

バージョン管理基礎演習

第2回

基本操作編

①Gitユーザを設定

画面共有をしながら説明

【ユーザ名設定手順】

① 「Git Bash」を起動する。(※コマンドプロンプト等でも可)

② 以下のコマンドを入力実行する。

```
git config --global user.name "{任意のユーザ名}"
```

→ エラーが発生しなければOK

③ 以下のコマンドを入力実行する。

```
git config user.name
```

→ ②で設定したユーザ名が表示されればOK

【メールアドレス設定手順】

① 「Git Bash」を起動する。(※コマンドプロンプト等でも可)

② 以下のコマンドを入力実行する。

```
git config --global user.email “[任意のメールアドレス]”
```

→ エラーが発生しなければOK

③ 以下のコマンドを入力実行する。

```
git config user.email
```

→ ②で設定したメールアドレスが表示されればOK

②Visual_Studio_Codeに 拡張機能を追加

画面共有をしながら説明

【拡張機能追加手順】

- ① 「Visual Studio Code」を起動する。
- ② 拡張機能の検索画面を開き、「Git」と入力する
- ③ 検索結果に表示された「Git Graph」と「Git Lens」をインストールする

※ Git Graph

Gitリポジトリの履歴やブランチを視覚化し
操作を容易にするためのGUIツール

※ Git Lens

開発者がコードの変更履歴やリポジトリの状態を
視覚的に理解しやすくするためのツール

③Visual Studio Codeを用いて Gitを実際に操作

画面共有をしながら説明

【前回の復習】

■コミット(commit)

ファイルの変更記録を保存する操作

■リポジトリ(repository)

ファイルやフォルダの1つ1つの変更(コミット)情報を保存しておく場所

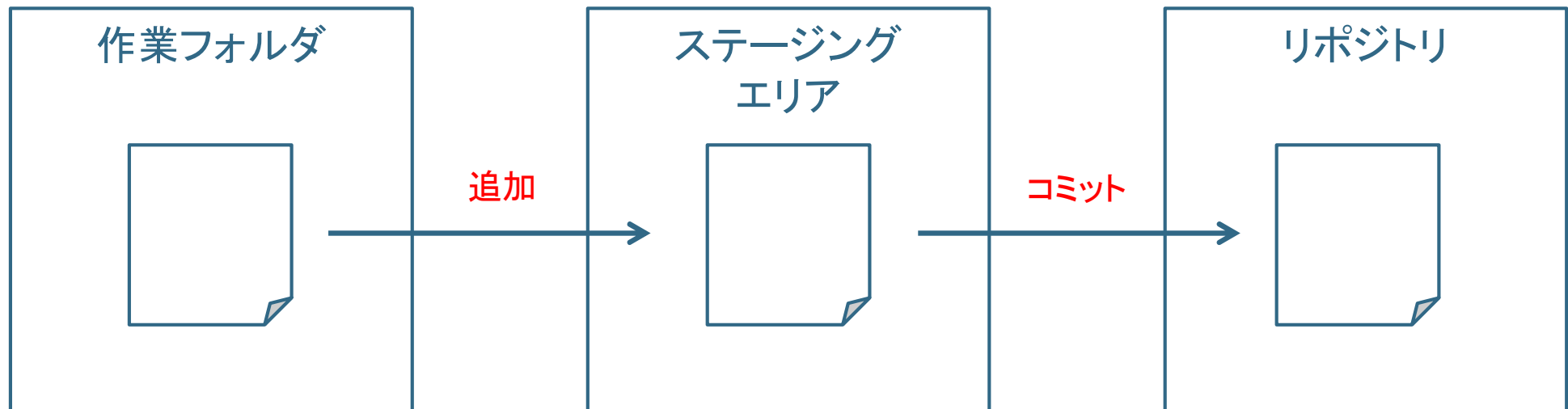
■ブランチ(branch)

履歴の流れを分岐して記録していくもの

【ステージング(staging)】

コミットする前に、変更したファイルを選択して準備する操作のこと。

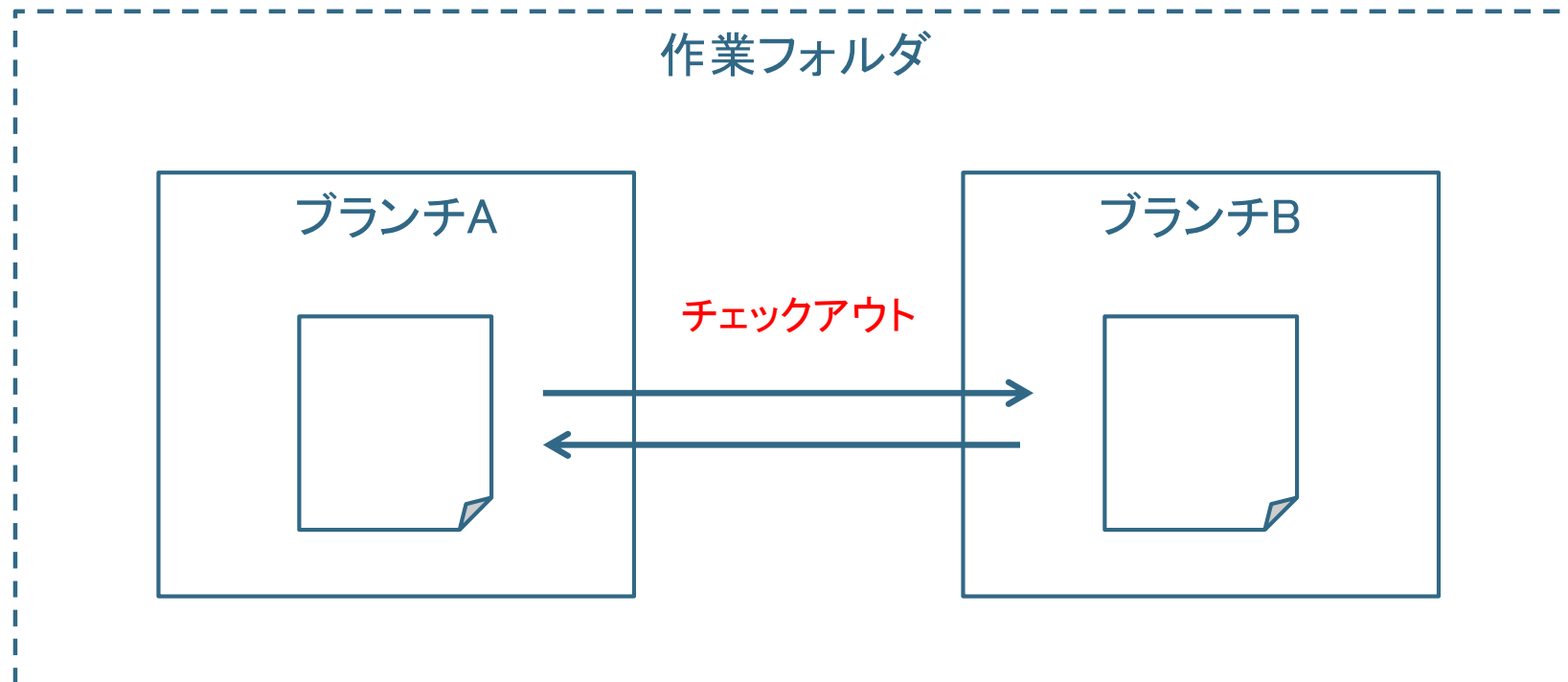
ステージングを行うことで、コミットに含める変更を明示的に指定することが可能。



【チェックアウト(checkout)】

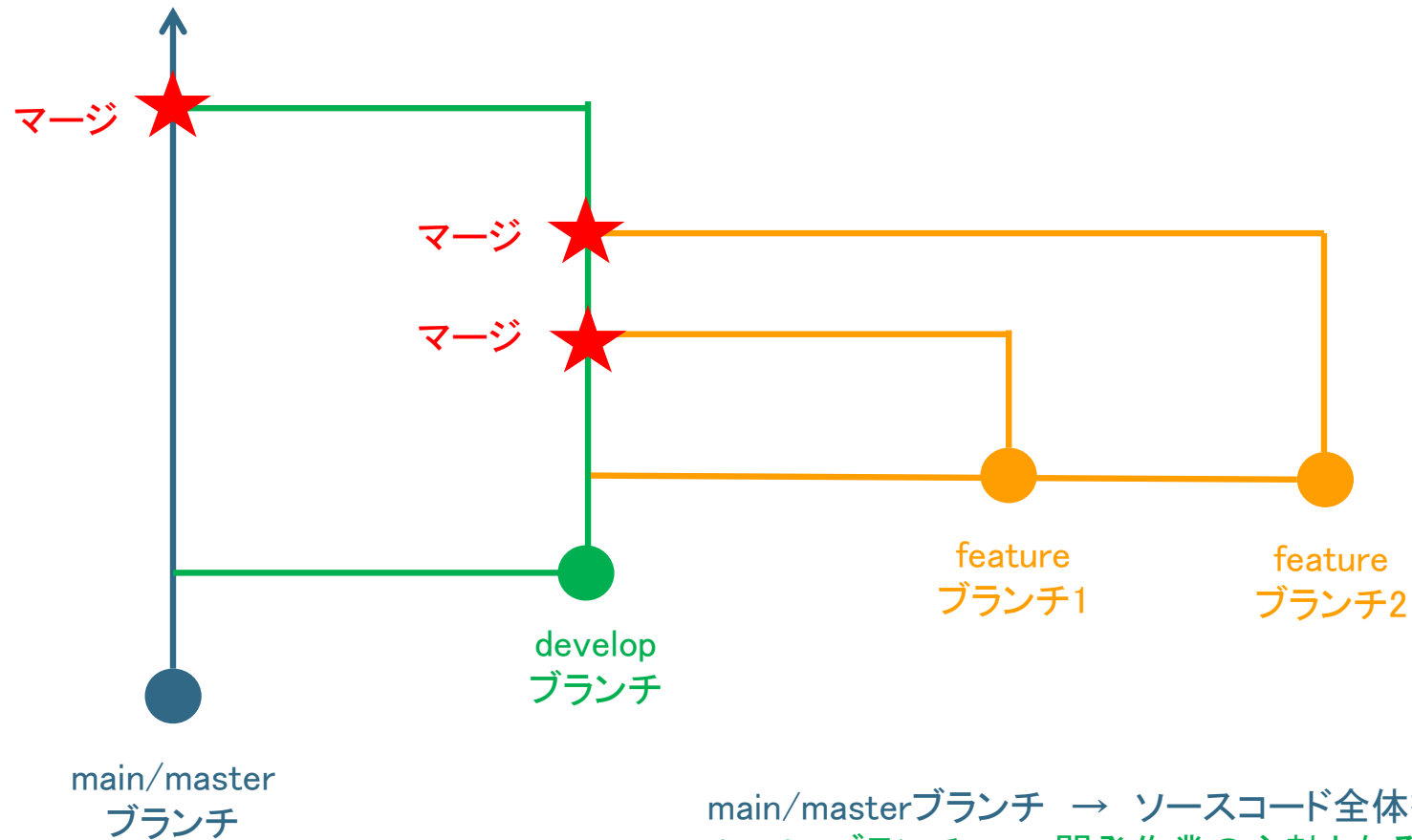
作成されたブランチ間を移動するコマンド。

ブランチをチェックアウトすることにより、作業ディレクトリ内のファイルがそのブランチに保存されているバージョンに更新される。



【マージ(marge)】

分岐した履歴を戻してひとつのブランチに統合するコマンド。



main/masterブランチ → ソースコード全体を管理するブランチ
developブランチ → 開発作業の主軸となるブランチ
featureブランチ → 実装する機能毎のブランチ