6. Rust FFI 程式設計 - libc crate (https://rustcc.cn/article?id=3a87a6b8-2f1c-4ac9-b962-5d9578eb5b1a)

7. libc GitHub (https://github.com/rust-lang/libc)

8. Jeremy Mill, Calling Rust from C# (https://dev.to/living_syn/calling-rust-from-c-6hk)

9. The Python Rust-aissance (https://baincapitalventures.com/insight/why-more-python-developers-are-using-rust-for-building-libraries/)

10. PyO3 套件 (https://github.com/PyO3/pyo3)

11. PyO3 使用指引 (https://pyo3.rs/v0.21.1)

12. PyO3 使用指引, Executing existing Python code (https://pyo3.rs/v0.21.0/python-from-rust/calling-existing-code)

13. Making Python 100x faster with less than 100 lines of Rust (https://ohadravid.github.io/posts/2023-03-rusty-python/)

14. ohadravid, poly-match GitHub (https://github.com/ohadravid/poly-match)

第 **17** 章

1. Artificial Intelligence vs Robotics vs Machine Learning vs Deep Learning vs Data Science (https://medium.datadriveninvestor.com/artificial-intelligence-vs-robotics-vs-machine-learning-vs-deep-learning-vs-data-science-70ff828cdf39)

2. Awesome Rust-Machine Learning (https://github.com/vaaaaanquish/Awesome-Rust-MachineLearning)

3. Are we learning yet? (https://www.arewelearningyet.com/)

4. ndarray (https://github.com/rust-ndarray/ndarray)

5. Polars (https://pola.rs/)

6. Plotters (https://github.com/plotters-rs/plotters)

7. Linfa (https://github.com/rust-ml/linfa)

8. Plotters Gallery (https://github.com/plotters-rs/plotters/tree/master?tab=readme-ov-file#gallery)

9. Anaconda (https://www.anaconda.com/download)

10. tch-rs 套件 (https://github.com/LaurentMazare/tch-rs)

11. Candle 套件(https://github.com/huggingface/candle)

12. TensorFlow Rust (https://github.com/tensorflow/rust)

13. burn 套件 (https://github.com/tracel-ai/burn)

14. PyTorch 官網 (https://pytorch.org/)

15. Kaggle MNIST Dataset (https://www.kaggle.com/datasets/hojjatk/mnist-dataset)

16. Pre Trained Models for Image Classification–PyTorch for Beginners (https://learnopencv.com/pytorch-for-beginners-image-classification-using-pre-trained-models/)

17. ms_coco_classnames.txt (https://gist.github.com/AruniRC/7b3dadd004da04c80198557db5da4bda)

第 **18** 章

1. awesome-blockchain-rust (https://github.com/rust-in-blockchain/awesome-blockchain-rust)

2. List of Top Blockchains Using The Rust Programming Language (https://101blockchains.com/top-blockchains-using-rust-programming-language/)

3. Building a Blockchain in Rust (https://casper.network/en-us/web3/web3-development/building-a-blockchain-in-rust/)

4. Mario Zupan Blog (https://blog.logrocket.com/author/mariozupan/)

5. How to build a blockchain in Rust (https://blog.logrocket.com/how-to-build-a-blockchain-in-rust/)

6. libp2p tutorial: Build a peer-to-peer app in Rust (https://blog.logrocket.com/libp2p-tutorial-build-a-peer-to-peer-app-in-rust/)

7. awesome-rust (https://github.com/rust-unofficial/awesome-rust)