

Bot開発のタネがたっぷり!手を動かせば絶対できる!

DISCORD BOT 開発実践入門

著 cod

本文中のシステム名称や製品名などは、各社の登録商標または商標です。

筆者は本書の内容が正確であるよう努めました。が、筆者および関係者は本書の内容に基づく運用結果には一切の責任を負いません。ご了承ください。

はじめに

Discord Bot を作ってませんか？

対象読者

この本を読むとわかること

この本の構成

第 1 章 予備知識 - Discord と Python

1.1. ボイスチャットサービス Discord	12
1.1.1. Discord はゲーマーに大人気	
1.1.2. Discord の特徴を紹介	
1.2. プログラミング言語 Python	13
1.2.1. Python の歴史と特徴	
1.2.2. 強みは豊富なライブラリ	
1.2.3. いろいろな用途で人気の Python	
1.3. Bot 開発フレームワーク discord.py	14
1.3.1. Python で一番人気の多機能 API ラッパー	
1.3.2. 強力な拡張フレームワーク	

第 2 章 開発全体マップ - 作る Bot の全貌

2.1. そもそも、Bot ってなんだろう	15
2.1.1. 脳みそ「プログラム」と身体「Bot アカウント」で Bot が動く	
2.1.2. 自分のアカウントを Bot で操作してもいい？	
2.2. おたすけ Bot を作ろう	16
2.2.1. あなたのお手伝いをします	
2.2.2. あなただけのおたすけ Bot	
2.2.3. おたすけ Bot 開発時の注意点 - 技術を正しく使おう	
2.2.4. サンプルコードをまとめたリポジトリがあります	
2.3. 開発ステップを紹介	18
2.3.1. discord.py を理解するための 5 ステップ	
2.3.2. ステップ 1: 開発環境 - Bot 作りの土台作り	
2.3.3. ステップ 2: 準備編 - Bot の身体を手に入れる	
2.3.4. ステップ 3: 初級編 - Bot を動かしてみる	
2.3.5. ステップ 4: 中級編 - Bot とやりとりする	
2.3.6. ステップ 5: 上級編 - コード整理と高度な機能	

第3章 開発環境 - 開発の準備をしよう

3.1. Bot の開発環境について知ろう	21
3.1.1. 最小構成 - 最低限の動作環境	
3.1.2. PC 環境 - 現実的な最小構成	
3.1.3. サーバー環境 - 実用的な構成	
3.1.4. どの環境を使えばいいの？	
3.2. 環境を構築しよう	24
3.2.1. 基礎用語の紹介	
3.2.2. インストールするものの一覧	
3.2.3. 必要な PC 性能スペックは？	
3.2.4. うまくいかないときのチェックポイント	
3.2.5. Python と Git のインストール - Windows 編	
3.2.6. Python と Git のインストール - macOS 編	
3.2.7. Visual Studio Code のセットアップ	
3.3. 動作確認をしよう	31
3.3.1. ターミナルの起動	
3.3.2. Python コマンドの動作確認	
3.3.3. おつかれさまでした！	
📌 Bot 開発のタネ 開発者のコミュニティに参加しよう	

第4章 準備編 - Bot の準備をしよう

4.1. Bot アカウントを作成しよう	35
4.1.1. Application の作成	
4.1.2. Bot アカウントの作成	
4.1.3. トークンの取得	
4.2. テスト用 Guild を作ろう	39
4.2.1. なぜテスト用 Guild が必要なの？	
4.2.2. テスト用 Guild を作成する - Bot と対話する準備	
4.3. Discord の権限について知ろう	41
4.3.1. 権限は Guild の治安を守る仕組み	
4.3.2. Bot 開発に権限の検討はつきもの	
4.4. Bot を Guild に招待しよう	42
4.4.1. 招待用 URL の作成	
4.4.2. Bot を Guild に招待	
4.4.3. Bot を公開したいときは？	

第5章 初級編 - Python 仮想環境・イベント

5.1. Python 仮想環境を作ろう - venv	47
5.1.1. pip - Python のパッケージ管理ツール	
5.1.2. 仮想環境を使ってみよう	
5.2. discord.py をインストールしよう	53
5.2.1. 仮想環境にモジュールをインストールする	
5.2.2. 使用パッケージを設定ファイルに書き出す	
5.3. プログラミングの基礎用語	55
5.3.1. 関数	
5.3.2. クラスとインスタンス	
5.4. Bot にログインしてみよう	57
5.4.1. はじめての Bot プログラミング	
5.4.2. はじめての Bot 実行	
5.4.3. エラーが出たら実力アップのチャンス	
5.4.4. よくあるまちがい	
5.5. イベントについて知ろう	63
5.5.1. イベントは Bot に起きる出来事	
📌 Bot 開発のタネ イベントで何ができるか考える	

第6章 中級編 - コマンド・対話的処理・UI

6.1. トークンを隠そう	70
6.1.1. トークンは漏洩厳禁なので隠す	
6.1.2. config モジュールにトークンを隠す	
6.1.3. config モジュールを使う	
6.2. Git リポジトリを作ろう	73
6.2.1. Git ってなに？	
6.2.2. Git リポジトリの作成	
6.2.3. Git に入れたくないファイルがあるときは？	
6.2.4. Git に変更をコミットする	
6.2.5. コミットの追加	
6.3. Bot にコマンドを追加しよう - Commands	79
6.3.1. コマンドは Bot への命令	
6.3.2. はじめてのコマンド実装	

6.4. Bot と対話的にやりとりしよう..... 83

- 6.4.1. 対話的なやりとりの動作例
- 6.4.2. 対話的なやりとりを実装する
- 6.4.3. 対話の打ち切り - タイムアウト処理を入れる
- 📌 Bot 開発のタネ イベントを待って何かしてみよう

6.5. メッセージを見やすく表現しよう - Embed..... 89

- 6.5.1. Bot にしかできない Embed 形式の表示
- 6.5.2. ただのメッセージを Embed に書き換える
- 6.5.3. Embed の属性紹介

第7章 上級編 - Cog・パラメーター・定期処理

7.1. Bot の機能をファイルに分割する - Cog..... 94

- 7.1.1. コード整理のためのしくみ - Cog
- 7.1.2. 動的なモジュール読み込み - Extension
- 7.1.3. あいさつコマンドを Cog に切り出す
- 📌 Bot 開発のタネ さまざまな Cog、Extension を参考にしてみよう

7.2. Cog について詳しく知ろう..... 102

- 7.2.1. Cog にイベントリスナーを定義する
- 7.2.2. Cog 内で発生したエラーの処理を定義する
- 7.2.3. Cog 単位の実行権限を付ける

7.3. Command をもっと便利に使おう..... 103

- 7.3.1. パラメーターとは
- 7.3.2. パラメーターを持つコマンドを実装する
- 7.3.3. Converter について知る
- 7.3.4. コマンドのエラー処理

7.4. Bot に定周期で処理をさせよう - Task..... 107

- 7.4.1. Task ってなに？
- 7.4.2. Task の動作確認
- 7.4.3. 時報機能の実装
- 7.4.4. Task について詳しく知る
- 📌 Bot 開発のタネ 定時処理をする Bot のアイデア

第 8 章 Bot 開発のタネ-便利な機能・コード集

8.1. この章のつかいかた	117
8.1.1. 重要：この文書はリファレンスではありません	
8.1.2. 応用のしかた	
8.2. Guild	120
8.2.1. 情報を表示する	
8.2.2. 名前を変える	
8.2.3. テキストチャンネルを作成する	
8.2.4. メンバーを Kick、Ban する	
8.2.5. Guild を退出する	
8.3. Member	125
8.3.1. 情報を表示する	
8.3.2. メンションで通知を送る	
8.3.3. メンバーが Bot かどうか確認する	
8.3.4. メンバーに DM を送信する	
8.3.5. ボイスチャットの参加状況を取得する	
8.3.6. メンバーを別のボイスチャンネルに移動する	
8.4. Client	132
8.4.1. Bot アカウントに「～をプレイ中」を表示する	
8.4.2. Bot を終了する (Bot アカウントからのログアウト)	
8.4.3. Bot アカウントの情報を表示する	
8.4.4. アプリケーションの情報を表示する	
8.5. TextChannel	136
8.5.1. 時間のかかる処理中、入力中の表示をする	
8.5.2. 条件にあうメッセージをまとめて削除する	
8.5.3. テキストチャンネル (Guild) への招待を送信する	
8.5.4. テキストチャンネルに画像を投稿する	
8.6. Message	141
8.6.1. メッセージにリアクションを追加する	
8.6.2. メッセージに添付されたファイルを保存する	

おわりに

あとがき

拙著「Discord Bot 運営実践入門」について

謝辞

はじめに

Discord Bot を作ってみませんか？

こんにちは、著者の cod (コッド) です。私は 2019 年 5 月から、日本語読み上げ Discord Bot 「shovel (シャベル)」の開発・運営をしています。「shovel」のユーザー数は 2020 年 9 月現在、約 50 万にもなりました。わずかの期間にすっかり大きくなった「shovel」ですが、最初はバグだらけの小さなプログラムを、たった 4 人の小さなコミュニティで使用するから始まったのです。Bot を開発・運営するのはとても楽しいことです。自分用に作るサンプルプログラムや、「Hello world」と表示するだけの画面と異なり、Bot 開発は、実際に Discord を使った遊びやゲーム生活をより便利で楽しいものにしてくれます。

この本は、Discord Bot 開発、とくにその実装 (コーディング) について書いた本です。2019 年 9 月に発行した技術書「Discord Bot 運営実践入門」は、Bot 開発・運営にまつわることについて網羅的に書きました。そこで次は「とにかく Bot を作ってみたい！」という方の需要に応えたいと考え、実装に絞った観点で執筆し、「ちょっと欲張りなチュートリアル」である本書ができあがりました。また、「Bot 開発のタネ」として、Bot を作るためのアイデアも散りばめています。

本書のチュートリアルはすべて、インターネットに繋がった PC さえあれば、すぐに実践できます。また本書で紹介する Bot のソースコードは著者 GitHub にて公開しています。その他にも挫折しないようたくさん工夫をしていますので、ぜひ取り組みやすい部分から気軽に挑戦してみてください。

2020 年 9 月 cod

対象読者

- Discord をもっと便利に使いたい人
- Discord の Bot を作ってみたい人
- 簡単なサンプル Bot を作って動かしたことはあるが、次に何をすべきか悩んでいる人
- Bot を作ってみたいが、作る Bot の機能が思いつかない人

この本を読むとわかること

- Python 仮想環境のメリットと使い方
- Git の基礎
- discord.py を使った Bot 開発の方法
- ユーザーの発言を拾って Bot に処理させる方法
- Bot に対話的処理をさせる方法
- Bot にコマンドを処理させる方法
- Bot 開発のアイデアの出し方
- discord.py のコードの整理方法
- Bot に定時処理をさせる方法
- 他にもいろいろ

この本の構成

はじめに、Discord および Discord Bot について簡単に解説します。また、プログラミング言語「Python」についても簡単に紹介します。次に、環境構築を実施します。Git やコードエディターなど、開発に必要なものをインストールします。つづいて、本書のメインコンテンツ、Bot の開発チュートリアルです。「初級編」「中級編」「上級編」の3ステップで、実用的な Bot を開発するための知識を、100 件をこえるコード例とともに提供します。

第 1 章 予備知識 - Discord と Python

この章では、まず Discord というサービスについて簡単に説明します。次に、本書で Bot を作る際に使用する Python (パイソン) と、Python で Discord Bot を作る際にもっともよく利用されているライブラリ「discord.py」について紹介します。

1.1. ボイスチャットサービス Discord

1.1.1. Discord はゲーマーに大人気

Discord (ディスコード、<https://discordapp.com/>) は、2015 年に開始したゲーマー向けのボイスチャットサービスです。2019 年 5 月の公式発表によると、ユーザー数は全世界で 2.5 億人。基本的な機能はすべて無料で利用できます。ゲーマーの間では、オンラインプレイでボイスチャットをするときには、まず選択肢としてあがるサービスです。ボイスチャットがメインのサービスではありますが、テキストチャットも利用できます。代替ツールとしては、Skype (スカイプ)、LINE (ライン) などが挙げられます。本書のチュートリアルを実践するには、Discord アカウントが必要です。もし Discord アカウントをお持ちでない場合、まずはアカウント作成を行ってから先に進んでください。

1.1.2. Discord の特徴を紹介

Discord では多彩なコミュニティを作ることが可能です。任意のメンバーを集めたクローズドなコミュニティから、コミュニティに参加するための招待用 URL を公開しているオープンなコミュニティまで、Discord にはさまざまなコミュニティがあります。Discord では、このコミュニティを指して「サーバー」と呼びます。ユーザーが「サーバー」を作成し、そこに他のユーザーを招待します。

Discord のコミュニティである「サーバー」という名前を本書で使うと、一般名詞の「サーバー」と混同してしまいます。そのため、本書では、Discord の開発者用公式ドキュメントに倣い、このコミュニティのことを以後「**Guild (ギルド)**」と呼ぶことにします。Discord ユーザーは、100 個までの Guild に参加でき

ます。それぞれの Guild にはいくつかの「チャンネル」があります。テキストチャットは「テキストチャンネル」、ボイスチャットは「ボイスチャンネル」で行います。

1.2. プログラミング言語 Python

1.2.1. Python の歴史と特徴

Python は 1991 年、グイド・ヴァンロッサムによって開発されたプログラミング言語です。オブジェクト指向の言語仕様を持ち、スクリプト言語のためコンパイルが不要で気軽に実行できます。特筆すべき点として、シンプルで覚えやすく、読みやすく書きやすい文法があげられます。インデントによってコードブロックの範囲を示す構文のため、コードの可読性が担保されやすいのです。ブラケット ({}) を使ってコードブロックを表現する言語 (C++ など) ではインデントが文法上の意味を持たないため、インデントは IDE (Integrated Development Environment) の機能やコーディング者の美意識に委ねられていますが、Python では構造のとおりインデントを入れないと正しく動かないためバグを排除しやすくなっています。また、できあがったコードは構造とインデントが一致し、美しく読みやすいものになります。

1.2.2. 強みは豊富なライブラリ

Python 言語のみで実現されている機能は多くありません。ですが、それはシンプルさを保つためにあえてそうしてあり、各種機能は標準ライブラリとして用意されています。必要に応じて標準ライブラリを選択し追加することで多様な機能を使うことができます。Python の機能拡張は標準ライブラリだけにとどまりません。強力な数値計算を効率的に行える NumPy や、データ解析を支援する pandas といった外部ライブラリが充実していることも特徴のひとつです。この本で利用する discord.py も外部ライブラリのひとつです。

1.2.3. いろんな用途で人気の Python

2020 年現在需要が高まり続けている機械学習に使われる開発言語として、

Python はファーストチョイスの言語となっています。また、Python はその文法的な特徴により初心者にも直感的にわかりやすいため、プログラミング初学者への教育用言語として人気です。外部ライブラリが豊富で、数値計算、画像処理、ウェブアプリケーションのフレームワーク、自然言語処理など、広範囲の開発分野をカバーしています。

1.3. Bot 開発フレームワーク discord.py

1.3.1. Python で一番人気の多機能 API ラッパー

discord.py は、Python で Discord Bot を作る際にもっともよく使われている API (Application Programming Interface) ラッパーです。特徴として、モダンな Python インターフェイス、レートリミット制御、Discord API 完全準拠、速度とメモリ効率の両立が挙げられます。GitHub (ギットハブ) でソースコードが公開されています。

1.3.2. 強力な拡張フレームワーク

discord.py には、基本機能に加えていくつかの拡張フレームワークがあります。拡張フレームワークを使うと、基本機能だけで実装するには手がかかる Bot を簡単に実装できます。

discord.ext.commands

`discord.ext.commands` は、コマンドで制御する Bot を実装するのに大変役立つフレームワークです。筆者が開発している shovel も、このフレームワークに沿って設計されています。

discord.ext.task

`discord.ext.task` は、v1.1.0 で追加された新しいフレームワークです。1時間ごとに時報を出すなど、定期的な処理をする Bot が簡単に実装できます。著者の Bot 「shovel」でもこのフレームワークを利用し、ログのダンプ処理や Bot の稼働状態を DB (DataBase) に出力する処理を実装しています。

第 2 章 開発全体マップ - 作る Bot の全貌

ここでは、Bot とは何か？ということ、この本でどのような Bot を作るのかということや、Bot を開発するうえでの注意事項を説明します。

2.1. そもそも、Bot ってなんだろう

2.1.1. 脳みそ「プログラム」と身体「Bot アカウント」で Bot が動く

Bot というのは、そもそも一体何なのでしょう。本書は「Python を使った Discord Bot プログラミングのチュートリアル」の本ですので、きっと「Python で作ったプログラム」が Bot を動かすのだらうということは想像できると思います。しかし、Bot を動かすには「Python のプログラム」、つまり脳みそだけでは不十分です。Discord にアクセスし、ユーザーの前に登場するための身体、つまり「Bot アカウント」が必要です（図 2.1）。

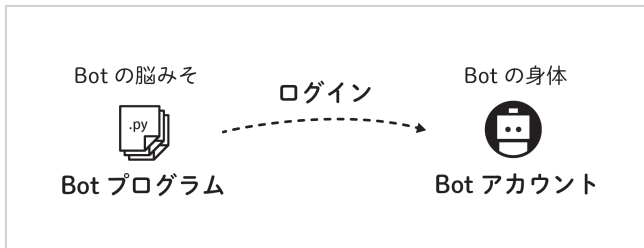


図 2.1: Bot に必要な「脳みそ」と「身体」

2.1.2. 自分のアカウントを Bot で操作してもいい？

普段自分が使ってる Discord アカウントを自動で動かすことはできないのかと疑問に思うかもしれません。これは **self-bot** といわれる行為で、技術的には可能であり、手順を紹介する記事なども存在しますが、Discord の公式ヘルプセンターで**明確に禁止**である旨が告げられています。絶対行わないようにしましょう。

2.2. おたすけ Bot を作ろう

2.2.1. あなたのお手伝いをします



図 2.2: おたすけ Bot

これから作成するのは、「おたすけ Bot (図 2.2)」。

これはあなたのお手伝いをしてくれる Bot です。本書「上級編」までを終わらせると、以下のようなことができるようになります。もちろん、これはほんの一例です。学んだ内容をもとに好きな機能を追加できます。

- 誰かが Guild に新しく参加したら、あなたのかわりに挨拶してくれる
- 誰かが NG ワードを発言したら、警告を出す
- 毎日、9 時から 21 時まで、時報を出してくれる

2.2.2. あなただけのおたすけ Bot

便宜上この本では前述の Bot を「おたすけ Bot」と呼びますが、あなたがこの Bot のことをなんと呼ぶかは自由です。好きな名前とアイコンを設定し、好きな口調で喋らせてあげましょう。

また、開発の手順をお話する合間に、「Bot 開発のタネ」として、Bot を開発するアイデアの出し方、考え方についてもお話します。これを読んで「おたすけ Bot」の機能を拡張するアイデアを思いついたら、いったん先に進む手をとめて、ぜひ拡張してみましょう。

2.2.3. おたすけ Bot 開発時の注意点 - 技術を正しく使おう

Bot 開発は楽しいものですが、注意すべきこともあります。ここに挙げたことはほんの一部で、これらがすべてではありません。

Discord の規約、Discord API の規約を守る

Discord はもちろん、Discord API にも利用規約があります。プラットフォームである Discord、Discord API の規約を理解し、守るようにしましょう。ユーザーデータに関する条項以外にも多くの内容がありますので、Bot を作るなら必ず確認しておきましょう。

<https://discordapp.com/developers/docs/legal>

ユーザーの情報の扱いに注意する

Bot の機能に関係ない情報を閲覧・保存することは、Discord API で禁じられているだけでなく、Bot 運営者としてのモラルに反します。ユーザーの同意なく会話ログを保存するなど、興味本位で Bot 利用者の信頼を裏切るような不要な処理を組み込むことがないようにしましょう。

他人の権利を侵害しない

著作権や肖像権など、他人の権利を侵害ないようにしましょう。アイコンに設定する写真やイラストは、自作のもの、作者に許可を得たもの、アイコンとしての使用が許可されているもののいずれかを設定しましょう。

2.2.4. サンプルコードをまとめたリポジトリがあります

本書のチュートリアルコードは、下記の GitHub リポジトリですべて公開しています。詰まった際にはぜひこちらも参考にしてください。

https://github.com/cod-sushi/discord_dev_sample

2.3. 開発ステップを紹介

2.3.1. discord.py を理解するための 5 ステップ

この本では、「開発環境」「準備編」「初級編」「中級編」「上級編」の順番で、「おたすけ Bot」をどんどん成長させていきます (図 2.3)。

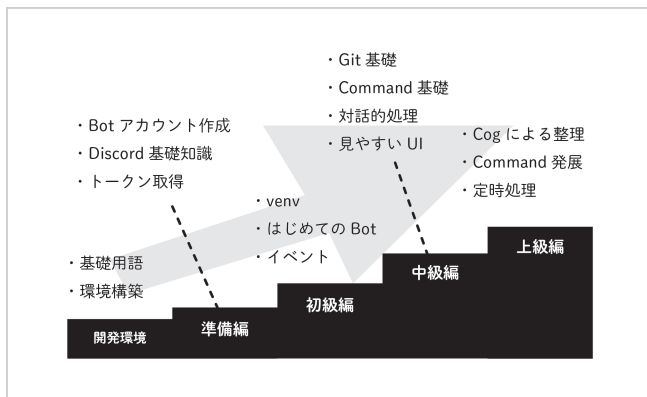


図 2.3: 本書の内容

ちなみに、「初級編」以降の手順は筆者が実際に Bot を開発するときのものと同じです。まずは確実に動く小さなプログラムを作成し、そこから必要な機能をすこしずつ付け加えます。同時にコードを整理整頓し、のちのちの機能拡張に備えます。この繰り返しで、徐々に複雑な機能を持つ Bot ができあがります。

2.3.2. ステップ 1：開発環境 - Bot 作りの土台作り

第 3 章では、どのような環境で Bot を動かすのかを解説します。また、Windows (ウィンドウズ)、macOS (マックオーエス) それぞれの OS (Operating System) で開発環境をととのえる方法についても解説します。

この章で学ぶこと

- Bot 開発に必要なもの
- 開発における基礎用語

- Windows、macOS それぞれの環境構築手順

2.3.3. ステップ 2：準備編 - Bot の身体を手に入れる

第 4 章では、Bot の身体である **Bot アカウント**を作成します。Discord における **権限**や、Bot を **Guild に招待**する方法についても解説します。

この章で学ぶこと

- Discord Bot アカウントの作成方法
- テスト用 Guild の必要性と作り方
- Discord の権限についての基礎
- Bot のトークンについて、トークンの取得方法
- Bot を Guild に招待する方法

2.3.4. ステップ 3：初級編 - Bot を動かしてみる

まず第 5 章では、venv (Virtual Environment) による**仮想環境**のメリットと使い方を学びます。仮想環境の上で、単一の Python ファイルだけで動作する、ほんとうに**シンプルな Bot**を作成します。まずは手っ取り早く Bot を動かしてみて、「動いた！」という気持ちを味わいましょう。とはいえ、ただ「Hello world」と表示するような Bot ではちょっと味気ないので、Guild に入ってくれた人に**挨拶**をしたり、**NG ワード**に警告を出したりしてみましよう。

この章で学ぶこと

- Discord Bot 開発のはじめかた
- Python 仮想環境のメリットと使い方
- Discord Bot のイベントについて

2.3.5. ステップ 4：中級編 - Bot とやりとりする

次に、第 6 章では「おたすけ Bot」に**あいさつ機能**を追加します。あいさつ機能では **Command (コマンド)** の初歩とユーザーと**対話的**にやりとりする方法について学びます。また、**Embed (エンベッド)** を使って Bot からのメッセージを見やすく表示してみます。

本章では、**Git (ギット)** でのバージョン管理の初歩についても簡単に触れます。Git については奥が深いので本書で詳細は述べませんが、まずは例のままに手を動かしてみてください。

この章で学ぶこと

- トークンを安全に管理する方法
- Git の基礎
- Bot をコマンドで操作できるようにする方法
- 対話的にユーザーとやりとりする方法
- 見やすいメッセージの出し方

2.3.6. ステップ 5：上級編 - コード整理と高度な機能

第 7 章では、はじめに discord.py の **Cog (コグ)** という仕組みを使って少しコードを整頓しましょう。コードを整理することのメリットについてもここで触れます。「おたすけ Bot」には、**時報機能**を追加します。時報機能の実装を通じて、**Task** という仕組みについて学びます。最後に、第 6 章で学んだコマンドについて、**パラメーター**の利用や**エラー処理**ができるよう、すこし知識を発展させましょう。

この章で学ぶこと

- Cog を使って Bot のコードを整頓する方法
- Bot に定時処理をさせる方法
- もっと便利にコマンドを使う方法
 - パラメーター受け取り
 - エラー処理