

---

# git & GitHub

gitの仕組み

---



時間目

# バージョン管理ツールとは

## 解説

チームで開発を行うと、「誰が今どの作業をしているのか」などの管理が煩雑になり、誰かがやった作業を上書きしてしまうことが発生します。

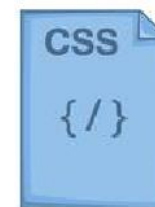
index.htmlの中の  
tableタグを修正しました。



え！？私も、さっき  
修正したのに・・・



index.html



style.css

# バージョン管理ツールとは

gitとは、このような問題を避ける為に、  
各種プログラミングのコードを  
簡単に管理できる「バージョン管理ツール」です。

## バージョン管理ツールの主な役割

- 1 ファイルの作成日時管理
- 2 ファイルの変更日時管理
- 3 ファイル内のコードの変更箇所の管理

# バージョン管理ツールとは

世の中には、たくさんのバージョン管理ツールがある  
本研修では、「git（ギット）」を学びます。

git

Subversion  
(=SVN)

Mercurial

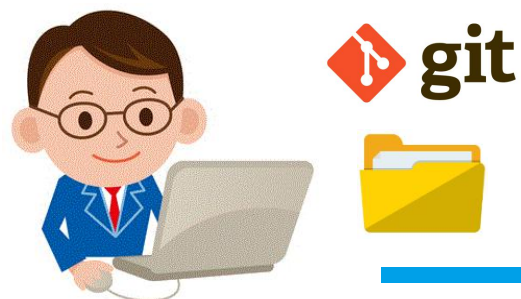
Bazaar

CVS

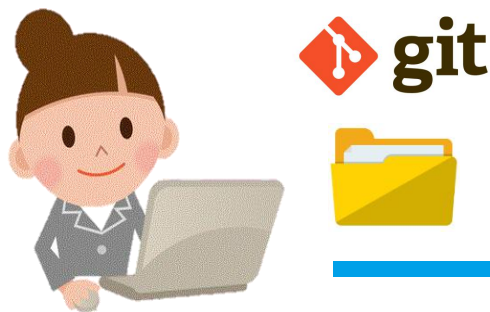
# git と GitHub

## 解説

gitは単体で使うことはなく、下記のような情報の共有場所を作ります。  
共有場所の為に自社で専用サーバーを作ることできますが、通常は「GitHub」というwebサービスを活用します。



自分の作業場所  
(=ローカルリポジトリ)



自分の作業場所  
(=ローカルリポジトリ)



**GitHub**

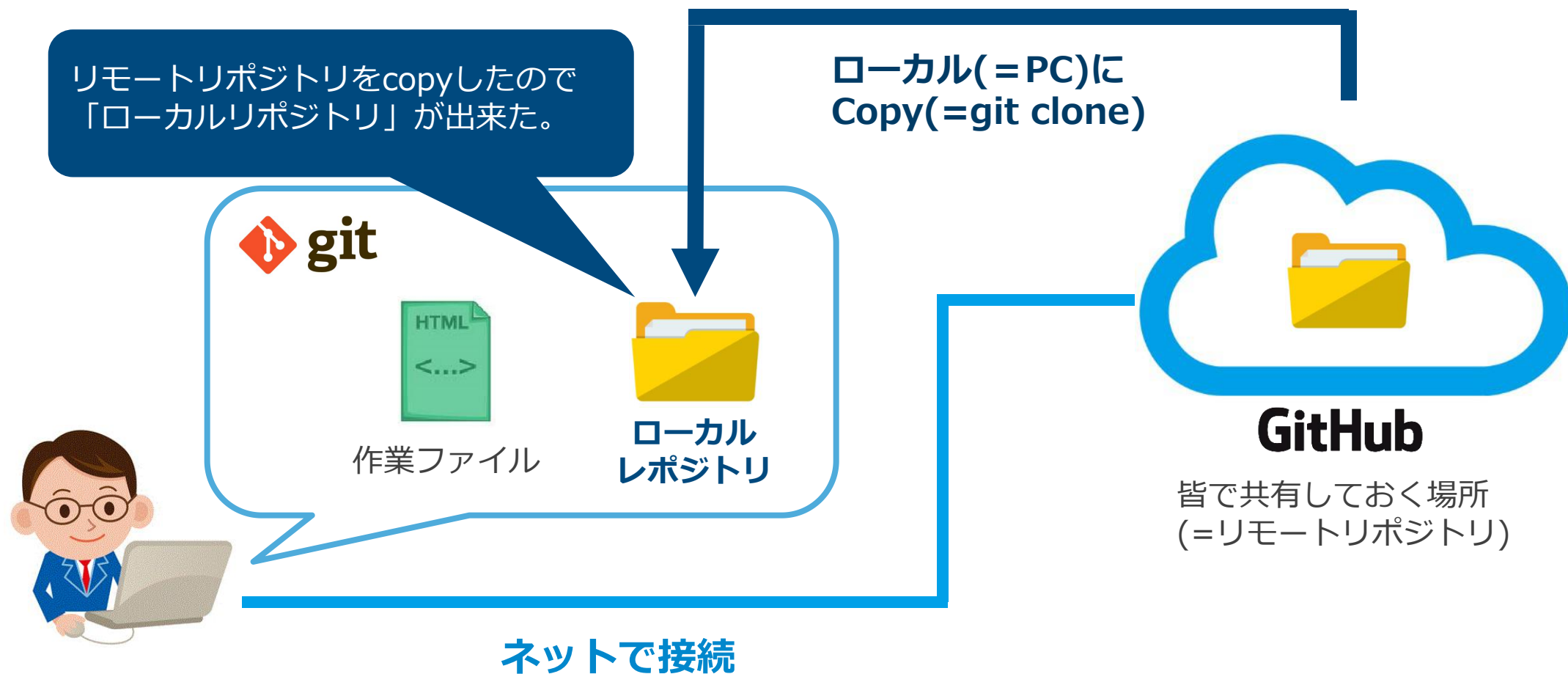
皆で共有しておく場所  
(=リモートリポジトリ)

# gitの作業手順

- 1 リモートリポジトリ（=フォルダ=ディレクトリ）をローカル（=PC）にcopy（=**git clone**）
- 2 「コードの修正・追加・削除」「ファイルの修正・追加・削除」毎に、ローカルリポジトリに変更履歴を記録（=**git commit**）
- 3 他のメンバーが「コードの修正・追加・削除」や「ファイルの修正・追加・削除」をリモートリポジトリにアップしたモノをローカルに反映させる（=**git pull**）
- 4 ローカルの変更内容をリモートリポジトリに反映させる（=**git push**）

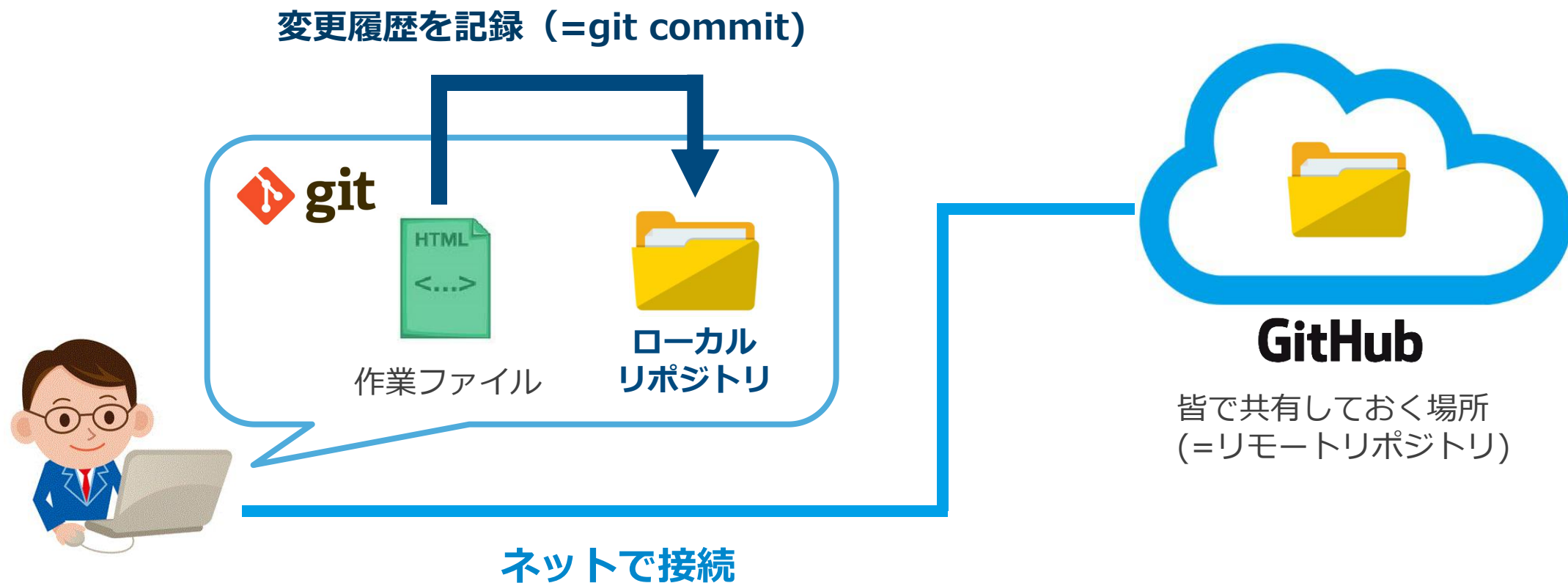
# gitの作業手順

- 1 リモートリポジトリ（=フォルダ=ディレクトリ）をローカル（=PC）にcopy(=**git clone**)

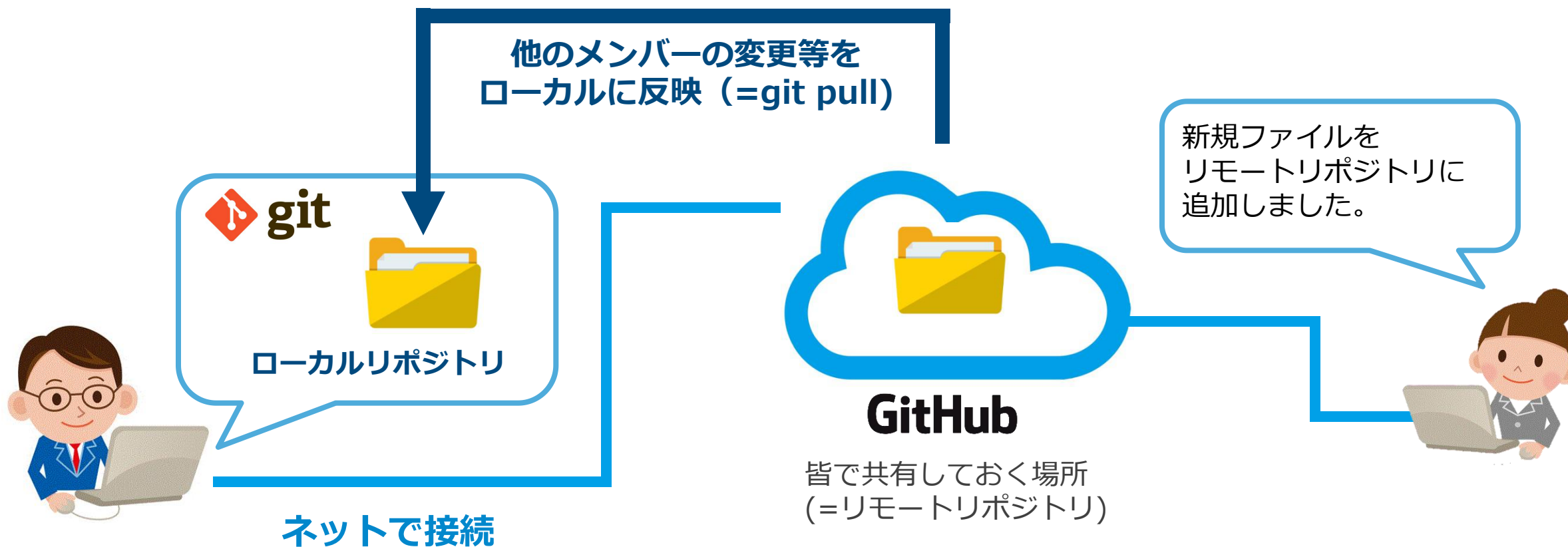




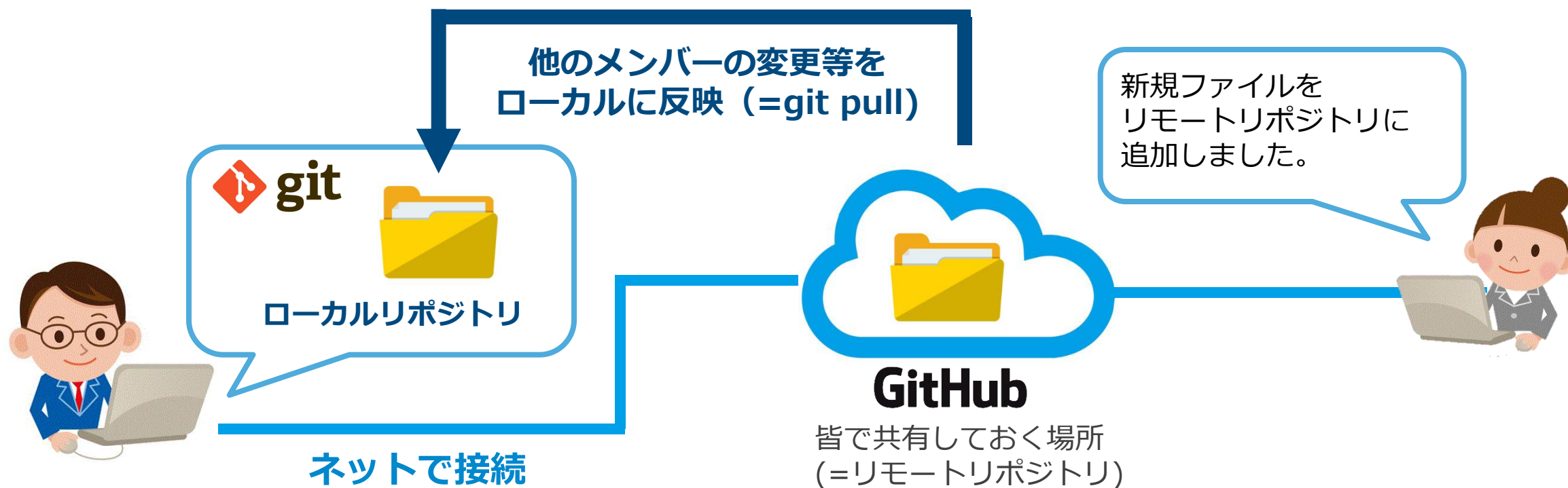
- 2 「コードの修正・追加・削除」「ファイルの修正・追加・削除」毎に、ローカルリポジトリに変更履歴を記録(=git commit)



- 3 他のメンバーが「コードの修正・追加・削除」や「ファイルの修正・追加・削除」をリモートリポジトリにしたモノをローカルに反映させる (**=git pull**)



- 3 他のメンバーが「コードの修正・追加・削除」や「ファイルの修正・追加・削除」をリモートリポジトリにしたモノをローカルに反映させる (**=git pull**)



## 4 ローカルの変更内容をリモートリポジトリに反映させる (=git push)

