

はじめての

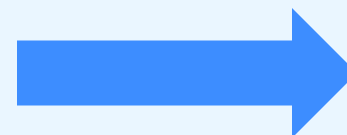
GitHub 入門

プログラムを「置く・見つける・育てる」
ための
最初の一歩

対象：初心者／ブログ・学習・個人開発にも

Git

履歴を残す仕組み



GitHub

共有・公開・協働する場所

GitHubは「作品の置き場」＋「変更履歴」

ファイルをただ保管するだけでなく、いつ・誰が・何を変えたかを追えます。

リポジトリ

プロジェクト用の箱
コード・画像・説明書
・履歴

Git

変更を記録する仕組み
「セーブポイント」の
連続

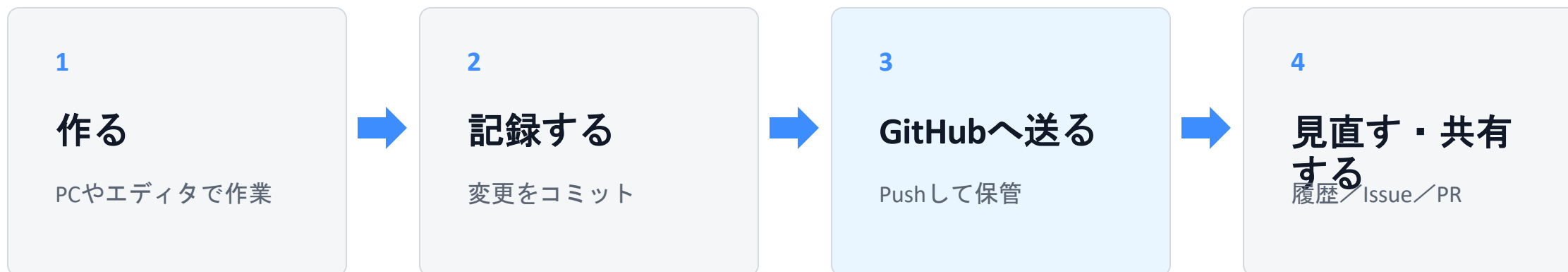
GitHub

Gitの保管・公開・
共同作業のサービス

- Git と GitHub は別物。ただし組み合わせて使うもの
- 最初はブラウザだけでもファイルを置ける

GitHubを使うと、作る→残す→直すがつながる

個人でも、将来の自分に引き継ぐために便利です。



「コミット」 = その時点の変更に名前を付けて残すこと

アカウント作成は、メール確認と2段階認証まで

最初のアカウントは個人用で構いません。無料プランから始められます。



github.com を開く

Sign up を選択



メール・パスワードを設定

メール認証を完了



2段階認証を設定

認証アプリかパスキーを推奨

ログイン後に 確認したいこと

- プロフィール
- メール認証
- 2段階認証

最初の置き場は「Private」で作れば安心

公開してよい内容か迷う間は、まず非公開リポジトリにします。

New repository



公開（Public）＝世界中の誰でも閲覧できる

プロジェクトは「読める形」で置くと、後で自分が助かる

コードだけでなく、説明書・設定例・ライセンスも一緒に管理できます。

```
my-project/  
├── README.md    # 何をするもの？  
├── src/         # プログラム本体  
├── docs/        # 補足資料  
├── .gitignore   # 入れない物の指定  
├── requirements.txt # 必要な部品  
└── LICENSE      # 利用条件
```

- README : 目的・使い方・動かし方
- src : プログラム本体をまとめる
- docs : 画像や説明資料など
- 秘密情報は置かない。後のスライドで解説

置き方は、最初はブラウザ／慣れたら GitHub Desktop

大きな作業や継続的な開発には、PC側のGit連携が便利です。

A. ブラウザでアップロード

- Add file → Upload files
- 小さな修正もWebで可能
- 初心者の最初の一步に

B. GitHub Desktop / Git

- PCのフォルダと同期
- 変更を見てコミットできる
- 複数ファイルの作業向き

MacでもWindowsでも、まずはAから始めて問題ありません。

参考プログラムは「目的＋言語＋人気度」で探す

検索して、README・更新日・ライセンスを確認してから使います。

検索例 `python websocket stars:>500 pushed:>2025-01-01`

README

何ができるか
導入方法は明快か

更新日

最近も保守
されているか

Stars / Issues

利用者の目安
未解決問題も見る

License

使ってよい条件
商用可否も確認

「見て学ぶ」と「コピーして使う」は別の話

公開コードでも、ライセンスと利用条件を必ず確認します。

学習として安全な使い方

- コードを読んで仕組みを理解
- 自分の小さな実験で動かす
- Forkして元の履歴を残す
- 変更点をメモする

注意が必要な使い方

- 無断で丸ごと自作として公開
- ライセンスを読まずに商用利用
- APIキー等が入ったまま共有
- 依存関係の安全性を未確認

プラグイン連携では、GitHubが「作業の共通ノート」になる

AIや開発ツールから、正しいプロジェクトの中身を参照・更新しやすくなります。



- 同じ最新版を見ながら相談できる
- 変更履歴が残るので、戻す・比較するがしやすい
- アクセス権を絞れば、必要な範囲だけ共有できる

最重要：公開リポジトリは「全世界に見える」

削除しても履歴や複製に残ることがあるため、最初から入れない運用が大切です。

絶対に置かないもの

- APIキー・パスワード・トークン
- 個人情報・顧客データ・住所録
- 秘密鍵（.pem / id_rsa など）
- 本番用の設定ファイル

守るための基本

- .gitignore で対象外にする
- Secrets 機能など安全な保管を使う
- 2段階認証を有効にする
- 公開前に「Files changed」を確認

もし誤って公開したら：すぐ無効化・再発行し、履歴を含めて対処する

今日のゴールは「1つの安全な置き場」を作ること

一度作れば、次からはプロジェクトごとに同じ型を使えます。

1

① アカウント作成

メール認証・2段階認証

2

② Privateで新規作成

README付きリポジトリ

3

③ 小さなファイルを置く

メモ／Python／ブログ素材

4

④ READMEを書く

目的・使い方・次の作業

公開・共有・プラグイン連携は、その後に少しずつ広げれば大丈夫です。