

# 業務効率化！プログラミングの裏技

尺一麟 著

## 裏技 1

# キーボードショートカットをマスターして IDE の 操作を高速化

プログラミングにおいて、キーボードショートカットの活用は極めて重要です。適切にキーボードショートカットを使いこなすことで、絶えずマウスを移動したり、メニューをクリックする手間を省くことができます。例えば、コードの行を簡単に上下に移動したり、特定のコードブロックを選択したりするのに役立ちます。

IDE（Integrated Development Environment）を効果的に操作するためには、適切なキーボードショートカットを覚えておくことが不可欠

です。IDE が提供する多くの機能は、特定のキーの組み合わせでスムーズにアクセスできるようになっています。これにより、作業効率が飛躍的に向上し、開発プロセスがスムーズに進行します。

キーボードショートカットをマスターすることは、プログラミング作業において大きなメリットをもたらします。IDE の操作を高速化し、生産性を飛躍的に向上させるために、キーボードショートカットの活用方法を習得してみてください。

## 裏技 2

**スニペットツールを活用して頻繁に使うコードを  
瞬時に挿入**

スニペットツールを上手に活用することで、同じコードを繰り返し入力する手間を大幅に削減できます。頻繁に使用するコードをスニペットツールに登録しておくことで、作業効率を格段に向上させることができます。例えば、テンプレート、汎用的な関数、またはよく使用されるコマンドなどをスニペットとして登録しておくことで、瞬時にコードを挿入することが可能となります。これによって、作業中のストレスも軽減され、より集中して効率的に業務を遂行することができるでしょう。是非、スニペットツールを駆使して、プログラミング作業の効率化を図ってみてください。

## 裏技 3

# バージョン管理システムのブランチ機能を活用してタスクごとに作業を分割

バージョン管理システムのブランチ機能を活用することは、業務効率化において非常に重要です。ブランチを使うことで、開発プロジェクトや作業をタスクごとに分割し、個々の作業を独立して進めることができます。これにより、複数人で同時に作業を行う際にも互いの作業が干渉することなく、効率的に進捗を管理することができます。また、各タスクごとにブランチを切り分けることで、修正や変更の履歴を明確に管理することができ、コードの品質やセキュリティを向上させることができます。ブランチ機能をしっかり活用することで、チーム全体の作業の進捗

管理やコミュニケーションの円滑化を図り、業務効率化を実現することができるよう。

## 裏技 4

# デバッグプリント文よりもインタラクティブデバッガを使用

プログラミングにおいて、デバッグは非常に重要な作業です。多くの開発者は、デバッグプリント文を使用して変数の値を確認しながらデバッグを行っていますが、実はそれよりも効率的な方法があります。それが、インタラクティブデバッガを使用することです。

インタラクティブデバッガを使用すると、コードの実行中に変数の値をリアルタイムで確認しながら逐一ステップ実行していくことができます。これにより、デバッグプリント文を挿入して何度もコードを編集・再実行する手間が省けるだけでなく、バグの特定や修正がより迅速に行えるようになります。

さらに、インタラクティブデバッガを使うことで、コードの実行中に条件式や変数の値を動的に変更することもできます。この機能を駆使すれば、複雑な条件分岐やループの中での挙動をリアルタイムに確認しながら、効果的なデバッグを行うことが可能です。

デバッグプリント文は便利なツールではありますが、インタラクティブデバッガを取り入れることで、より効率的にデバッグ作業を行うことが

できます。プログラミングの裏技として、是非インタラクティブデバッガの活用を検討してみてください。

## 裏技 5

### 正規表現を駆使してテキスト処理を高速化

プログラミングにおいて、正規表現はテキスト処理を効率化するための強力なツールです。正規表現を駆使することにより、複雑なパターンや条件に基づいてテキストを検索、置換、抽出する作業が劇的に簡素化されます。例えば、特定の文字列やパターンを一括で置換する際には、正規表現を使うことで手作業での作業量を大幅に削減することができます。



正規表現をマスターすることで、テキスト処理の手法に変革をもたら  
し、業務の効率化を実現できるでしょう。そのため、時間や労力を削減  
し、作業のミスを防ぐためにも、正規表現を使ったテキスト処理は必須  
のスキルと言えるでしょう。是非、正規表現を学び、業務効率化のため  
に活用してみてください。

## 裏技 6

### タスクランナーやビルドツールを使用して繰り返し作業を自動化

タスクランナーやビルドツールを使用することで、煩雑な繰り返し作業

を自動化することができます。これらのツールを活用することで、同じ手順を繰り返し行う手間を省くことができます。たとえば、ファイルの結合や圧縮、テストの実行など、頻繁に行う作業をツールに任せることで、作業効率を大幅に向上させることができます。

さらに、タスクランナーやビルドツールを使用することで、作業内容を細かく設定して効率的に処理することが可能です。例えば、特定の条件下でのみ特定のタスクを実行するように設定することで、より柔軟な自動化処理を行うことができます。

このように、タスクランナーやビルドツールを活用することで、作業効率を向上させ、生産性を高めることができます。煩雑な繰り返し作業にかかる時間を削減し、より重要な業務に集中するために、ぜひこれらの

ツールを導入してみてください。

## 裏技 7

### API テストには Postman や Swagger を活用

API テストを行う際には、Postman や Swagger のようなツールを活用することで、業務効率を大幅に向上させることができます。これらのツールを使うことで、API のリクエストやレスポンスを容易に確認したり、自動化されたテストを行ったりすることが可能となります。

Postman は直感的な UI と豊富な機能で、API のテストや開発を効率よく行うことができます。一方、Swagger は API のドキュメント作成やモックサーバーの構築を手助けしてくれるため、開発プロセスをスムー

ズに進めることができます。これらのツールを活用することで、プログラミングの裏技を駆使して、より効率よく API テストを行うことができるでしょう。

## 裏技 8

### コードレビューを自動化するために静的解析ツールを導入

コードレビューを自動化するために、静的解析ツールの導入は業務効率化の重要なステップの一つと言えます。静的解析ツールはコードの品質や潜在的なバグを素早く発見し、開発者が作業時間をムダにすることなく、修正する際の手助けをしてくれます。大規模なコードベースにおい

ては、人間だけでは見逃してしまう可能性のある問題を的確に指摘してくれる静的解析ツールの導入は、開発の生産性向上や品質向上につながることでしょう。

## 裏技 9

### コマンドラインツールをカスタマイズして日常的な作業を自動化

コマンドラインツールを使って日常的なタスクを自動化することは、業務効率を向上させるための重要なスキルです。カスタマイズされたスクリプトを実行することで、繰り返し行う作業を簡素化し、時間と労力を節約することができます。このようなプログラミングの裏技を習得する

ことで、業務効率化を実現し、より効果的な作業環境を構築することが可能です。

## 裏技 10

### Docker を用いて開発環境をコード化し、チーム 間で共有

Docker を使用することで、開発環境をコード化することが可能です。

これにより、開発環境の構築が簡素化され、チーム全体で共有することが容易になります。各メンバーが同じ環境を持つことで、開発プロセスが一貫性を持ち、コミュニケーションもスムーズになります。さらに、Docker を使うことで、開発環境のセットアップが大幅に効率化され、

新しいメンバーがプロジェクトに参加する際にもスムーズに導入することが  
とができます。業務効率化の観点からも、Docker を活用した開発環境  
のコード化は非常に有益です。

## 裏技 11

# REST API を活用して外部サービスとの連携を自 動化

REST API を活用して外部サービスとの連携を自動化することは、業務  
効率化を図る上で有益な裏技と言えます。REST API を使用すること  
で、外部サービスとのデータのやり取りを自動化し、手作業で行ってい  
た作業を劇的に短縮することが可能です。

例えば、社内のデータベースと外部サービスのデータを連携させる際、REST API を活用すれば、毎日定時にデータを同期するよう設定することができます。これにより、定型的な作業を自動化することで、社内の業務効率を向上させることができます。

さらに、REST API を使用することで、異なるアプリケーション間でデータの受け渡しを効率化することができます。例えば、顧客情報を社内の CRM システムから営業支援ツールに自動的に転送することにより、営業担当者はより効果的に顧客とのコミュニケーションを行うことができますでしょう。

つまり、REST API を活用して外部サービスとの連携を自動化すること



は、業務プロセスを効率化し、生産性向上に貢献する素晴らしい裏技なのです。

## 裏技 12

# テンプレートエンジンを使用して HTML の生成を 効率化

テンプレートエンジンを使用することで、HTML コーディングの効率が飛躍的に向上します。テンプレートエンジンは、HTML の構造を事前に設計し、動的にコンテンツを差し込むことができます。これにより、同じ構造のページを何度も繰り返し作成する際に、手作業での作業が大幅に削減されます。

また、テンプレートエンジンを使用することで、デザイナーとデベロッパー間のコミュニケーションもスムーズになります。デザイナーが作成したデザインをテンプレートに落とし込むだけで、デベロッパーはそれを元にコーディングを行うことができます。このように、各専門家の役割を明確に分担することで、業務効率が格段に向上することが期待できます。

さらに、テンプレートエンジンを使用することで、コードの再利用性も高まります。共通の部分をテンプレートとして切り出し、適宜差し込むことで、同じようなページを作成する際に大幅な手間を省くことができます。これにより、プロジェクト全体の効率化が図れます。

全ての要素を考慮すると、テンプレートエンジンを使用して HTML の生成を効率化することは、プログラミングにおける裏技として大変有用です。業務の効率化を図り、プロジェクトの成功に貢献するために、積極的に取り入れていくことが重要です。

## 裏技 13

### メールやチャット通知の自動化でプロジェクトの 状態をチームに報告

プロジェクトの進捗状況を効果的にチームに共有するために、メールやチャット通知の自動化が大いに役立ちます。特定のイベントやタイミングで通知が送られることで、チーム全体がプロジェクトの現在の状況を

把握しやすくなります。さらに、これらの自動化された通知は、チームメンバーがリアルタイムでアップデートを受け取ることを可能にし、コミュニケーションや調整がスムーズに進むでしょう。業務効率を高めるために、プログラミングの裏技を駆使してメールやチャット通知の自動化を取り入れてみてはいかがでしょうか。

## 裏技 14

### データベースマイグレーションツールを活用して

### スキーマ変更を管理

データベースマイグレーションツールは、開発プロセスにおいてスキーマ変更を管理する上で非常に効果的なツールです。これらのツールを活

用することで、データベースのスキーマ変更を追跡し、バージョン管理することが可能となります。

データベースマイグレーションツールを使うことで、スキーマ変更が発生した際に手動での変更作業を行う必要がなくなります。また、変更履歴や適用状況を追跡することで、全体のデータベース構造を把握しやすくなります。

さらに、マイグレーションファイルを自動で生成してくれるツールも多く存在し、手作業での作成作業が不要となるため、開発効率の向上が期待できます。データベースマイグレーションツールを活用して、スキーマ変更をスムーズかつ効率的に行い、開発プロセス全体の効率化に貢献

しましょう。

## 裏技 15

# Git フックを利用してコミット前のコード検証を 自動化

Git フックを利用してコミット前のコード検証を自動化

プログラミングにおいて、コードの品質を維持することは非常に重要です。Git フックを利用することで、コミット前に自動的にコードの検証を行うことができます。例えば、コードスタイルの統一や構文エラーのチェック、テストの実行などを自動化することで、開発効率を高めるこ

とができます。

Git フックは、特定のイベント（例えば、コミット前やプッシュ時）が発生した際に実行されるスクリプトです。これを利用することで、開発チーム全体が一貫したルールに従ってコードを書くことができます。たとえば、コミット前にコードの静的解析ツールを実行することで、バグやセキュリティホールを事前に検知することが可能です。

さらに、テストの自動実行も Git フックを利用して行うことができます。コミット前に自動的にテストスイートを実行することで、問題を早期に発見しやすくし、リリース時の品質を向上させることができます。

Git フックを利用してコミット前のコード検証を自動化することで、開

開発プロセスを効率化し、品質を向上させることができます。是非、チームで導入してみてください。

## 裏技 16

### モニタリングスクリプトを作成してシステムの異常を早期に検知

プログラミングにおいて、システムの適正な稼働状態を維持するためには、異常を早期に検知し対処することが極めて重要です。そのためには、モニタリングスクリプトを作成することが有効です。このスクリプトは、システムの動作を定期的に監視し、予め設定した条件に合致する異常が発生した際に適切なアラートを発信します。これにより、システ



ムの問題を素早く把握し、迅速な対応を行うことが可能となります。モニタリングスクリプトを導入することで、業務効率が向上し、システムの安定性を確保することができます。

## 裏技 17

### マルチスレッドや非同期処理を使用して処理速度を向上

プログラミングにおいて、業務効率を向上させるためにはマルチスレッドや非同期処理を活用することが重要です。これらの技術を駆使することで、処理速度を飛躍的に向上させることが可能です。例えば、複数のスレッドを同時に実行することで、複数の処理を効率的に並列で行う

ことができます。これにより、待ち時間や処理のブロックを最小限に抑え、作業の効率化を図ることができます。さらに、非同期処理を利用することで、処理の待ち時間を有効活用し、リソースを最適化することができます。このようにマルチスレッドや非同期処理を駆使することで、業務プロセスの効率化を図り、より効果的なプログラムを開発することが可能となります。

## 裏技 18

# CI/CD パイプラインを構築してデプロイプロセスを自動化

CI/CD パイプラインを構築してデプロイプロセスを自動化

プログラミングの裏技を使って、業務効率化を図る上で CI/CD パイプラインの構築は非常に重要です。CI（Continuous Integration）と CD（Continuous Deployment/Delivery）は、コードの統合からデプロイメントまでの自動化を可能にする手法です。これにより、開発チームは継続的かつ効率的にソフトウェアを開発し、デプロイすることができます。

CI/CD パイプラインを構築する際には、様々な裏技が存在します。例えば、テストやビルド、デプロイなどのステップを自動化するためにスクリプトを活用することで、作業の簡素化と確実性の向上が図れます。さらに、ツールやサービスを組み合わせることで効率的なパイプラインを構築することも重要です。

また、CI/CD パイプラインを使ったデプロイプロセスの自動化により、人為的なヒューマンエラーを減らし、品質向上にもつながります。素早く安全にソフトウェアをリリースするためには、CI/CD パイプラインの効果的な活用が欠かせません。業務効率化や開発スピード向上を目指す際には、是非 CI/CD パイプラインの構築を検討してみてください。

## 裏技 19

### クラウドストレージを利用してファイルの共有と

### バックアップを自動化

クラウドストレージを利用することで、業務効率化を図る方法の一つと

して、ファイルの共有とバックアップを自動化することが可能です。クラウドストレージを活用することで、ファイルを手動で送信する手間やバックアップを取る手間を省くことができます。また、クラウド上に保存されたファイルは、どこからでもアクセス可能であるため、柔軟かつ効率的な業務が実現できます。さらに、共有フォルダを設定することで、チーム内でのファイルの共有や同時編集がスムーズに行えるため、コミュニケーションや意思疎通の向上にもつながります。このように、クラウドストレージを利用してファイルの共有とバックアップを自動化することは、業務効率を飛躍的に向上させる裏技の一つと言えるでしょう。

## 裏技 20

# テスト駆動開発（TDD）を実践してバグを減らし、開発速度を向上

ソフトウェア開発において、テスト駆動開発（TDD）は非常に効果的な手法である。TDD を実践することで、コードの品質が向上し、バグの発生確率を減らすことができる。また、テストコードを最初を書くことで、開発者はコーディングの目的や要件を明確に把握し、設計を改善することができる。これにより、開発速度が向上し、開発プロセス全体の効率が格段に向上する。したがって、TDD を取り入れることで、業務効率化と品質向上が同時に実現できると言えるだろう。

## 裏技 21

# カンバンやスクラムといったアジャイル手法を採用してタスク管理を効率化

カンバンやスクラムといったアジャイル手法を採用することで、チームはタスク管理を効率化することができます。アジャイル手法の柔軟性と可視性によって、チームはプロジェクト全体を把握しやすくなり、優先順位を明確にし、タスクの進捗状況をリアルタイムで把握することが可能となります。さらに、各メンバーは自己管理を促進するため、責任感やチームワークが向上し、業務効率も向上します。アジャイル手法を取り入れることで、プロジェクトの成果物を迅速に提供するだけでなく、

高品質で効果的な成果を生むことができます。

## 裏技 22

# Web スクレイピングを活用してデータ収集を自動化

Web スクレイピングを活用してデータ収集を自動化することは、業務効率化において非常に有用な手法です。Web スクレイピングは、Web ページから必要な情報を抽出するプログラミング技術の一つであり、人による情報収集作業を大幅に短縮し、効率化を図ることができます。

例えば、競合他社の価格情報や商品情報を定期的に収集し、自社のデー



データベースに反映させる場合、Web スクレイピングを活用することでリアルタイムな情報収集が可能となります。また、市場動向や特定キーワードの情報収集など、様々な分野で活用することができます。

Web スクレイピングを行う際には、robots.txt を尊重し、サーバーへの負荷をかけないような配慮が必要です。また、データの使用方法に関する法的な規制や利用規約にも十分な注意を払い、合法かつ倫理的な方法でデータを収集することが重要です。

業務効率化を図るためには、Web スクレイピングの技術を使いこなし、的確にデータを収集・処理することが重要です。データを正確かつ効率的に取得できれば、意思決定や戦略立案においても大きなメリットをもたらすことでしょう。Web スクレイピングをマスターし、ビジネス

スに活かすことで、業務効率化の道はさらに開けることでしょう。

## 裏技 23

### シェルスクリプトを駆使してルーチン作業を一括

#### 実行

プログラミングにおける裏技の一つとして、シェルスクリプトを駆使してルーチン作業を一括実行する方法があります。シェルスクリプトは、コマンドラインで実行できるスクリプト言語であり、Linux や Mac などの UNIX 系システムで利用されています。例えば、フォルダ内の特定のファイルを自動でリネームしたり、ファイルのバックアップを自動で作成したりする際に便利です。

シェルスクリプトを利用することで、同じ作業を繰り返し行う手間を省くことができます。定型的な作業を一括実行することで業務効率を向上させることができるため、プログラマーやシステム管理者など、さまざまな分野で広く活用されています。是非、シェルスクリプトを使って業務効率化を図る裏技を取り入れてみてください。

## 裏技 24

### バッチファイルや Cron ジョブを設定して定時タスクを自動実行

バッチファイルや Cron ジョブを設定すれば、定時タスクを自動実行す

ることが可能です。これにより、業務効率を向上させることができます。例えば、定期的に行うデータバックアップやレポートの作成、データベースの更新などを自動化することで、作業時間を大幅に節約することができます。さらに、ヒューマンエラーを減らし、作業の正確性を確保することもできます。定時タスクを自動化することで、作業のミスや忘れ物を防ぎ、生産性を向上させるため、プログラミングの裏技であるバッチファイルや Cron ジョブの活用は非常に重要です。

## 裏技 25

**クラウド関数（AWS Lambda など）を利用して**

**サーバーレスアーキテクチャを採用**

近年、多くの企業が業務効率化とコスト削減のために、クラウド関数

（AWS Lambda など）を利用してサーバーレスアーキテクチャを採用

する動きが加速しています。サーバーレスアーキテクチャを導入するこ

とで、従来のサーバー管理や運用にかかる手間やコストを大幅に削減す

ることが可能となります。

クラウド関数を使用することで、サーバーのプロビジョニングやスケー

リングにかかる手間を省略し、開発者は自身のアプリケーションのコー

ドに集中することができます。また、クラウド関数は必要なときにのみ

実行されるため、コスト効率も向上します。

さらに、クラウド関数を使用することで、自動化されたタスクや定期的

な処理を効率的に実行することが可能となります。例えば、定期的なデータの処理やバッチ処理、イベント駆動型の処理など、様々な業務を自動化するための機能を容易に実装することができます。

そのため、クラウド関数を利用してサーバーレスアーキテクチャを採用することで、企業は業務の効率化やコスト削減だけでなく、柔軟性や拡張性の向上も実現することができます。今後、ますます多くの企業がクラウド関数を活用してサーバーレスアーキテクチャを採用することが予想され、その効果に期待が寄せられています。

## 裏技 26

# リモート API を呼び出してアプリケーションの機能を拡張

プログラミングにおいて、業務効率を向上させる裏技の 1 つとして、リモート API を呼び出してアプリケーションの機能を拡張する方法があります。この方法を利用することで、既存のアプリケーションに新機能を追加する際に、自前で機能を開発する手間を省くことができます。

リモート API を利用すると、他のサービスやプラットフォームが提供する豊富な機能を容易に組み込むことができます。例えば、地図サービスの API を利用して位置情報を取得したり、決済サービスの API を利用

してオンライン決済機能を追加したりすることが可能です。

さらに、リモート API を利用することで、自社開発チームの負担を軽減

し、素早く新機能をリリースすることができます。加えて、既存のサー

ビスやプラットフォームが提供する API は、セキュリティや品質に対す

る保証が高いため、安心して利用することができます。

したがって、リモート API を呼び出してアプリケーションの機能を拡張

することは、効率的かつ効果的な方法であり、プログラミングにおける

重要な裏技の 1 つと言えるでしょう。



## 裏技 27

# データ可視化ツール（Tableau、Power BI）を利用してレポート作成を自動化

データ可視化ツールを使ってレポート作成を自動化することは、業務効率化において非常に重要です。Tableau や Power BI などのツールを活用することで、データを視覚的かつ分かりやすく表現することができます。このことは、レポート作成にかかる時間を大幅に短縮し、より効率的な業務運営が可能になります。

これらのツールを使ってレポートを自動化する際には、データソースの接続やデータの整形、グラフやチャートの設定などを自動化することが

重要です。さらに、スケジュールを設定して定期的にレポートを更新するようにすることで、常に最新の情報を提供することができます。

データ可視化ツールを活用することで、従来の手作業に比べて作業効率が向上し、さらに正確性や品質も向上します。これにより、時間と労力を節約しつつ、よりの確な意思決定が可能となります。データ可視化ツールを活用してレポート作成を自動化することは、プログラミングの裏技としても非常に役立つ手法であり、積極的に活用していくことが重要です。

## 裏技 28

# メモリキャッシュ（Redis、Memcached）を活用してデータアクセス速度を向上

データベースから取得するデータの読み込み時間を短縮し、システムの応答速度を向上させるためには、メモリキャッシュ（Redis や Memcached）を活用することが重要です。これらのツールを使用することで、データの一時的な保存や高速なアクセスが可能となります。メモリキャッシュを使用することで、データベースへの負荷を減らし、アプリケーション全体のパフォーマンスを改善することができます。メモリキャッシュの導入は、システム全体の効率化につながり、利用者の満

足度向上にもつながるでしょう。

## 裏技 29

### オブジェクトストレージを利用して静的コンテン

### ツの配信を高速化

オブジェクトストレージを利用することで、静的コンテンツの配信を効率化し、サイトの表示速度を飛躍的に向上させることが可能です。オブジェクトストレージは、大容量のデータを安全に保存し、必要な時に素早くアクセスすることができるストレージ形態であり、そのメリットを最大限に活用することで、ユーザーの利便性を高めることができます。

従来の方法では、静的なコンテンツをサーバーに保存し、それを一度に大量のユーザーに配信する際に、サーバーの負荷がかかり、表示速度が遅くなることがありました。しかし、オブジェクトストレージを使用することで、コンテンツが複製・分散保存され、ユーザーが近いデータセンターからデータを取得できるため、高速な配信が可能となります。

さらに、オブジェクトストレージは冗長性を持ち、データのバックアップや復旧が容易であるため、安全性の向上にも寄与します。これにより、データが失われるリスクを軽減しつつ、メンテナンスや運用の効率化も図れます。

オブジェクトストレージを活用することで、静的コンテンツの配信を効率化し、サイトの表示速度を飛躍的に向上させるだけでなく、セキュリ

ディや運用の効率も向上させることができます。これにより、ユーザーエクスペリエンスの向上だけでなく、ビジネスの利益性にも寄与することが期待できます。

## 裏技 30

### コードジェネレータやボイラープレートコードを生成するツールを使用して開発を加速

プログラミングにおいて、コードの再利用や効率的な開発を行うためには、コードジェネレータやボイラープレートコードを生成するツールを上手に活用することが重要です。これらのツールを使うことで、同じようなコードを繰り返し書く手間や時間を省くことができます。さらに、

一貫性のあるコードを簡単に生成することができるため、開発の効率が格段に向上します。プロジェクトごとに独自のツールを構築することも可能ですが、市販のツールを使用することでさらなる効率化や品質の向上が期待できます。是非、コード生成ツールの活用を検討して、開発効率の向上を図ってください。

**業務効率化！プログラミングの裏技**

**著者 尺一麟**

**発行日 令和 6 年 3 月 26 日**

**Copyright © 2024 Rin Sakakuni All rights reserved.**