バージョン管理システム

Gitのメリット

- 古いバージョンから新しいバージョンのファイルまで管理できる
- 簡単に古いバージョンに戻せる
- チームでファイルや変更履歴をスムーズに共有できる
- 豊富な機能を使いチームでの共同開発を効率化できる

  ✓ リモートにあるファイルや変更履歴を自分のパソコンに取り込む

  ✓ 他の人の行った修正を手元のファイルに反映させる

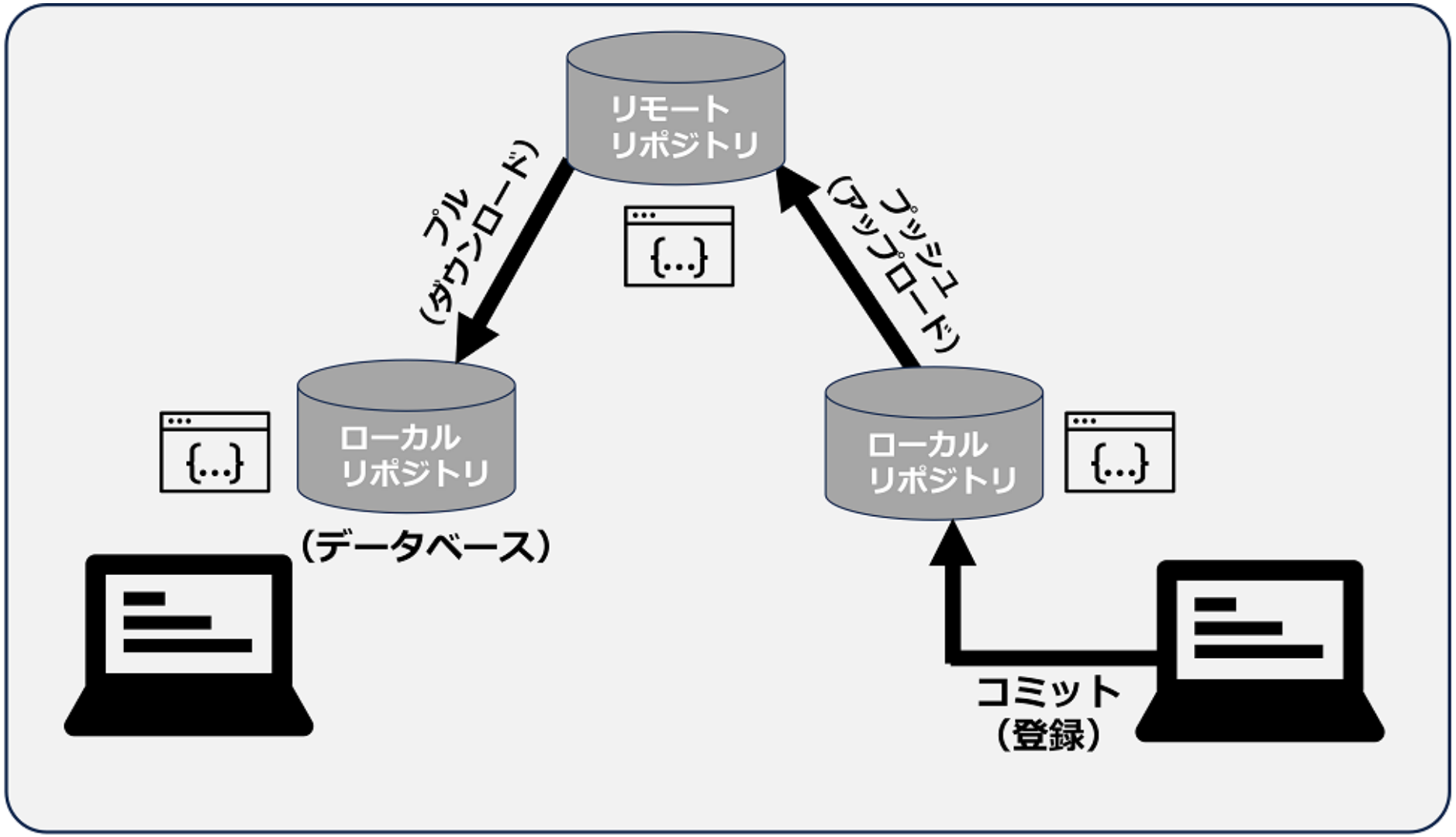
  ✓ バグ修正用や機能追加用などで、変更履歴を自由に枝分かれさせたり（ブランチを切る）、その枝分かれを統合したり（マージ）できる、など

Gitを使う際の注意点

- Git特有の概念や操作方法を習得する必要がある
- Gitを使用するチームメンバー全員がGitを理解する必要がある
- 混乱が起きないよう、運用ルールをきちんと決める必要がある

研修内で積極的に使用しながら慣れていきましょう！

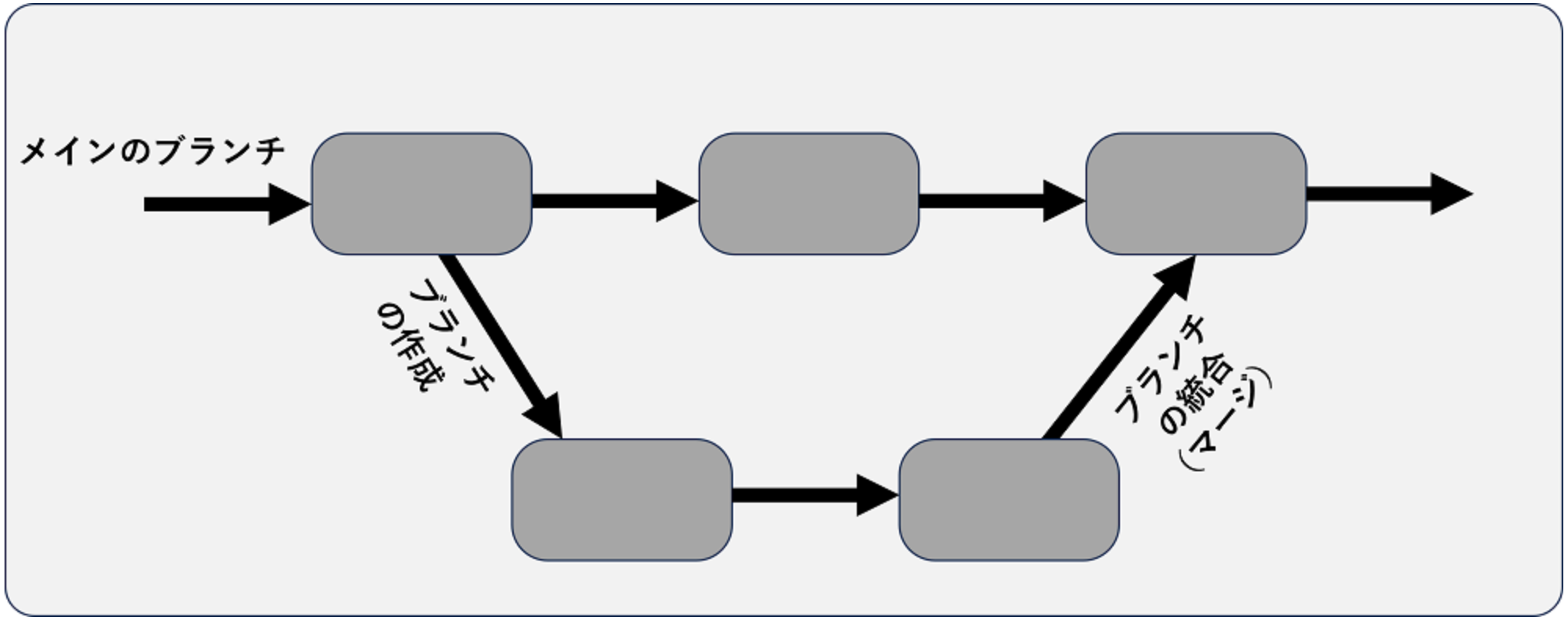
Gitの仕組み


リモート
リポジトリ
プル
(ダウンロード)
プッシュ
(アップロード)
{...}
ローカル
リポジトリ
{...}
(データベース)
ローカル
リポジトリ
{...}
コミット
(登録)

Gitで使用する用語 その1

用語名	概要
リポジトリ	ファイルや変更履歴を保存しておくデータベース。 ローカルにある「ローカルリポジトリ」と、ネットワーク上の「リモートリポジトリ」がある。 ユーザーがローカル環境でソースコードの編集などを行う際は、ローカルリポジトリを使う。 一方リモートリポジトリは、他ユーザーとファイル・変更履歴を共有する際に利用する。
コミット	ファイルや変更履歴をリポジトリへ登録すること
プッシュ	コミットを、ローカルリポジトリからリモートリポジトリへ反映させること。 プッシュを行うことで、ローカルリポジトリ上のファイルや変更履歴が、リモートリポジトリへアップロードされる。
プル	リモートリポジトリのコミットを、ローカルリポジトリへ反映させること。 プルを行うことで、リモートリポジトリ上のファイルや変更履歴が、ローカルリポジトリへダウンロードされる。
クローン	リモートリポジトリのコミットを、ローカルリポジトリへコピーすること。 プルとの違いは、プルでは差分のみコピーするのに対し、クローンでは丸ごとコピーする点にある。 そのため何も登録されていないローカルリポジトリへコピーしたいときは、クローンを使う。

ブランチの仕組み



Gitで使用する用語 その2

用語名	概要
ブランチ	同一リポジトリ内で変更履歴を分岐させること。 また、作成した分岐もブランチと呼ぶ。ブランチを作成した場合、ブランチ側で変更を反映させてもメインのブランチには影響しない。 バグ修正や機能追加などで、ソースコードのバージョン管理を分けて管理したいときに使う。 Githubではメインのブランチ名が「main」となっており、分岐するブランチ名は任意の文字列が使用できる。
マージ	作成したブランチをメインのブランチへ統合すること。 これによってブランチの変更内容が、メインのブランチに反映される。
フェッチ	リモートリポジトリの変更分を取得すること。 ローカルリポジトリのブランチに自動マージはされないため、必要に応じて手動でマージを行う。
コンフリクト	マージやプルを行った際に、異なるブランチで同じファイルに対して行った変更が競合してしまう現象のこと。 状況に応じてどの変更分を優先するかを判断し手動で競合を解消する必要がある。

Gitを使う準備

1. Gitのダウンロード＆インストール　※レンタルPCにはセッティング済み
2. GitのWebサービスの一種、Githubへのアカウント登録　※リスキルが準備したメールアドレスを使用
3. Githubの初期設定《Git Bash》
    ① ペアキー作成　※アルゴリズムED25519 ← 分かりやすい場所に保存(デスクトップ等)
    ② 公開鍵登録
    ③ リモートリポジトリの作成(または準備)
4. Gitの初期設定《Git Bash》
    ① ユーザ名とメールアドレスの設定
    ② ホスト認証鍵の取得
5. ローカルリポジトリの作成　※今回はVS Code上でリポジトリのクローンを作成