

[Redacted text block]

[Redacted text block]

.....
 特集：どうやってシステムの勉強をしていますか？

プログラミングは「怠惰」に楽しく

大谷 周平

.....

はじめに

怠惰 (Laziness) ・ 短気 (Impatience) ・ 傲慢 (Hubris)、この3つはプログラムの三大「美德」であるという箴言があります。¹ 一般的には悪徳とされる単語ですが、それぞれの以下の意味で用いられています。

怠惰…全体のコストを下げるために労力を惜しまないこと、同じ事を繰り返さないでいように再利用可能なプログラムを整備することやドキュメントを整備すること

短気…コンピュータが無精と感じた時や、プログラムが思い通りに動かない時に感じる怒り

傲慢…自分の書いたプログラムに文句を言われないようする傲慢さ

私はこの三大「美德」の中でも、「怠惰」であることが最も重要だと考えています。他の2つは、プログラムの高速化や完成度などスキルを上げるために必要なものです。しかし、「怠惰」であることはプログラミングを学ぶ動機に直結しているからです。本稿は出来る限りやらないで済むことはやらないという方針で、10数年前にプログラミングを学び始めた自分にアドバイスをするのであればという観点で執筆しました。

[Redacted text block]

自分が楽になることを目的に

自分にとって具体的なメリットになることを短期的な目的として設定しましょう。プログラミングは手段であって、目的ではありません。プログラミングで実現したい具体的なことをイメージ出来なければ、継続には強い意志が必要です。しかし一方で、広く一般に公開する Web サービスのような大きな目標だけを設定してしまうことも、実現までに多くの時間を要することにつながり、継続の妨げとなります。まずは、自分が業務や日常で繰り返し行っている作業をコンピュータに肩代わりしてもらうことを目指すと、プログラミングの効力を実感しやすく、継続的な学習に繋がると考えています。

プログラミング以外の方法を考える

目的が明確になっても、いきなりプログラミングに取り組む必要はありません。IFTTTⁱⁱ に代表される自動化ツールや、正規表現の基礎の習得によって、目的が達成できないか検討してみましょう。

IFTTT を利用すると、プログラムすることなくさまざまな Web サービスを相互に連携できます。Microsoft Flowⁱⁱⁱ や Integromat^{iv} といった類似サービスもあります。例えば、Web サイトの更新情報を Twitter へ投稿する、スマートフォンの GPS と Google スプレッドシートを連携して出退勤時間のログを作成する、といった処理を自動化できます。さまざまなレシピが公開されているので、自分がよく利用している Web サービスに関連したレシピをチェックするところから始めて、自動化を試してみてください。

正規表現は、テキストデータの抽出や編集に役立つ技術です。例えば、メールアドレスや ISBN、DOI のように一定の規則性がある情報を抽出する、行頭や行末などに特定の変

更を加える、などの作業は正規表現を使うことでかなり省力化出来ます。正規表現は少し高機能なテキストエディタがあれば、概ね利用できると思います^v。また、大抵のプログラミング言語でも利用できる汎用性の高い技術でもあります。

IFTTT や正規表現だけでは解決できない問題が出てきたタイミングで、プログラミングに取り組みましょう。漫然と入門書に取り組むのではなく、自分が解決したい問題とどのように関連付けられるかと意識するだけで、学習効率が大きく異なるはずで

先行事例を真似する

先行事例を真似しましょう。自分が解決したい問題をすでに誰かが解決していないか調べましょう。運良くプログラムのソースコードが公開されているのであれば、それを自分の目的に合わせてカスタマイズすることで、問題解決のための時間を短縮できます。さらに、他人のプログラムを読むことでプログラミングのスキルも向上します。また、調査の過程で、新たな目標のヒントも見つかるかもしれません。

図書館におけるプログラミングに限らず広く ICT に関する情報源として、2つ紹介します。情報科学技術協会の会誌「情報の科学と技術」^{vi} では、70 巻 4 号 (2020 年 4 月) で特集「インフォプロのためのプログラミング事例集」が組まれています。過去にも Web API の活用やスキルアップ等、プログラミングと関連性の高いテーマがしばしば設定されています。

Code4Lib Japan^{vii} は図書館における情報技術活用を促進し、図書館の機能向上と利用者の図書館に対する満足度向上を目指しているコミュニティです。毎年カンファレンスが開催され、その資料や動画は公開されていま

す。先進的な取り組みから、少し手を伸ばせば実現できそうな取り組みまではさまざまな実践例を把握できます。本家であるCode4Lib^{viii}のカンファレンスやCode4Lib Journalでもさまざまな事例が共有されています。

プログラミングが上手くいかないときも、先行事例を調査しましょう。表示されているエラーメッセージをプログラマ向けのQ&AサイトStack Overflow^{ix}で検索するだけでも、かなりの確率で解決のヒントを得られます。

アウトプットすることでフィードバックを得よう

一定の成果が出た段階で、同僚にも共有しましょう。業務上問題ない内容であれば、ブログ^xやGitHub^{xi}を通じて公開しましょう。自らの学習記録になりますし、アウトプットを意識することで、プログラミングの質も向上します。プログラムの改善点などフィードバックや自分では気づかなかったアイデアを貰えることもあります。また、問題意識を共有出来る人と共同で新たなプロジェクトを実施する契機になるかもしれません。まずは公開しなければ始まりません。先述のCode4Libのカンファレンスで発表することは、私の目標の一つでもあります。

おわりに

繰り返しになりますが、プログラミングはあくまでも手段であって、目的ではありません。出来なくてもそれで、問題が解決するなら十分だと考えています。一方で、プログラミングを身に付けたことで、私のキャリアや交友関係は非常に充実したものになりました。本稿で紹介した、可能な限りプログラムをせずに済ますような「怠惰」なところからでもプログラミングの世界に踏み込んでいただければ、執筆者として望外の喜びです。

- i Larry Wall, Schwartz, Randal L. Schwartz. Programming Perl. O'Reilly & Associates, 1992. (A nutshell handbook). で、プログラミング言語 Perl の開発者である Larry Wall 氏が述べている。
- ii IFTTT, <https://ifttt.com/>, (2020-07-10 参照)
- iii Microsoft Flow. <https://flow.microsoft.com/ja-jp>, (2020-07-10 参照)
- iv Integromat. <https://www.integromat.com/en>, (2020-07-10 参照)
- v 執筆者は Windows では Mery、Mac では CotEditor という無料のテキストエディタを利用しています。プログラミングまで視野にいと、Visual Studio Code がお勧めです。
- vi 情報の科学と技術. <https://www.jstage.jst.go.jp/browse/jkg/-char/ja>, (2020-07-10 参照)
- vii Code4Lib Japan. <http://www.code4lib.jp>, (2020-07-10 参照)
- viii Code4Lib. <https://code4lib.org>, (2020-07-10 参照)
- ix Stack Overflow. <https://stackoverflow.com/>, (2020-07-10 参照)
- x よしなしごと. <http://otani0083.hatenablog.com>, (2020-07-10 参照) GitHub. <https://github.com/otani0083>, (2020-07-10 参照) 執筆者のブログと GitHub リポジトリ
- xi GitHub. <https://github.co.jp/>, (2020-07-10 参照) バージョン管理やプログラムのソースコード共有のためのサービス。

(おおたに・しゅうへい)

琉球大学附属図書館