

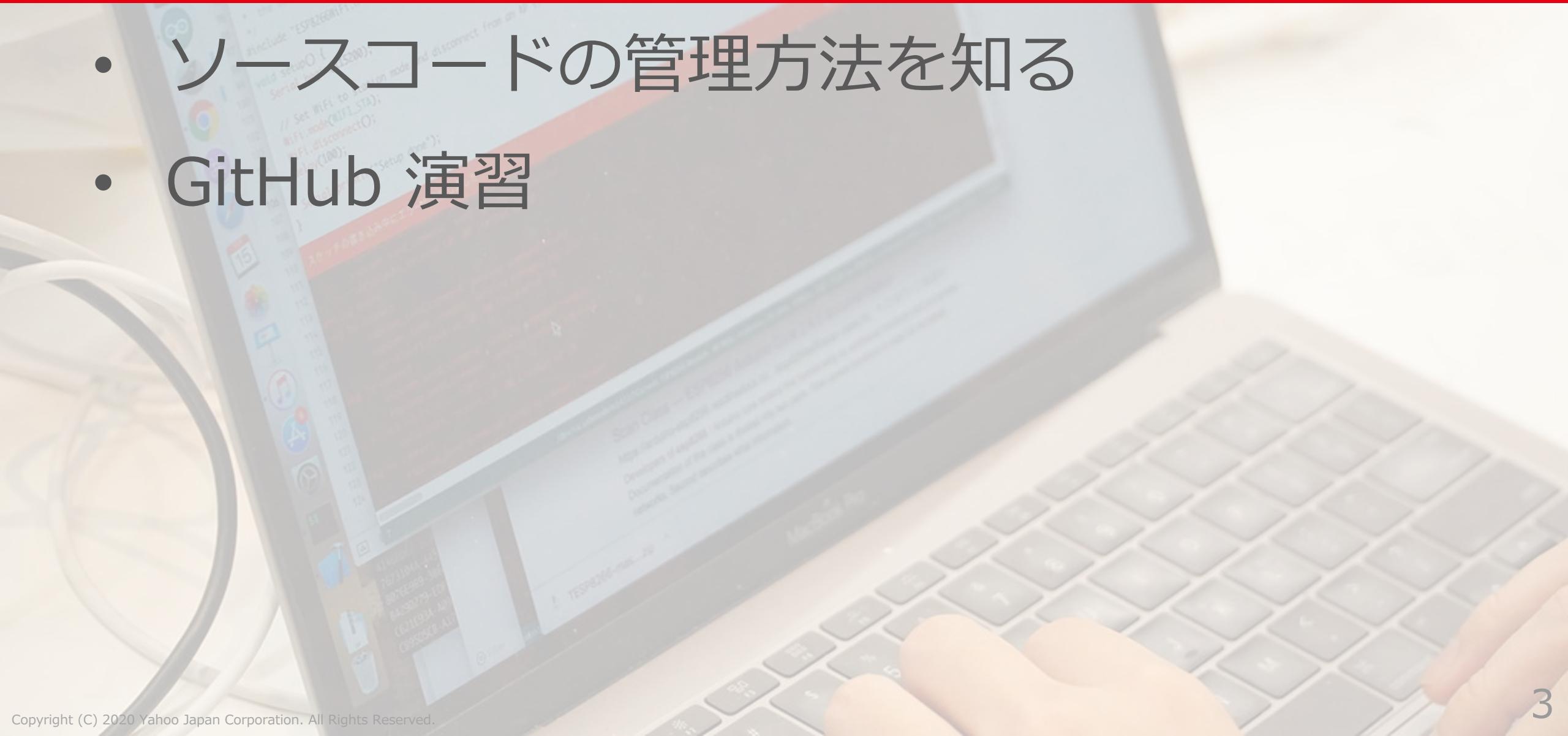
GitHub 入門編

vSCode版

Hack U Project

イベントで学ぶこと

- ・ソースコードの管理方法を知る
- ・GitHub 演習



このような経験をしていませんか？

よし、先週の開発の続きをやるぞ！



index
.html



index(1)
.html



最新版
index.html



index_final
.html

最後に編集したファイルどれだっけ？



このような経験をしていませんか？



こここの機能開発しておいたよ。
ソースコードはメールで添付して送るから取り込んでね。

ありがとう！早速コピペして・・・
あ！自分が作ってた部分が上書きされちゃった



このような経験をしていませんか？



こここの機能開発しておいたよ。

Git / GitHub を使って
解決しましょう！

ありが

あ！自分が作ってた部分が上書きされちゃった

んでね。



このイベントのゴール

- ・ コード管理の大切さを体験してもらう
- ・ 自分がつくったものをWebに公開できるようになってもらう
- ・ オープンソースを探しだして最先端の技術を見つけられるようになつてもらう

イベントの進め方

- ・ セットアップの確認
- ・ 各機能の解説
- ・ ツールの使い方
- ・ ハンズオン
- ・ まとめ
- ・ 懇親会

セットアップ
お済みですか？

YAHOO!
JAPAN

GitHub への疎通確認

The screenshot shows a GitHub repository page for the user 'github_ws'. The repository name is 'github_ws'. The 'Code' tab is selected. A blue callout box contains the text 'add/commit/push を行い' and '演習ファイルが上がっていればOK'. The commit history shows one commit from '50cdc45' made 14 minutes ago, with 1 commit. The commit details show 'src' and 'index.html' both added 14 minutes ago. At the bottom, there's a message to add a README and a 'Add a README' button.

Search or jump to... / Pull requests Issues Marketplace Explore

ユーザー名 / [github_ws](#)

<> Code Issues Pull requests > A

master 1 branch 0 tags

ユーザー名 Add index.html & src 50cdc45 14 minutes ago 1 commits

src Add index.html & src 14 minutes ago

index.html Add index.html & src 14 minutes ago

Help people interested in this repository understand your project by adding a README. [Add a README](#)

topics provided.

Releases

No releases published [Create a new release](#)

add/commit/push を行い
演習ファイルが上がっていればOK

Gitとは



バージョン管理とは

- 代表的なバージョン管理システム
 - Git
 - Subversion
 - etc
- **ファイルの変更を記録**しておくシステム
 - 誰が
 - いつ
 - どのファイルを
 - どう変更したか

バージョン管理とは

ファイルの変更を記録する場所
→ **リポジトリ**



APIサーバ
リポジトリ



Androidアプリ
リポジトリ



iPhoneアプリ
リポジトリ

バージョン管理とは

リポジトリ内のバージョン管理



バージョン管理とは

過去の状態との差分がわかる



バージョン管理とは

過去の状態に戻せる

過去の状態を最新として扱う



バージョン管理とは

- **リモートリポジトリ :**

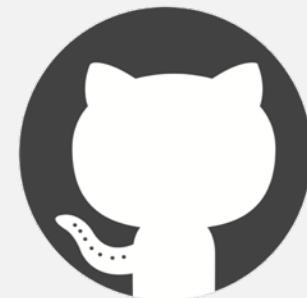
色々な人が共有できるようにしたリポジトリ
GitHub上にあるものはリモートリポジトリ

- **ローカルリポジトリ :**

自分のPC上にだけ存在するリポジトリ



ローカルリポジトリ



リモートリポジトリ

Git と Github

Git	バージョン管理の方式
GitHub	Gitでバージョン管理をしやすくする Webサービス ブラウザから変更履歴やコードを共有するのが簡単にできる

Gitを使うには

- コマンドラインでGitコマンドを使う
- GUIツールを使う
- Visual Studio CodeでGitコマンドを使う
(以後 VSCode と表現します)

Gitを使うには

- ・ コマンドラインでGitコマンドを使う
- ・ GUIツールを使う
- ・ Visual Studio CodeでGitコマンドを使う
(以後 VSCode と表現します)

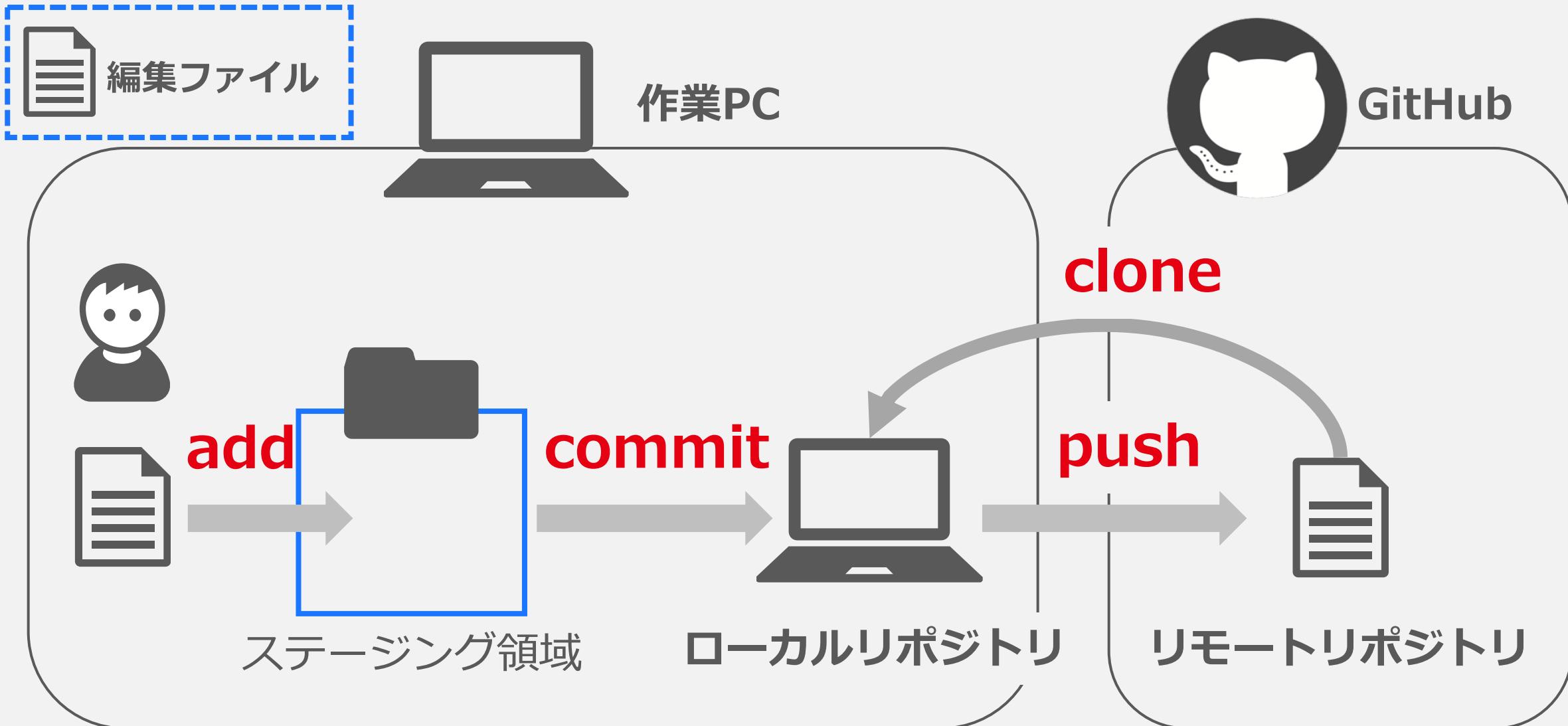
今回は VSCode で
Gitコマンド を使います

Step.1

GitHub の基本操作



GitHub の基本操作

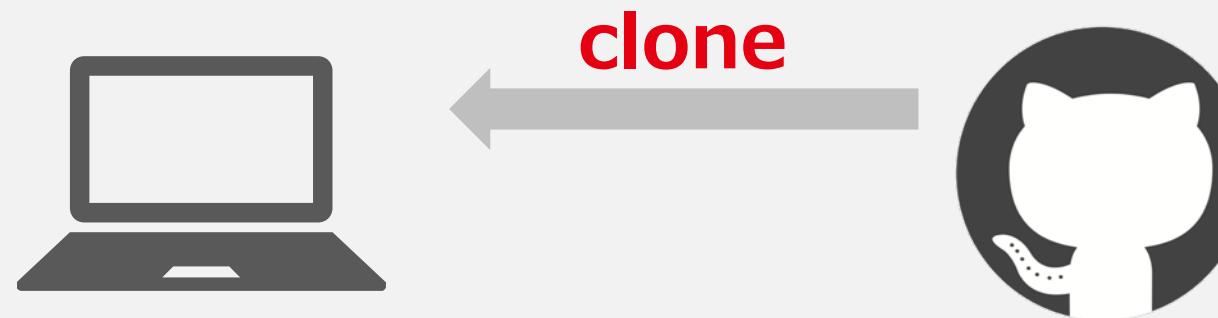


GitHub から自分のPCに持ってくる

- **clone :**

リモートリポジトリを自分のローカルリポジトリ
にコピーする

※ 初回のみの操作

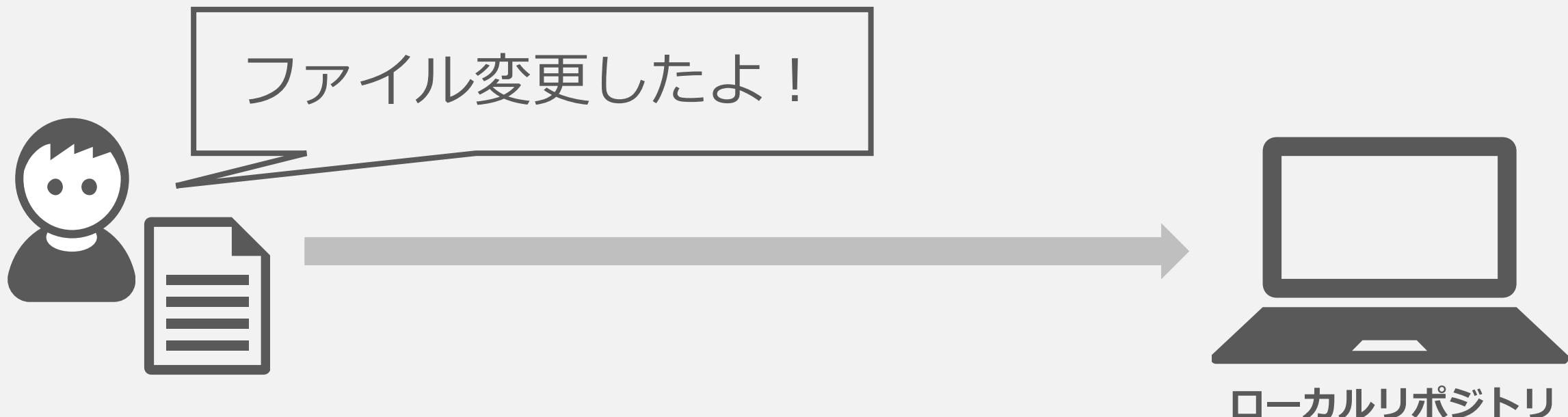


ローカルリポジトリ

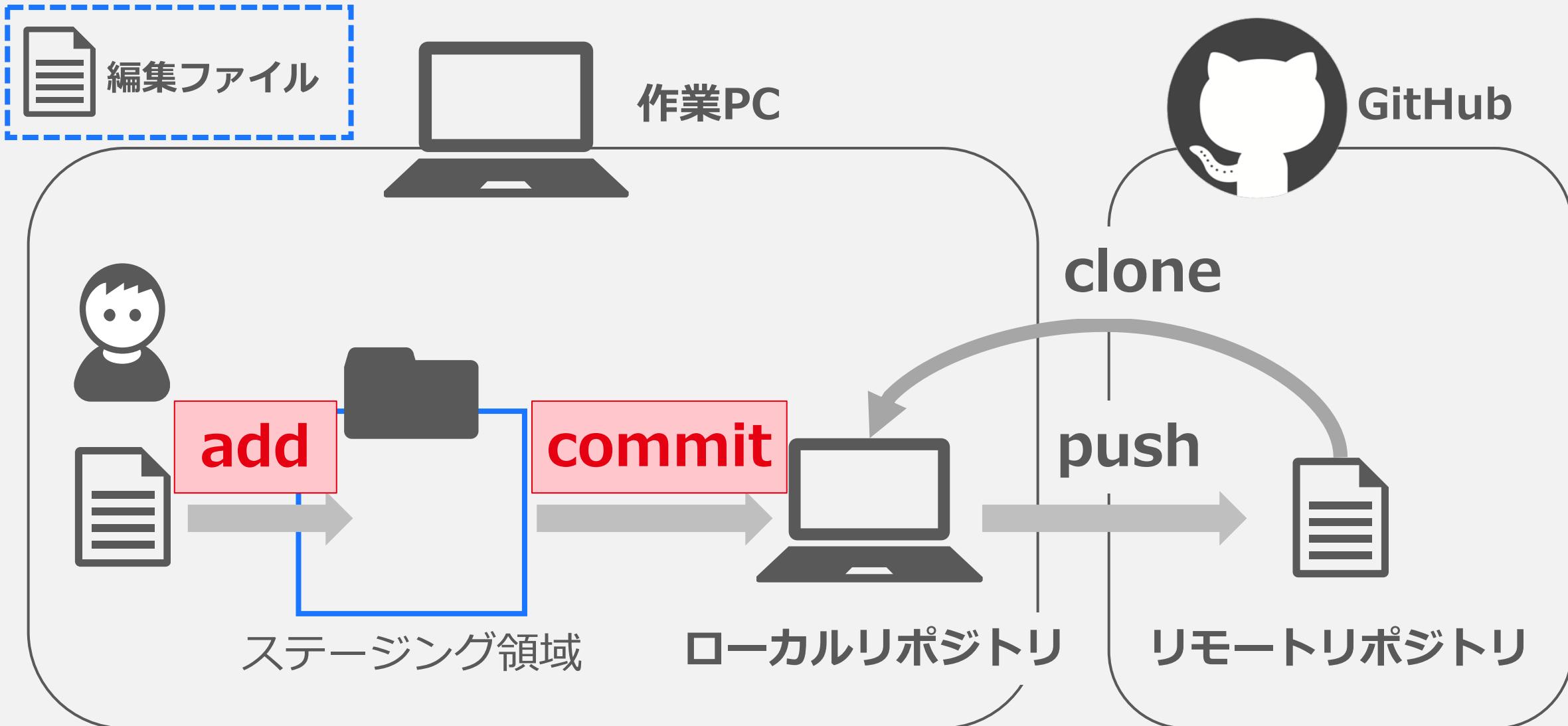
リモートリポジトリ

ファイルを修正してGitで扱う

ファイルの変更をローカルリポジトリに
登録するには**2つのステップ**が必要になります



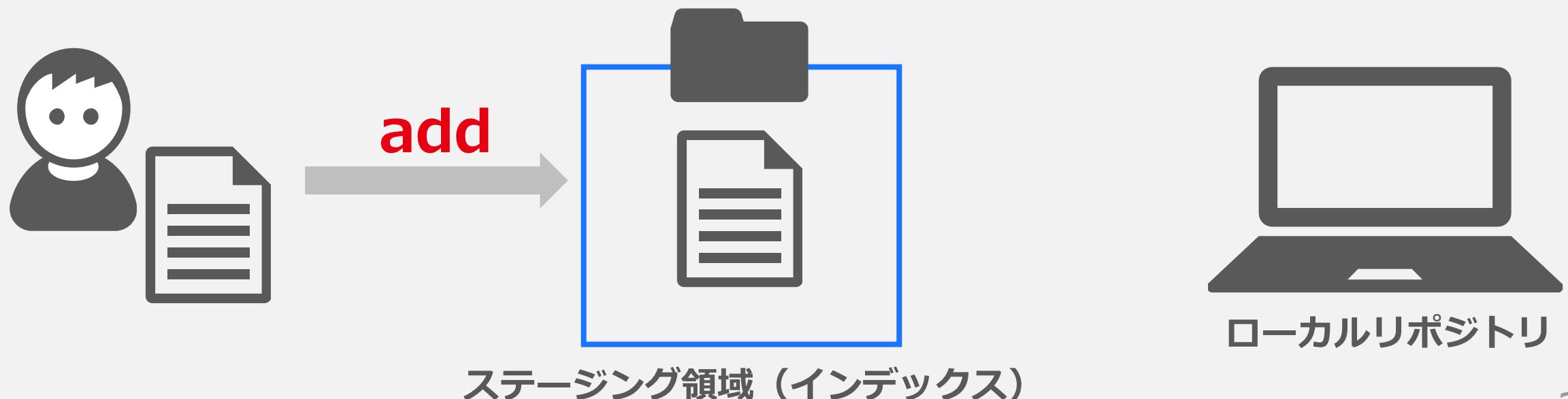
GitHub の基本操作



ファイルを修正してGitで扱う

Step.1

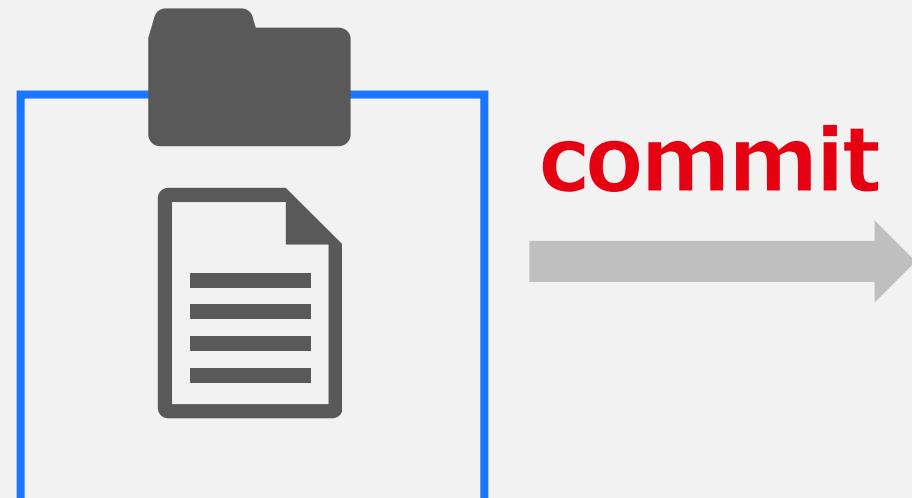
どのファイルの変更をしたかをステージング領域に一度登録する → **add** という



ファイルを修正してGitで扱う

Step.2

ステージング領域に登録した変更内容をローカルリポジトリに登録する → **commit** という



ステージング領域（インデックス）



ローカルリポジトリ

ファイルを修正してGitで扱う

Step.2の注意点

commit した時には必ず **どういう変更をしたかをコメント** で残す必要がある

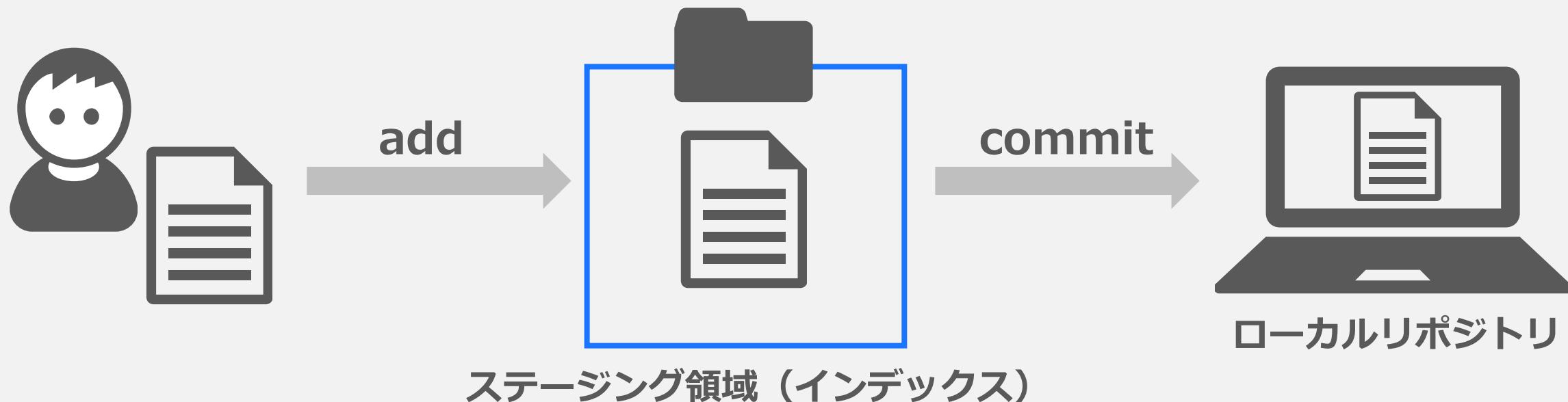


ステージング領域（インデックス）

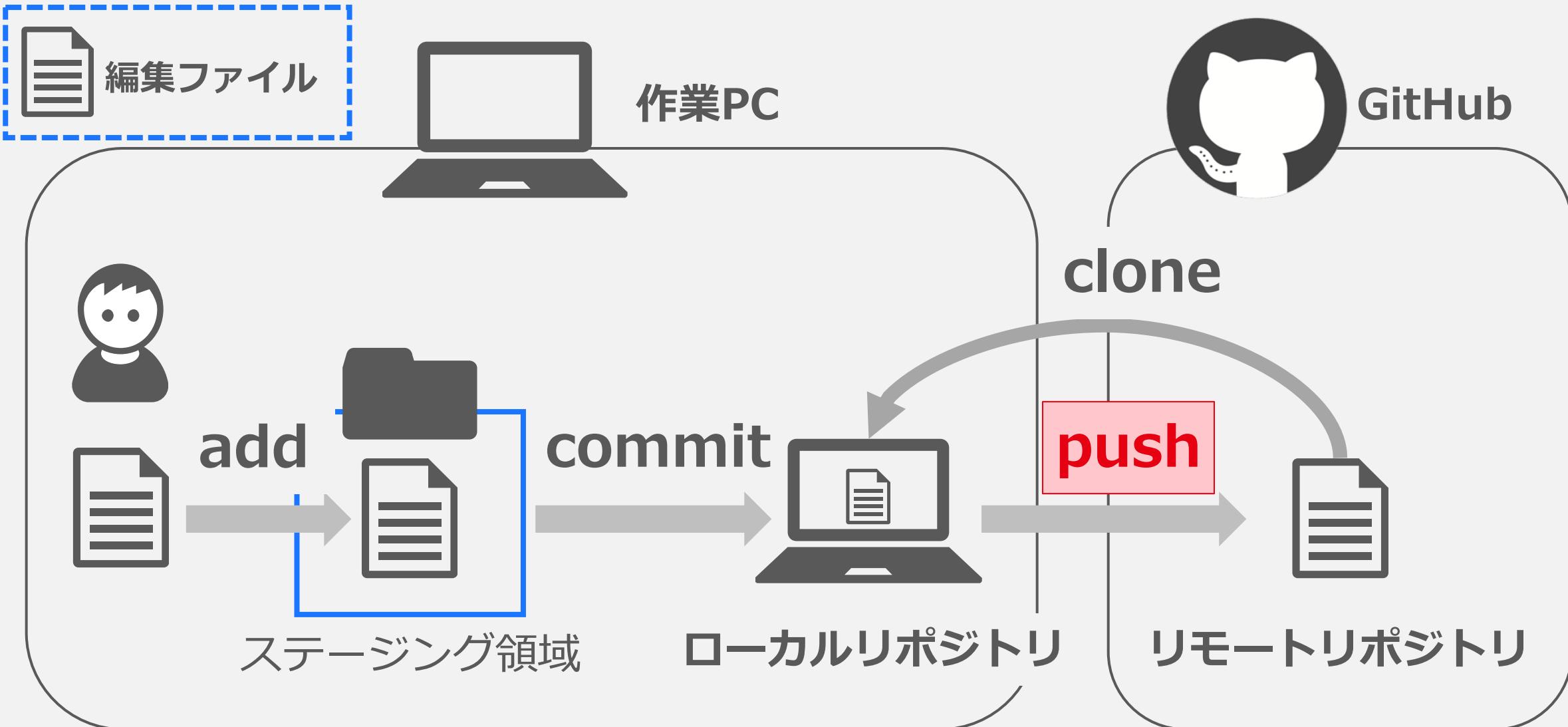


ファイルを修正してGitで扱う

- ファイルの変更は **add** して **commit**
- commit した時は**必ずコメントを残す**



GitHub の基本操作



作業PCからGitHubに反映する

push :

ローカルリポジトリの変更をリモートリポジトリに反映



ローカルリポジトリ

リモートリポジトリ

VSCodeで
cloneする

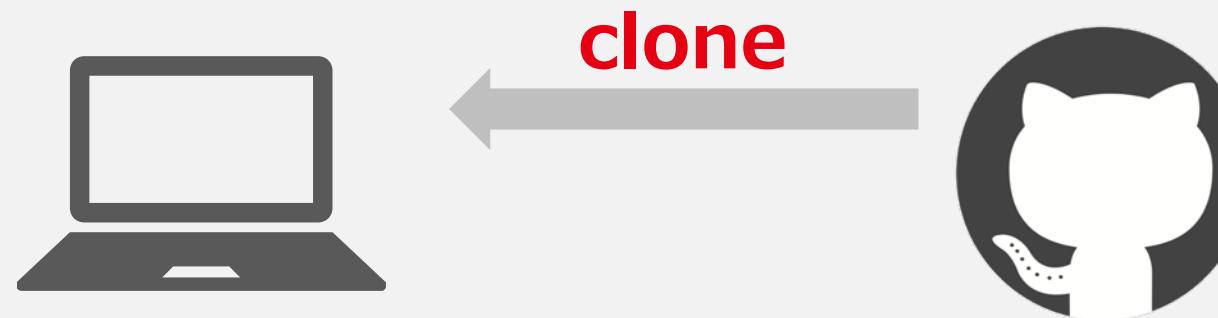


GitHub から自分のPCに持ってくる

- **clone :**

リモートリポジトリを自分のローカルリポジトリ
にコピーする

※ 初回のみの操作

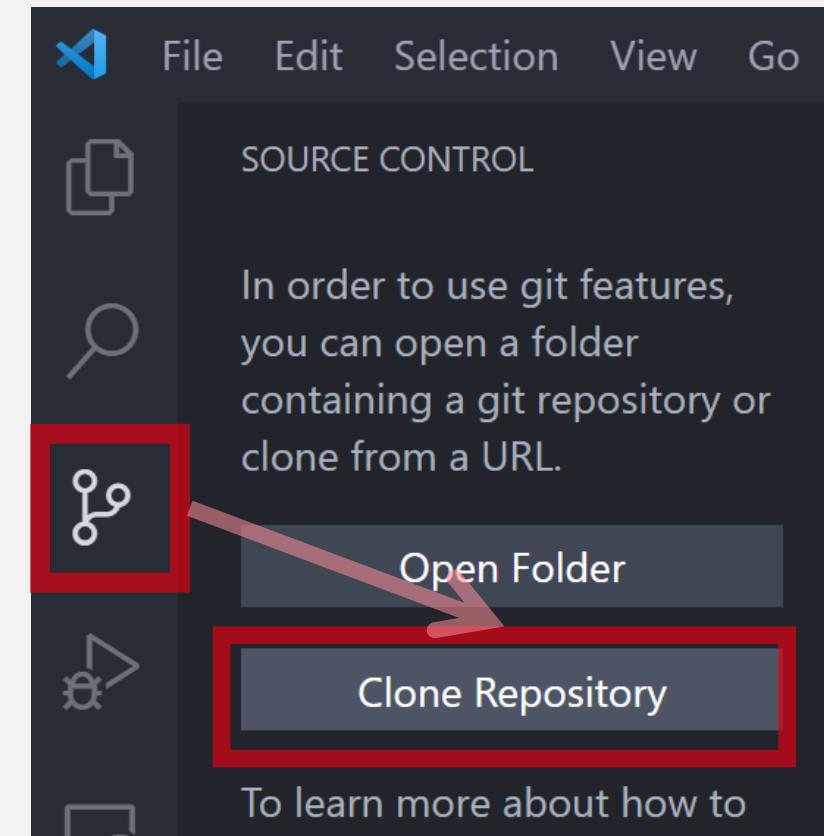
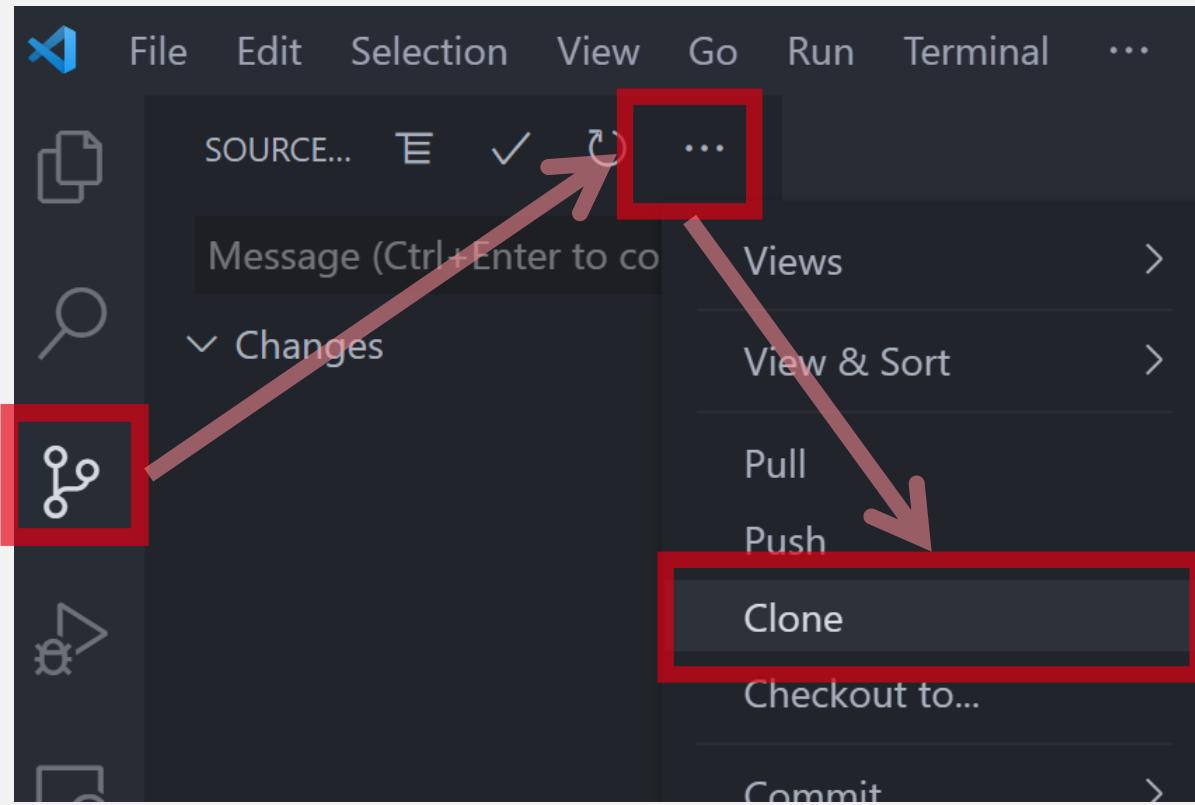


ローカルリポジトリ

リモートリポジトリ

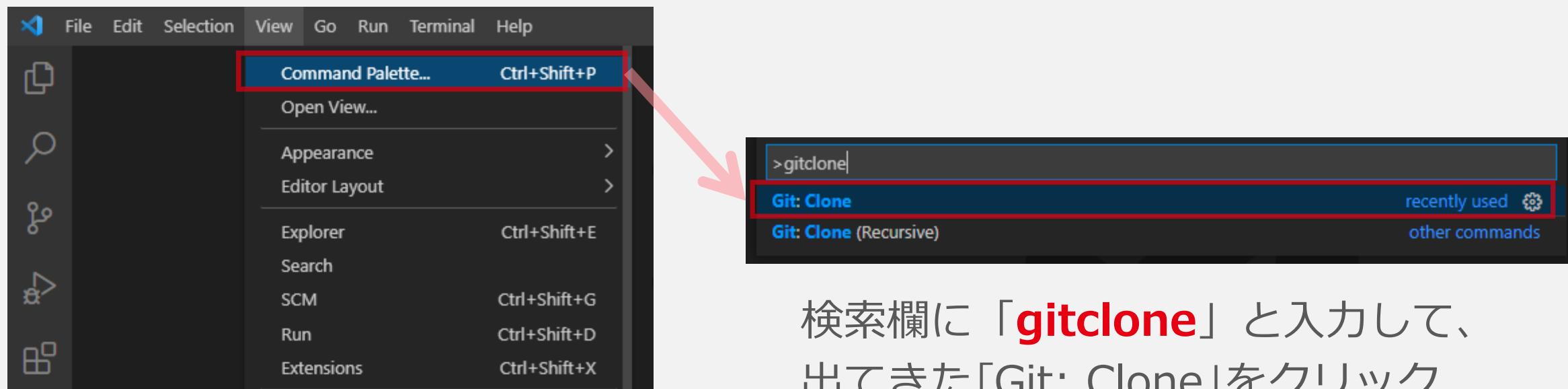
clone ①

- その時のVSCodeの状態によって、下2つのどちらかの手順でcloneを行えます



clone ① 補足

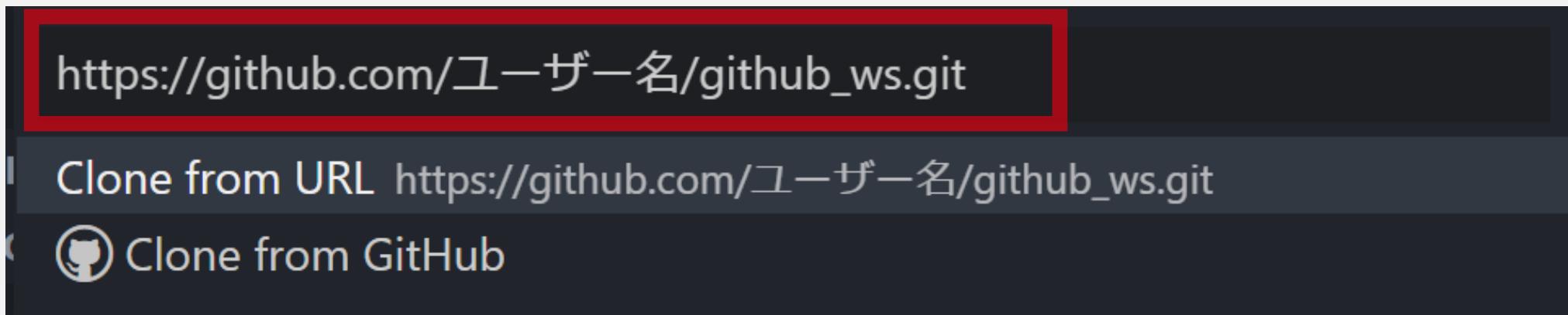
- VSCodeのUIが異なる際は、「Command Palett...」から検索します



検索欄に「**gitclone**」と入力して、
出てきた「Git: Clone」をクリック

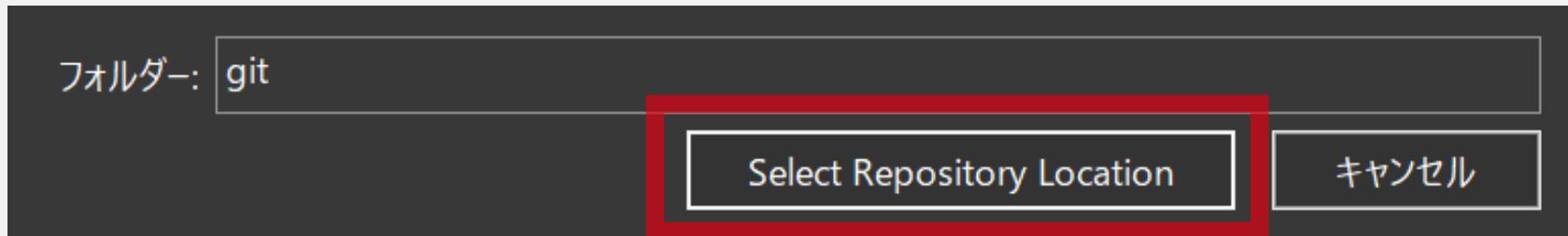
clone ②

- 出てきた入力欄に、自分で作成したリポジトリのURL
「https://github.com/ユーザー名/github_ws.git」 を入力し、Enterを押します

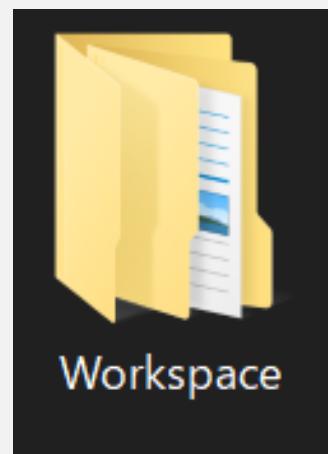
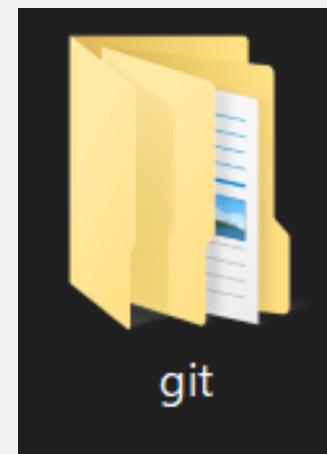


clone ③

- 適当なディレクトリを指定して
[Select Repository Location]を押す



- 「git」や「workspace」などの
ディレクトリ下にcloneすると
リポジトリが管理しやすいです



clone ④

The screenshot shows a Visual Studio Code interface. A modal dialog box is open at the top, asking "Would you like to open the cloned repository?". It has two buttons: "Open" (highlighted with a red box) and "Open in New Window". Below the dialog, the main VS Code window displays the "EXPLORER" view. A folder named "GITHUB_WS-1" is expanded, showing subfolders "src" and "index.html". The status bar at the bottom indicates the current branch is "master". A large red arrow points from the "Open" button in the dialog down to the "src" folder in the Explorer view.

- Openを押したら
Clone完了です

VSCodeで
addする



ファイルを修正してGitで扱う

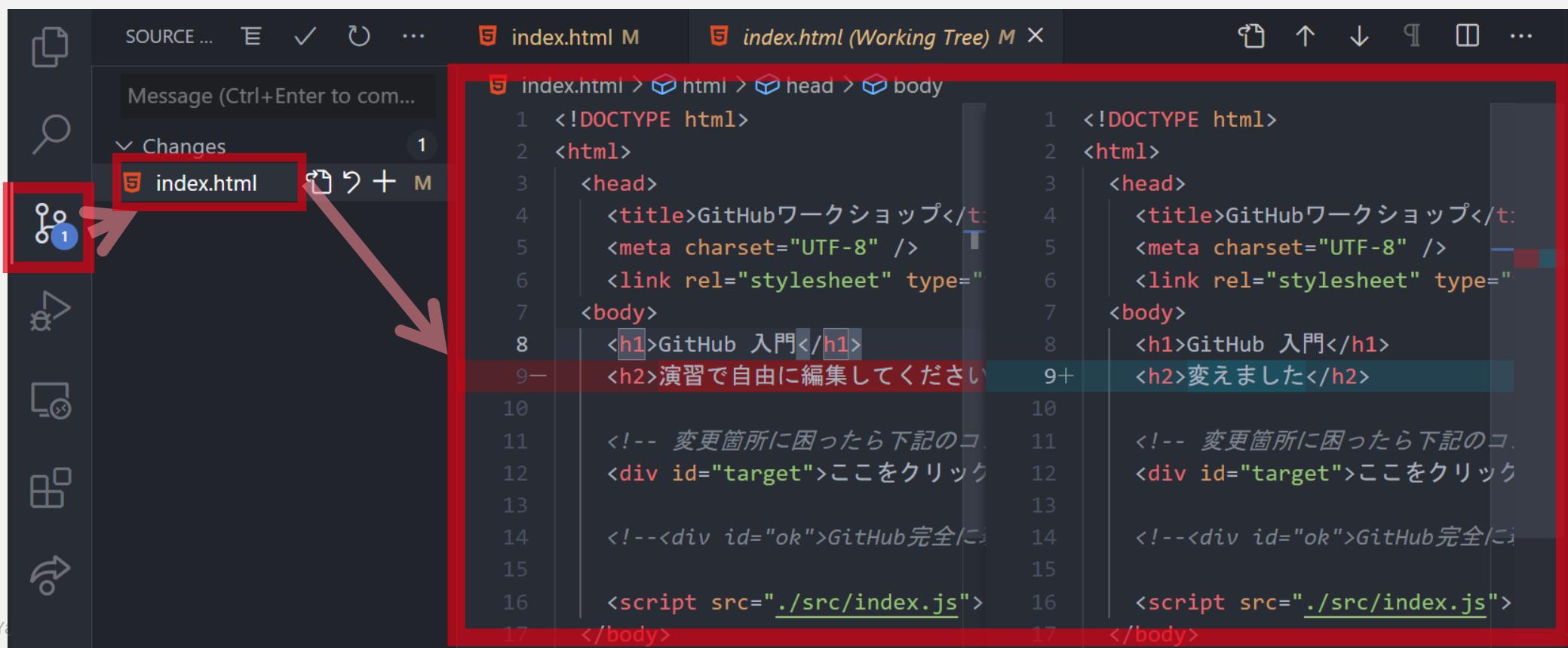
Step.1

どのファイルの変更をしたかをステージング領域に一度登録する → **add** という



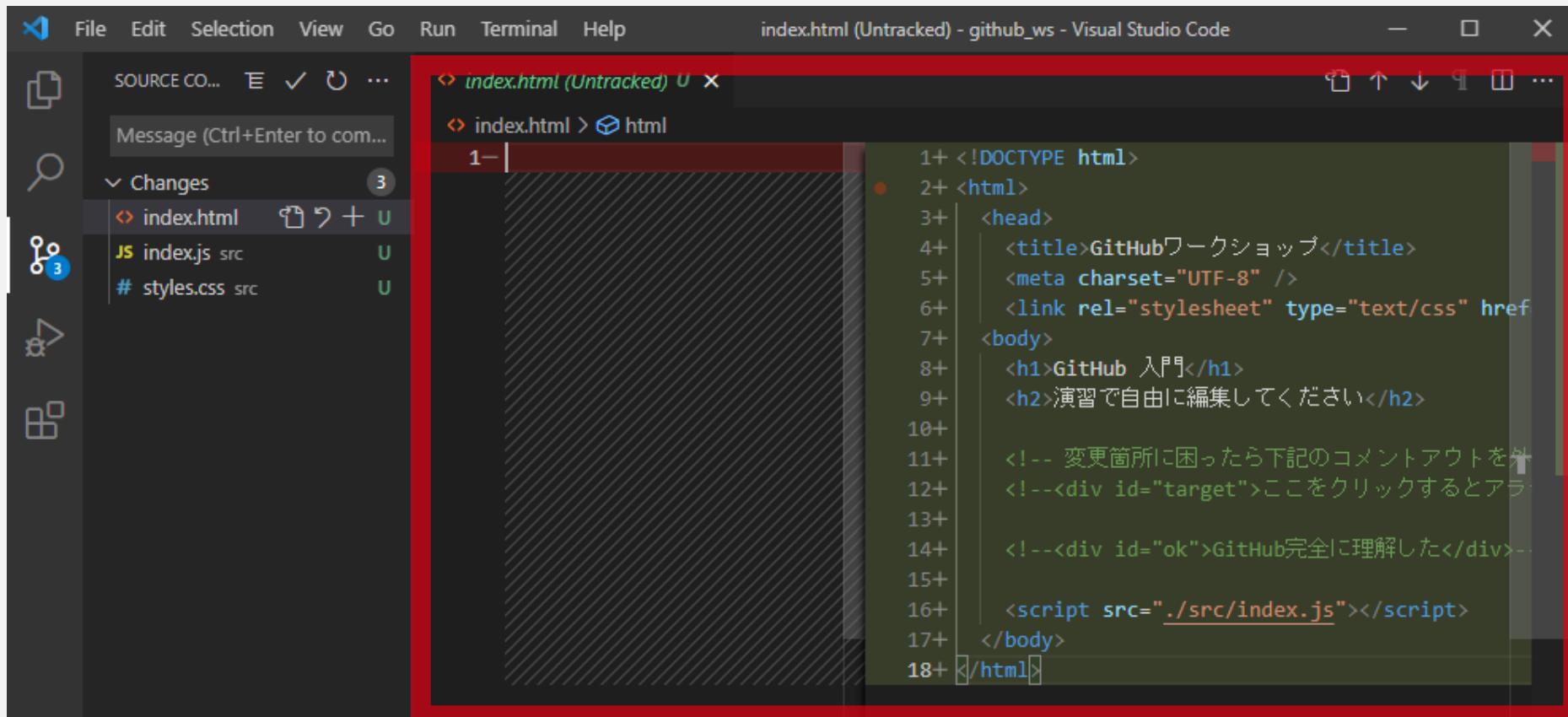
ステージングにaddする

- addする前にVSCodeのメニューから変更内容を確認できます



ステージングにaddする

- 新規ファイルをaddする際は下記のように差分表示されます

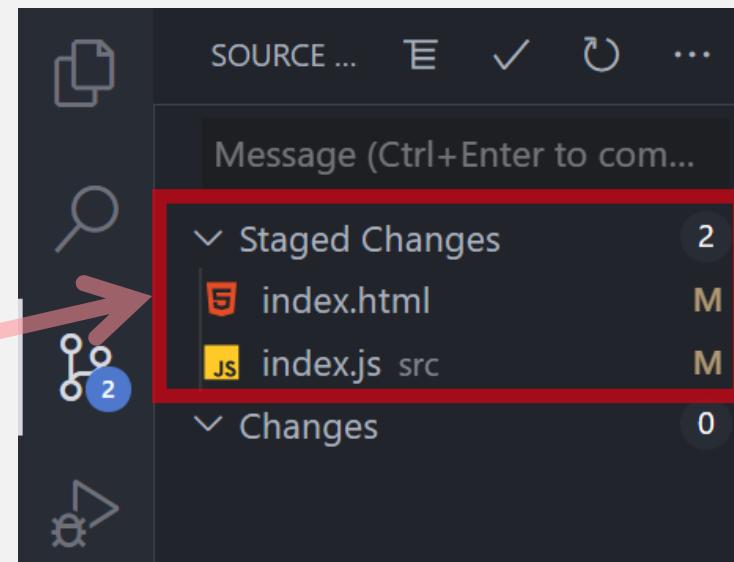
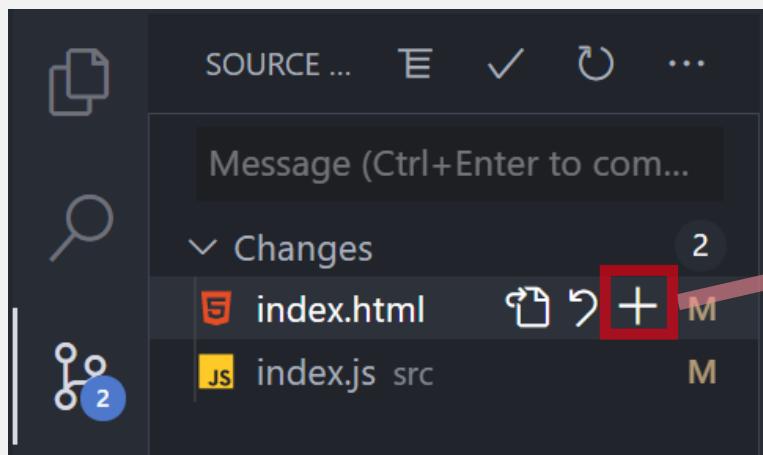


The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** index.html (Untracked) - github_ws - Visual Studio Code.
- Sidebar:** Shows a tree view with "Changes" expanded, listing "index.html" (Untracked), "index.js src" (Untracked), and "# styles.css src" (Untracked). A message "Message (Ctrl+Enter to com..." is also present.
- Editor Area:** The main editor shows the content of index.html. The code is color-coded for syntax (HTML, CSS, JS). The first line is "1<!DOCTYPE html>" and the last line is "18</html>". The code includes a title, meta charset, link to styles.css, and a body section with h1 and h2 headings. There are several comments starting with "11--<!--" followed by explanatory text.

ステージングにaddする

- addしたいファイルの「+」をクリックすると、「Changes」から「Staged Changes」にファイル名が移動します。
addするファイルを全て「Staged Changes」に入れましょう。



addする際の注意点

- add するファイルの内容を確認
 - 個人情報
 - パスワード
 - API Key などの鍵情報

リモートリポジトリに push したファイルは
消すことができません
これらの情報を含んだファイルは add しない
ように注意しましょう

VSCodeで commitする

YAHOO!
JAPAN

ファイルを修正してGitで扱う

Step.2

ステージング領域に登録した変更内容をローカルリポジトリに登録する → **commit** という



commit

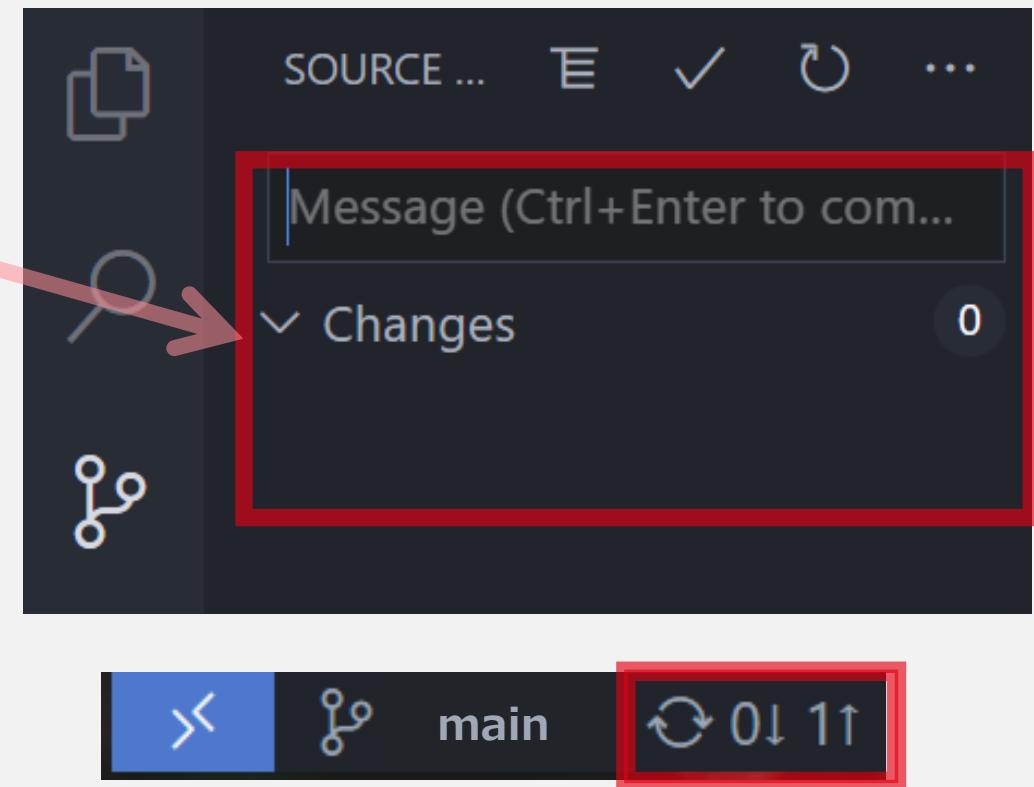
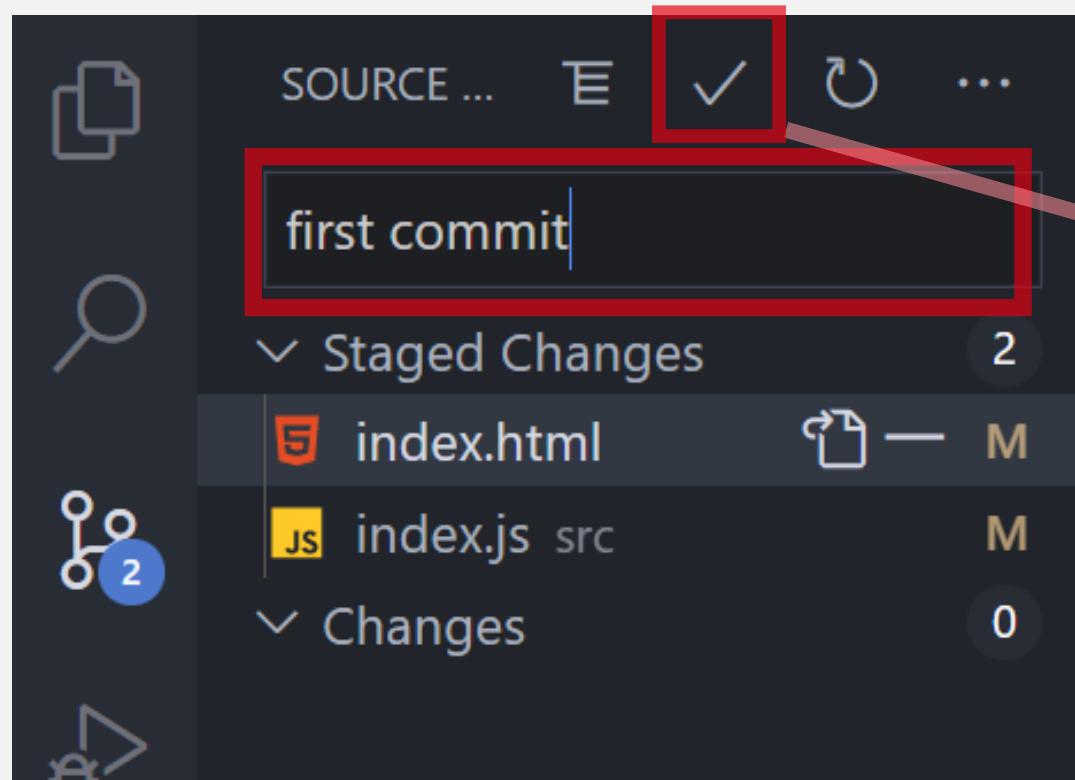


ローカルリポジトリ

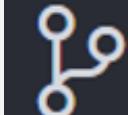
ステージング領域（インデックス）

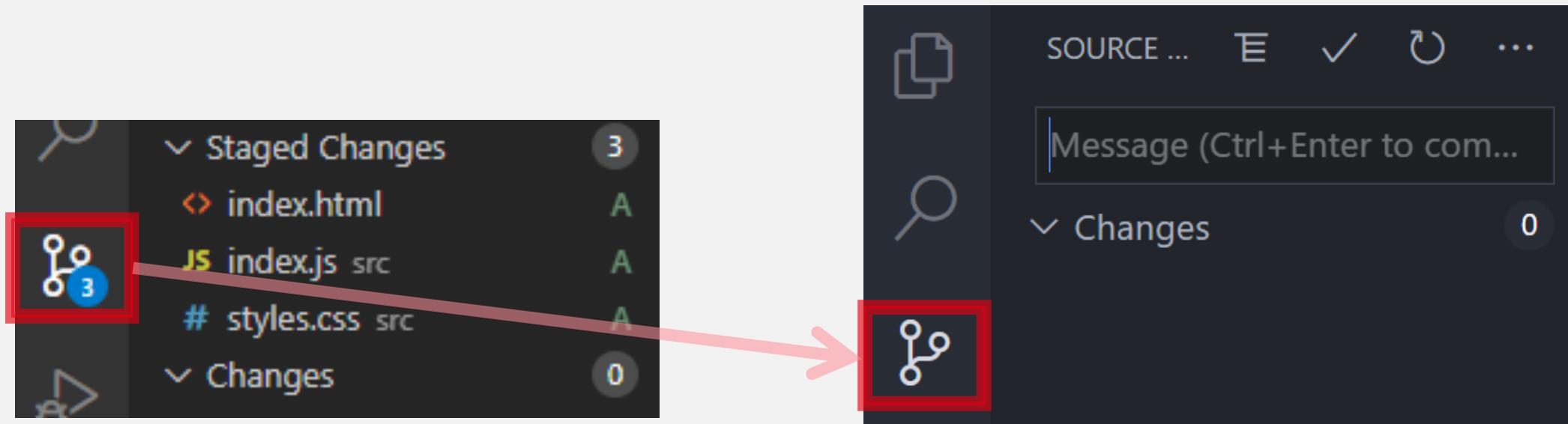
ローカルリポジトリにcommitする

- 任意のメッセージを入力し✓をクリックすると表示が更新されて画面下が「 $0 \downarrow 1 \uparrow$ 」となります



ローカルリポジトリにcommitする

- OSの違いやGit設定によっては画面下が「0↓0↑」のままである場合があります
- その場合は  の数字が消えていればOKです



VSCodeで
pushする



自分のPCからGitHubに反映する

push :

ローカルリポジトリの変更をリモートリポジトリに反映

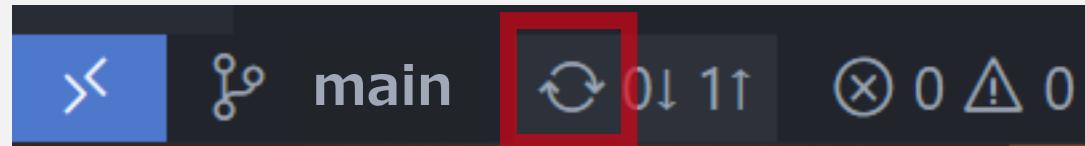


ローカルリポジトリ

リモートリポジトリ

GitHubへの疎通確認 (push)

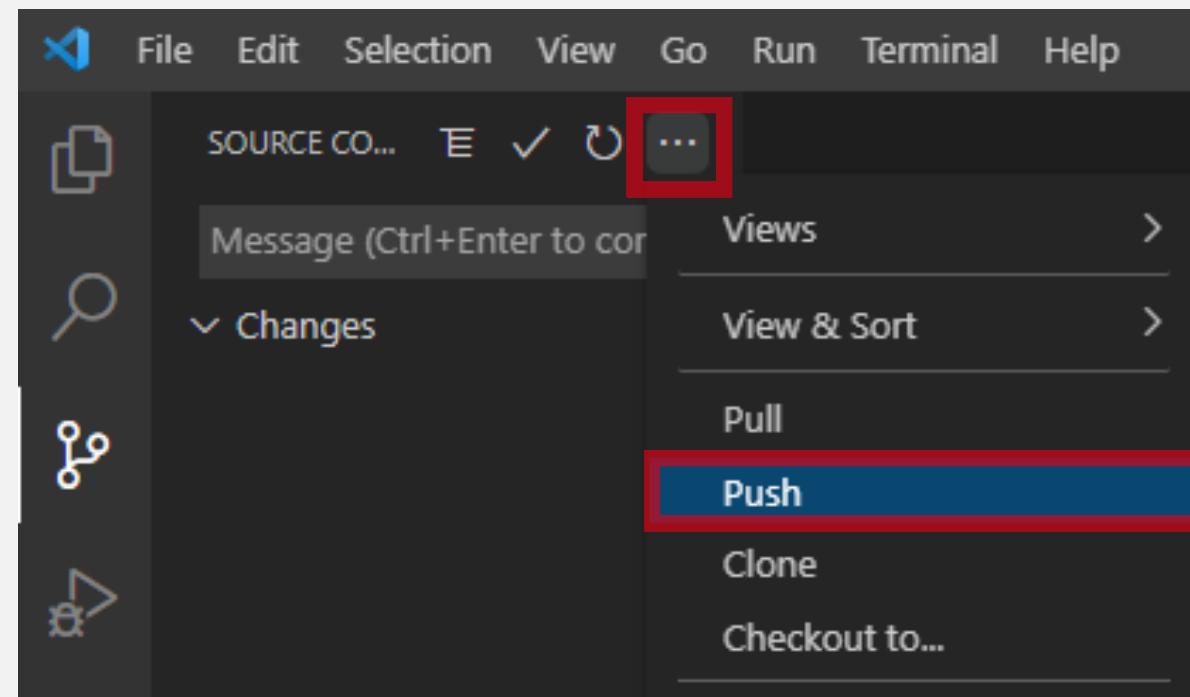
- 更新ボタン（赤枠のところ）を押すとpushされます



- 更新ボタンは押すことでpushの他にpullも行なうことができます
pullができるときは「 $1\downarrow 0\uparrow$ 」のように表示されます（状況により表示されない場合もあります）

GitHubへの疎通確認（push）補足

- 更新ボタンでのpushがうまく行かないときは、下図の「…」から「Push」を選択してください



GitHub に反映されているか確認

Screenshot of a GitHub repository page for the user 'github_ws'.

The repository details:

- Name: ユーザー名 / **github_ws**
- Code: master branch, 1 branch, 0 tags
- Last commit: 50cdc45, 14 minutes ago, 1 commits
- Commit details:
 - src: Add index.html & src, 14 minutes ago
 - index.html: Add index.html & src, 14 minutes ago
- Actions: Go to file, Add file, Code
- Watch: Unwatch (1)
- Star: Star (0)
- Fork: Fork (0)

About section:

No description, website, or topics provided.

Releases section:

No releases published

Add a README

Bottom navigation bar:

- Code
- Issues
- Pull requests
- Actions
- Projects
- Wiki
- Security
- Insights
- ...

GitHubに反映されているか確認

The screenshot shows a GitHub repository page for a user named 'github_ws'. The page includes a navigation bar with links for Pull requests, Issues, Marketplace, and Explore. Below the navigation bar, there are buttons for Unwatch (1), Star (0), and Fork (0). A main menu at the top of the repository page includes Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and a three-dot menu. A large blue callout box is overlaid on the page, containing the URL https://github.com/ユーザー名/github_ws and the text 'こちらにサンプルファイルが上がっていればOKです' (If a sample file is uploaded here, it's OK). At the bottom of the repository page, there is a message encouraging the addition of a README file, a green 'Add a README' button, and a 'Create a new release' link.

https://github.com/ユーザー名/github_ws

こちらにサンプルファイルが上がっていれば
OKです

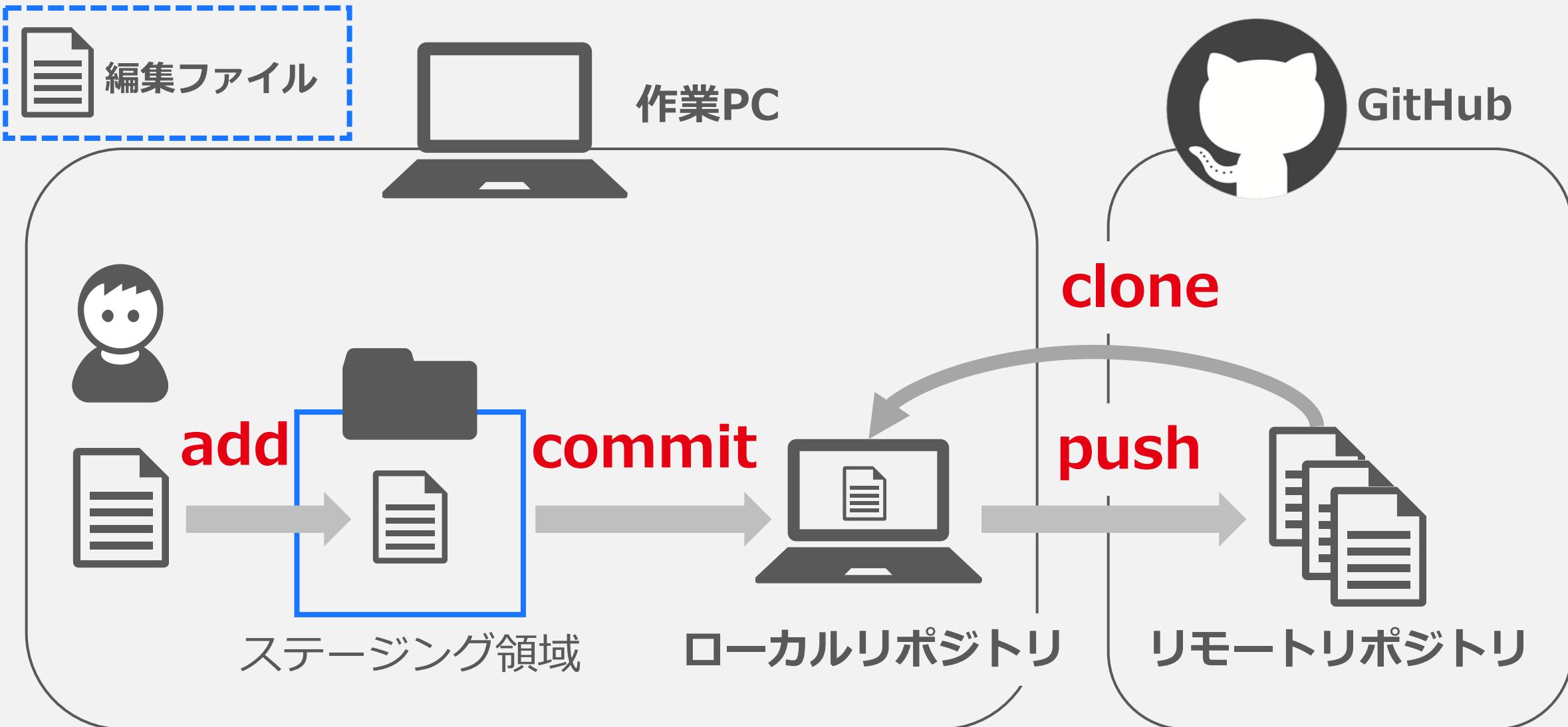
Help people interested in this repository understand your project by adding a README. [Add a README](#)

Create a new release

Pushがうまくいかない時①

- ネットワークの問題
 - 学校のWifiなどを使用していると、ネットワークの設定で制限がかけられ、Pushができない場合があります。
 - 別のネットワークやスマホのテザリングなどで再度試してみましょう

基本操作まとめ



Step.1

ハンズオン



ハンズオンの作業内容 その1

- ・ ファイルを追加/変更して、リモートに push
1. README.mdの新規作成 / index.htmlを編集する
 2. 作成/変更したファイルをaddする
 - ・ ファイルの差分を確認しよう

ハンズオンの作業内容 その2

3. インデックスされたファイルをcommitしてみよう
 - コミットコメントは変更内容がわかるように記述しよう
4. ローカルリポジトリの変更 をpushしてみよう
5. リモートリポジトリを確認してみよう

README.md とは

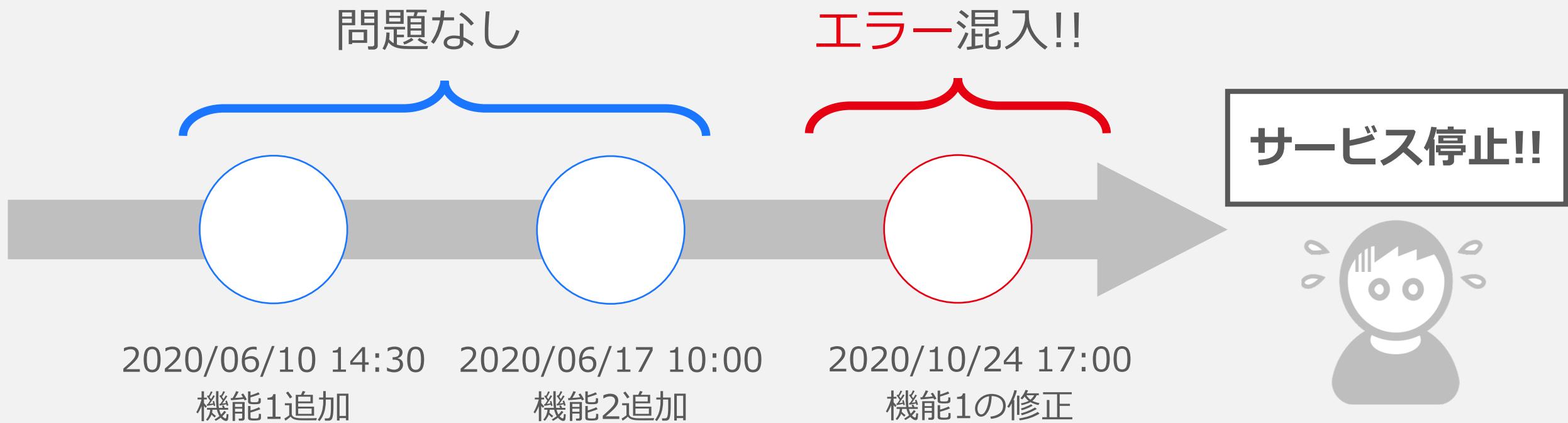
- **リポジトリで最初に見るべき資料**
 - 使用方法
 - 開発ルール
 - 変更履歴 ...など
- **自分のリポジトリの説明を追加しましょう**
 - Markdown記法で記述できます

Step.2-1

作業用ブランチの作成

ブランチ

リポジトリに動かないコードが上がってしまう



ブランチ

リポジトリに動かないコードが上がってしまう

既存のコードにバグを含まないよう

作業環境を分けましょう

機能1追加

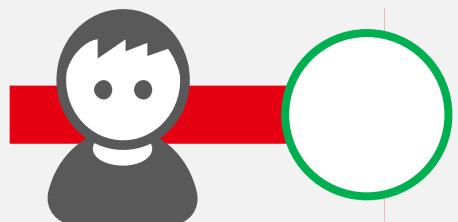
機能2追加

機能1の修正

止!!

ブランチ

最新の状態



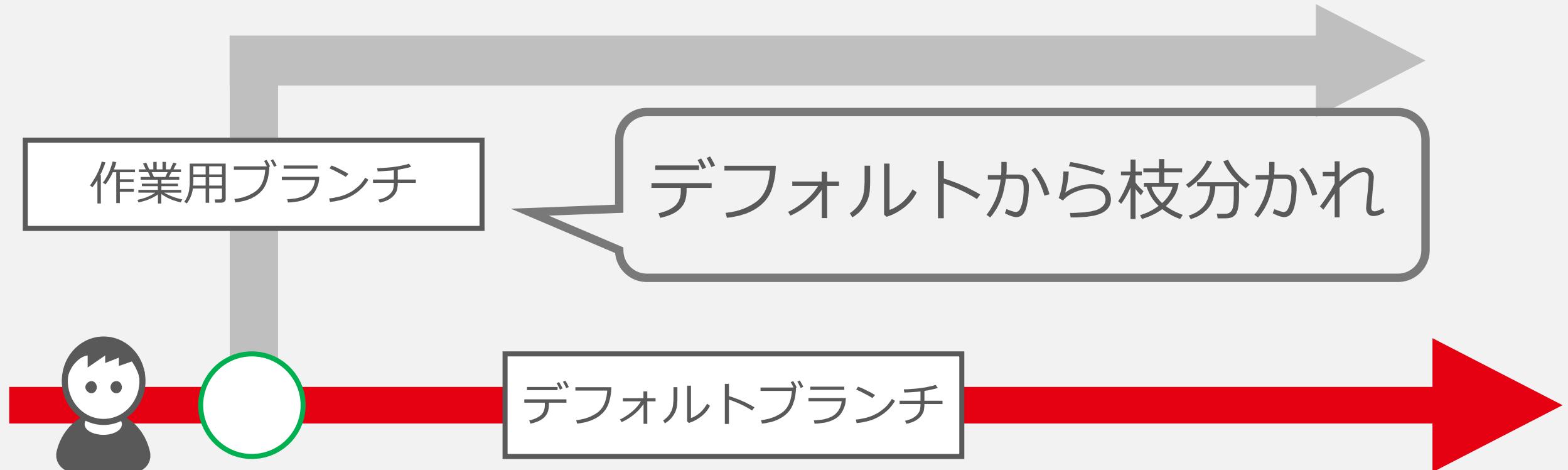
大元のブランチ

→ **デフォルトブランチ**

- ブランチ = 枝
デフォルトブランチから
枝を生やす
 - デフォルトブランチの名称は
変更可能
- ※この資料では**main**を使用

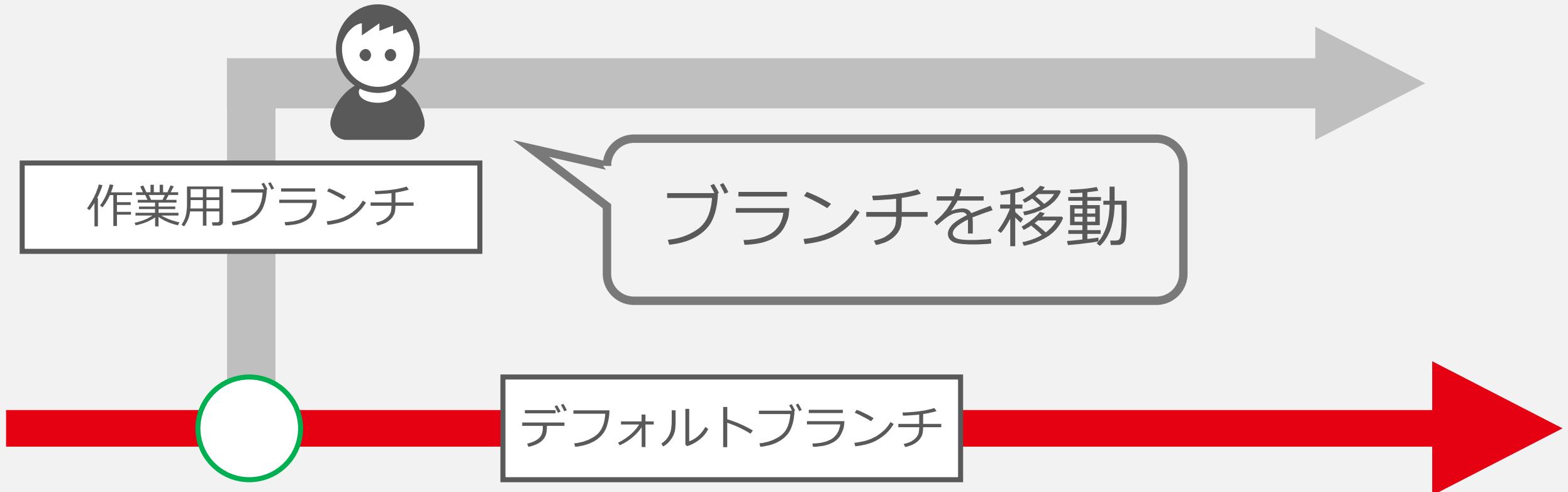
ブランチ

作業用ブランチを作成する



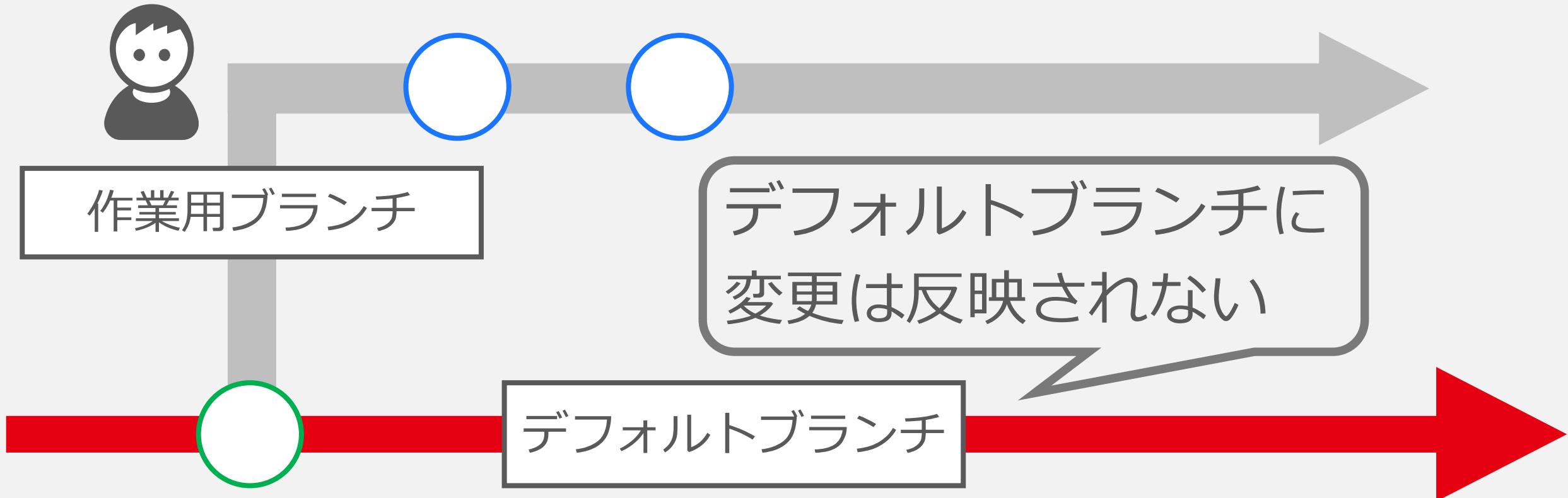
ブランチ

作業用ブランチにチェックアウトをする



ブランチ

作業ブランチで同じように commit/push する



VSCodeで ブランチを作成する



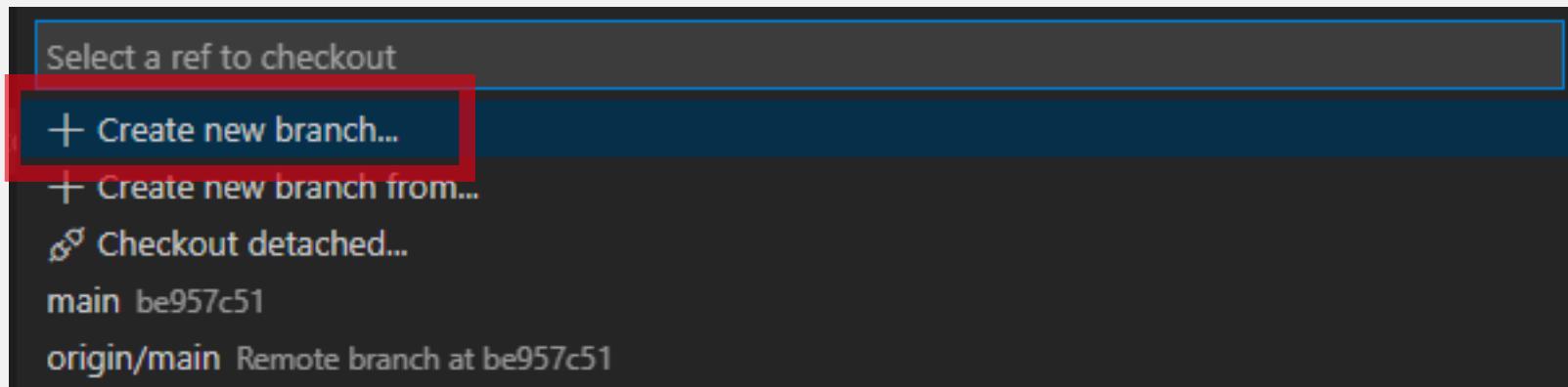
ブランチを作成する

- GitHubの「ユーザーネーム」でブランチを作成します

- 画面下のブランチ名をクリック



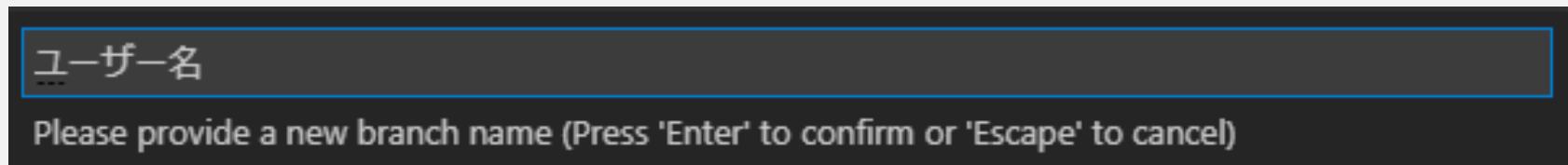
- 「Create new branch...」を選択



ブランチを作成する

- GitHubの「ユーザー名」でブランチを作成します

- 「Branch name」の入力欄にユーザー名を入力



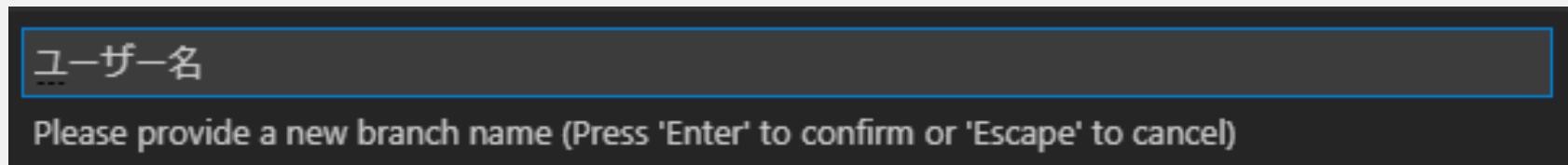
- 画面下のブランチ名が変更されたのを確認



ブランチを作成する

- GitHubの「ユーザー名」でブランチを作成します

- 「Branch name」の入力欄にユーザー名を入力



- 以下の手順で、新規ブランチを作成します。

ブランチ名は自分で好きに決められますが、今回は新規
ブランチ名をGitHubアカウントと同じにしましょう。

ブランチを指定して
ファイル変更



ブランチを作成してファイルをPush

- ファイルを変更してadd → commit
 - addする前にブランチがmainでないことを必ず確認しましょう
- commitしたら作成したブランチをリモートにpushします
 - mainブランチでpushしないように注意

GitHub でブランチを確認

The screenshot shows a GitHub repository interface. At the top, there is a navigation bar with tabs: Code (highlighted with a blue border), Issues, Pull requests, Actions, Projects, Wiki, and Security. Below the navigation bar, there is a yellow banner with the message "ユーザ名 had recent pushes less than a minute ago" and a green button labeled "Compare & pull request". In the center, there are several status indicators: a dropdown menu showing "main" (highlighted with a blue border), "2 branches", and "0 tags"; a "Go to file" button; an "Add file" button; and a "Code" button. Below these, there is a modal window titled "Switch branches/tags" with a search bar containing "Find or create a branch...". The modal has two tabs: "Branches" (highlighted with a blue border) and "Tags". Under the "Branches" tab, there is a list with one item: "main" (marked with a checkmark). Below the list is a button labeled "View all branches". At the bottom of the page, there is a "README.md" file.

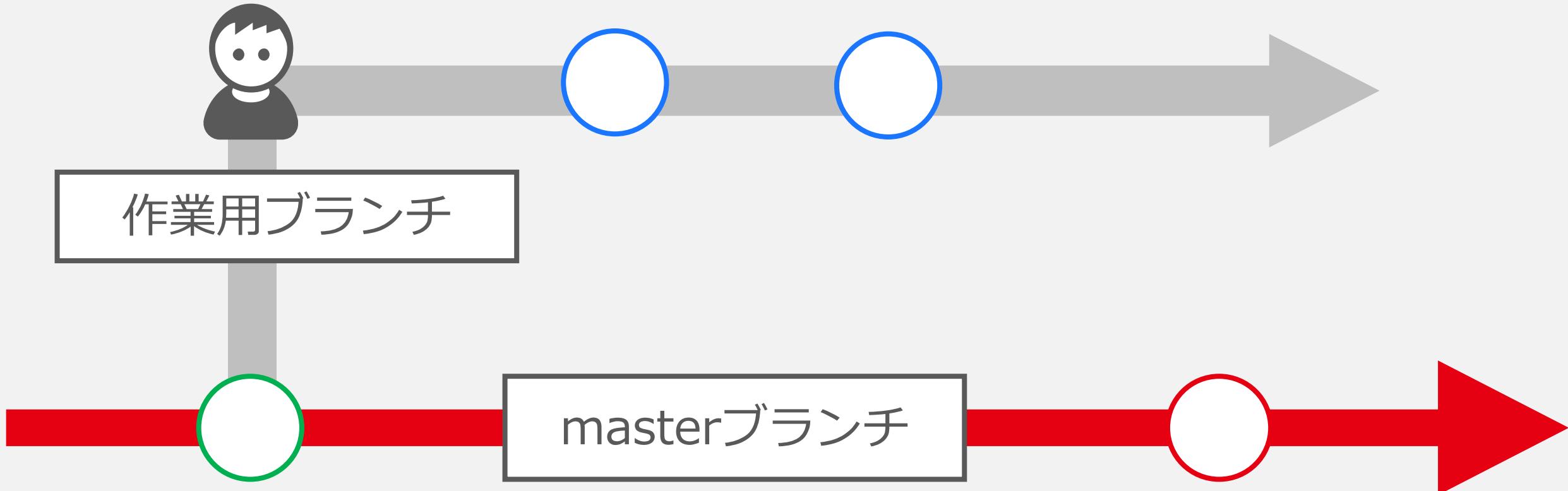
Code → main をクリック
Branches にユーザー名があればOK
クリックするとそのブランチの内容を確認できます

Step.2-2

mainに変更を反映する

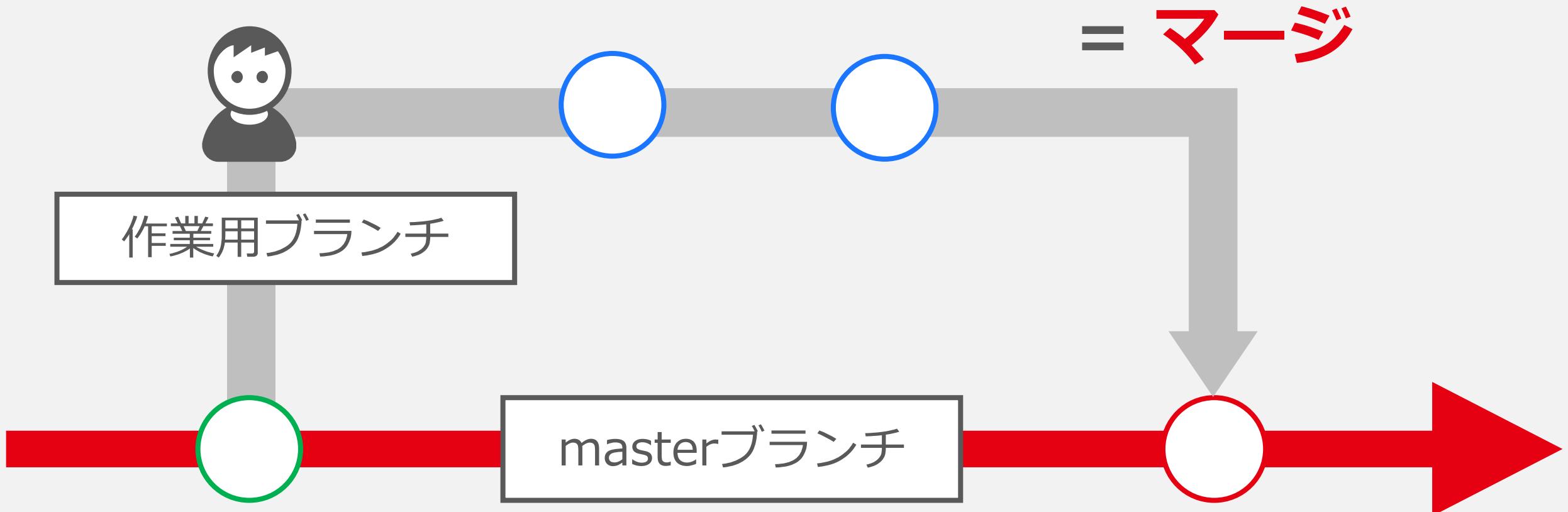
マージ

作業用ブランチの変更を master に反映する



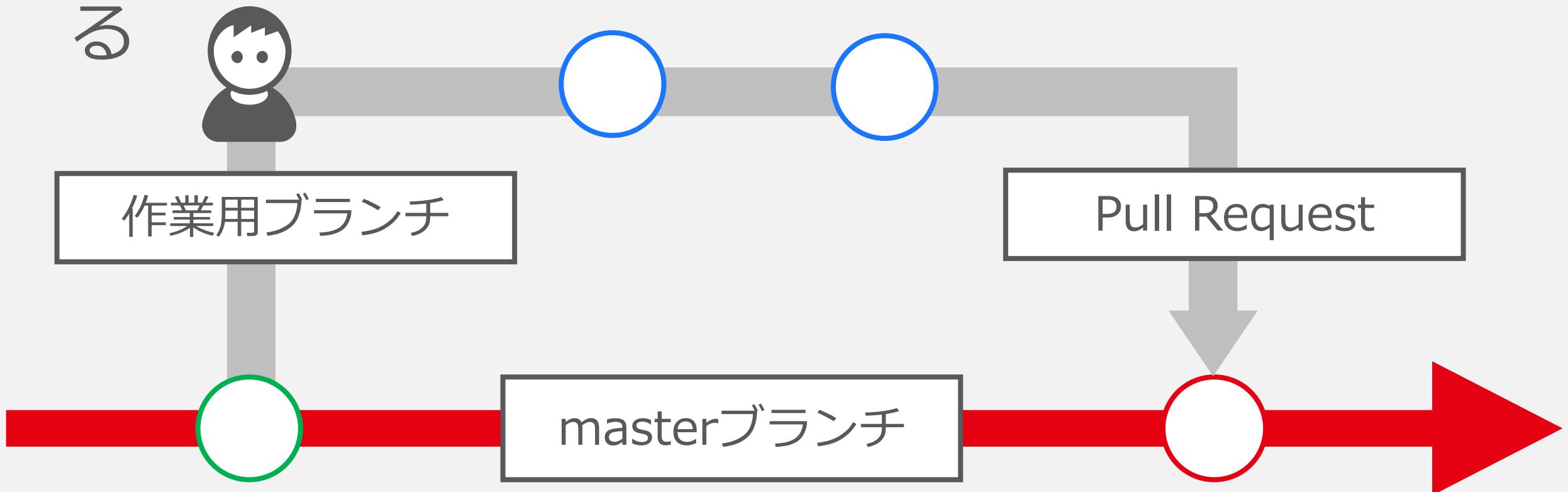
マージ

ブランチの変更を別のブランチに反映する



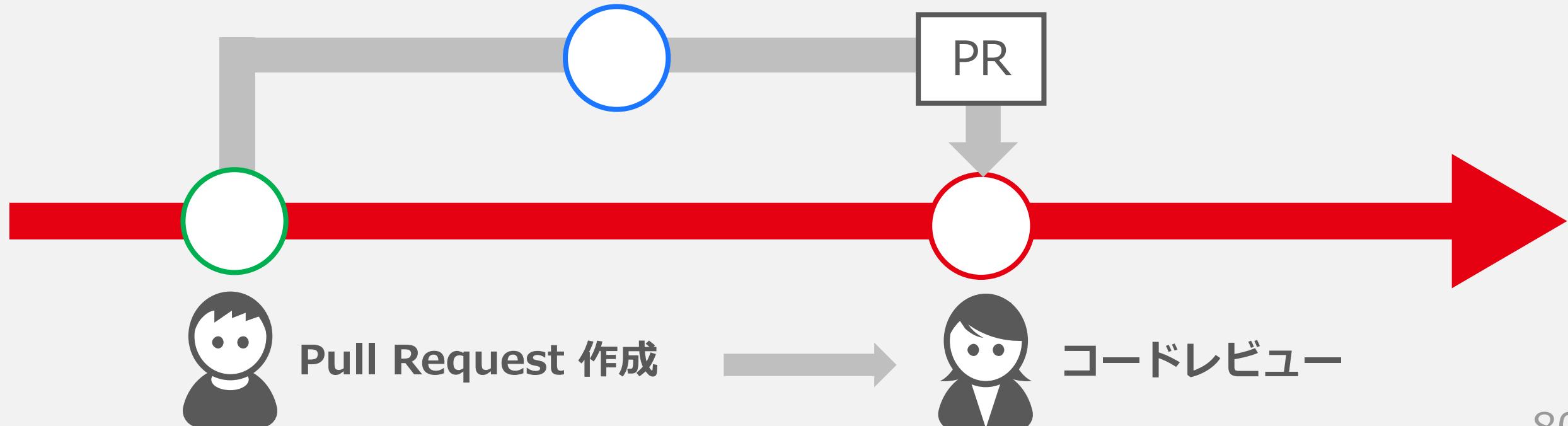
マージ

マージするために Pull Request (PR) を作成する



マージ

- **Pull Request** を作成する
 - GitHub の機能
 - マージしてOKかコードレビューをする



Pull Request の作成



Pull Request 作成してマージする

- GitHub で Pull Request を作成
 - 追加/変更した内容を記載
- Pull Request をマージ
 - 変更内容を確認して問題なければ **Confirm merge**
 - リモートリポジトリの master が更新されているのを確認

Pull Request を作成方法

ユーザーネーム had recent pushes less than a minute ago

Compare & pull request

master ▾ 2 branches 0 tags

Go to file Add file ▾ Code ▾

ユーザーネーム Add R

src

README.md

index.html

README.md

GitHub にいくとこのような画面が出てきます (青枠注文)

※もし出でない場合は Pull requests のページから作成できます

Pull Request を作成方法

Code Issues Pull requests Actions Projects Wiki Security

ユーザーネーム had recent pushes less than a minute ago Compare & pull request

master ▾ 2 branches 0 tags Go to file Add file ▾ Code ▾

ユーザーネーム Add REA/ Compare & pull request を押す

File	Commit Message	Time
src	Add index.html & src	1 hour ago
README.md	Add README.md	39 minutes ago
index.html	Add README.md	39 minutes ago

README.md

Pull Request の作成方法

Pull Request の作成方法

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

The screenshot shows the GitHub pull request creation interface. At the top, there are dropdown menus for 'base: master' and 'compare: ブランチ名'. A green checkmark indicates 'Able to merge. These branches can be automatically merged.' Below this, a large blue callout box contains Japanese text explaining the parameters:

- master ← 作成したブランチ名**
- base:** 変更を反映するブランチ
- compare:** 変更作業を行ったブランチ

The interface includes fields for 'PR タイトル' (Title), 'Write' (content area), 'Reviewers', 'Projects' (None yet), 'Milestone' (No milestone), and 'Linked issues'. A 'Create pull request' button is at the bottom.

① Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

Pull Request の作成方法

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

The screenshot shows the GitHub interface for creating a pull request. At the top, it says "base: master" and "compare: ブランチ名". A green checkmark indicates "Able to merge. These branches can be automatically merged." Below this, there's a title input field containing "PR タイトル", which is highlighted with a blue border. To the left of the title field is a red octocat icon. To the right are sections for "Reviewers" (No reviews), "Assignee" (Assign yourself), "Projects" (None yet), "Milestone" (No milestone), and "Linked issues". A large blue callout box is overlaid on the interface, containing the Japanese text "Pull Request のタイトル" and "→ 概要がひと目でわかるように", explaining the purpose of the title field.

PR タイトル

Pull Request のタイトル
→ 概要がひと目でわかるように

base: master ▾ ← compare: ブランチ名 ▾ ✓ Able to merge. These branches can be automatically merged.

Reviewers

No reviews

Assignee

Assign yourself

Projects

None yet

Milestone

No milestone

Linked issues

Attach files by dragging & dropping, selecting or pasting them.

Create pull request

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

Pull Request の作成方法

Open a pull request

Create a new pull request

Pull Request の概要

→ どのような変更をしたか説明を記載

Write

Preview



変更内容を記載

Attach files by dragging & dropping, selecting or pasting them.

Create pull request

No reviews

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Linked issues

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).



Pull Request の作成方法

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

The screenshot shows the GitHub interface for creating a pull request. At the top, it says "base: master" and "compare: ブランチ名". A green checkmark indicates "Able to merge. These branches can be automatically merged." Below this, there's a title field labeled "PR タイトル" and a rich text editor toolbar with "Write" and "Preview" tabs. The main body area has a placeholder "変更内容を記載". To the right, there are sections for "Reviewers" (No reviews), "Assignees" (No one—assign yourself), "Labels" (None yet), "Milestone" (No milestone), and "Linked issues". A large blue button at the bottom with the text "タイトルと概要入力したらクリック" (Click after entering title and summary) covers the "Create pull request" button. The "Create pull request" button itself is highlighted with a red box.

タイトルと概要入力したらクリック

Attach files by dragging & dropping, selecting or pasting them.

Create pull request

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

Copyright (C) 2020 Yahoo Japan Corporation. All Rights Reserved.

Pull Request の作成方法

The screenshot shows a GitHub Pull Request page with the following details:

- PR タイトル #1**
- Open** button
- ユーザーネーム wants to merge 1 commit into **master** from ユーザーネーム
- Conversation 0**, **Commits 1**, **Checks 0**, **Files changed 2**
- ユーザーネーム commented now**: 变更内容を記載
- Update index.html** commit by ユーザーネーム (78149a0)
- Add more commits by pushing to the ユーザーネーム branch on ユーザーネーム / github_ws.
- Continuous integration has not been set up**: GitHub Actions and several other apps can be used to automatically catch bugs and enforce style.
- This branch has no conflicts with the base branch**: Merging can be performed automatically.
- Merge pull request** button
- You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

Pull Request の作成方法

PR タイトル #1

Open ykato524 wants to merge 1 commit into master from ykato524

Conversation 0 Commits 1 Checks 0 Files changed 2 +4 -1 Review changes

Changes from all commits ▾ File filter... ▾ Jump to... ▾ 1 / 2 files viewed Review changes

3 README.md

2 index.html

Files changed に変更差分が掲載される

...
11	11	11
12	12	12
13	13	13
14	-	->
14	+	<div id="ok">GitHub完全に理解した</div>
15	15	15
16	16	<script src=".src/index.js"></script>
17	17	</body>

Pull Request の確認

A screenshot of a GitHub Pull Request review interface. At the top, there's a commit history showing a single commit: "Update index.html" with hash "78149a0". Below the commit, a message says "Add more commits by pushing to the ユーザー名 branch on ユーザー名 / github_ws." A green box highlights two status messages: "Continuous integration has not been set up" (with a note about GitHub Actions and other apps) and "This branch has no conflicts with the base branch" (with a note that merging can be performed automatically). A "Merge pull request" button is visible. Below this, a comment section is shown with a text area containing "レビューや気になった内容はコメントできる" (Review or comment on anything you like) and a "Comment" button.

Update index.html
78149a0

Add more commits by pushing to the ユーザー名 branch on ユーザー名 / github_ws.

Continuous integration has not been set up
GitHub Actions and several other apps can be used to automatically catch bugs and enforce style.

This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request

You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

Write Preview

レビューや気になった内容はコメントできる

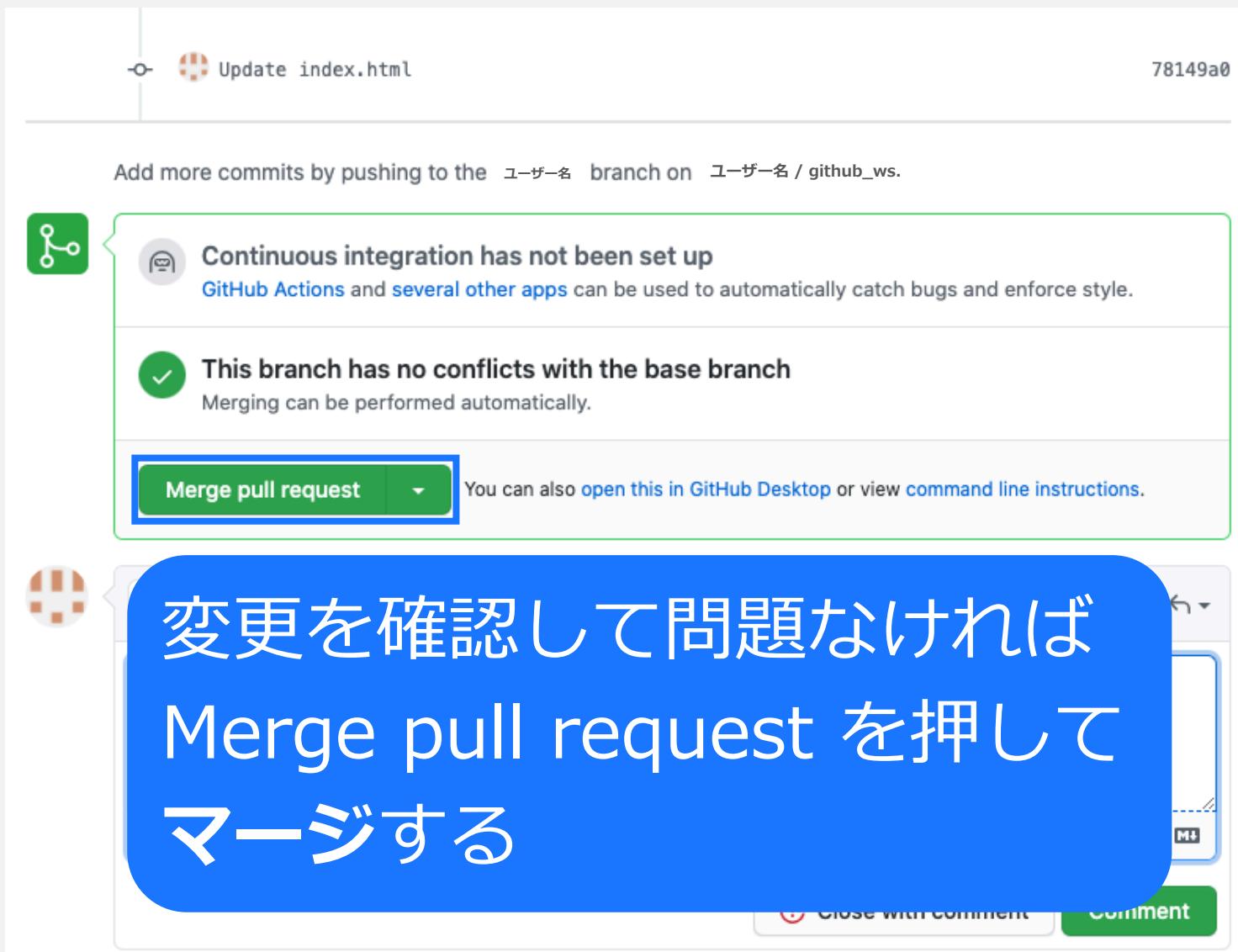
Attach files by dragging & dropping, selecting or pasting them.

Close with comment Comment

Pull Request の確認



Pull Request をマージ



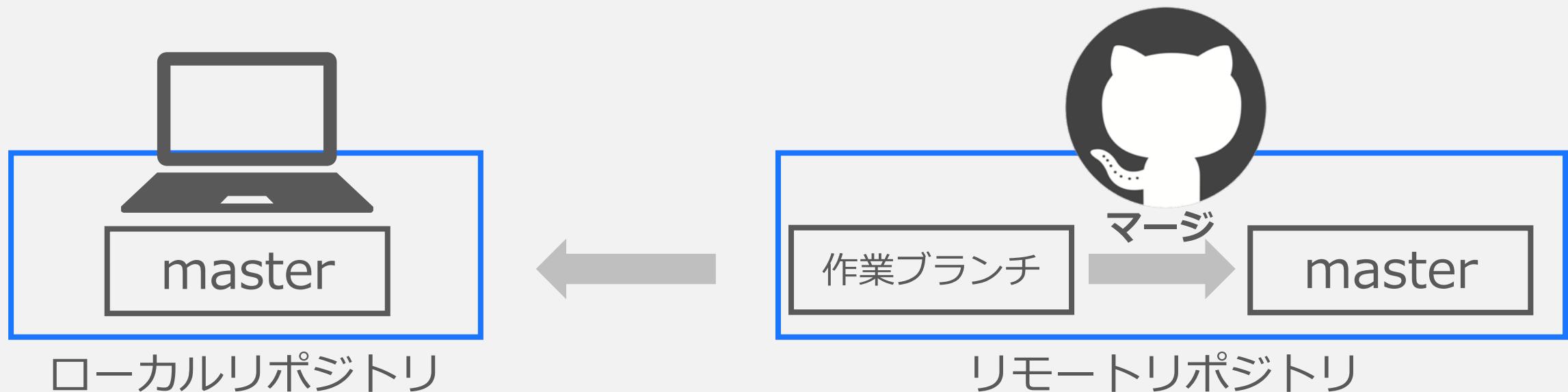
ローカルリポジトリに
pull する

YAHOO!
JAPAN

マージされたものをローカルに反映

Pull Request がマージされると

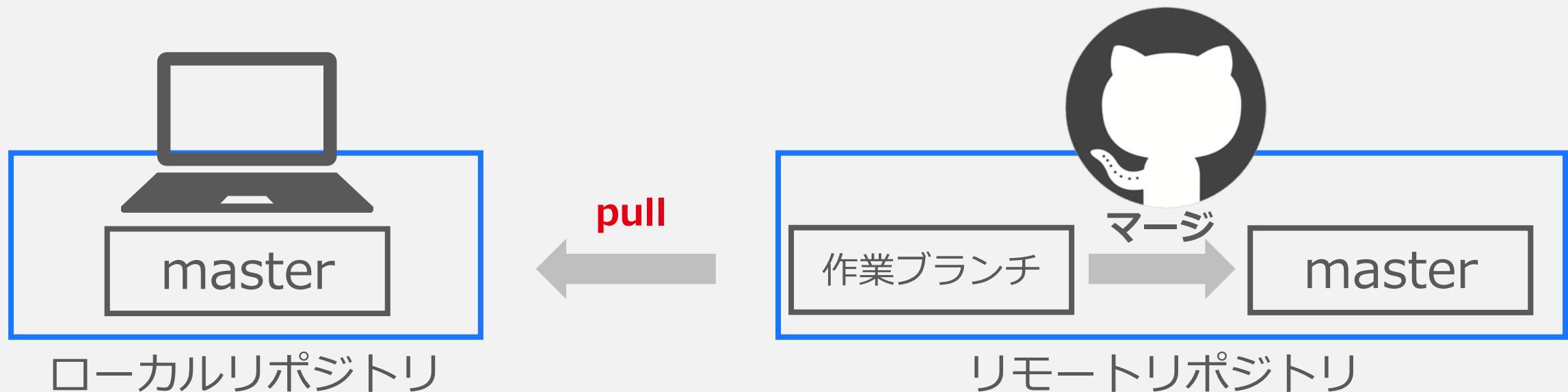
- ・ リモートの master には反映される
- ・ ローカルの master には反映されていない



マージされたものをローカルに反映

リモートの master の内容をローカルの master に反映

→ ローカルに **pull** する

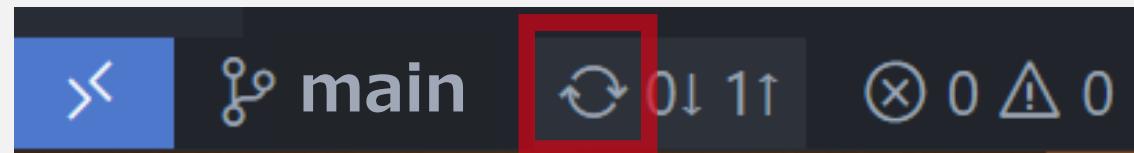


VSCodeで
pullする



ローカルにpullする

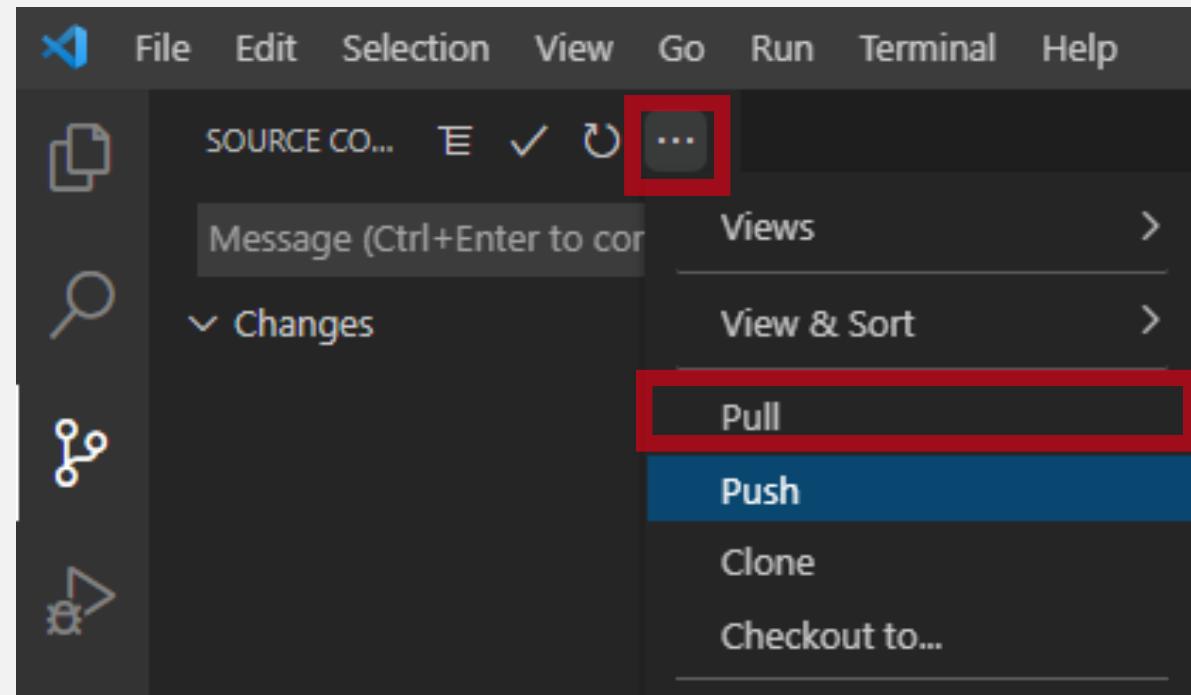
1. 画面下のブランチ名をクリックして、mainをクリックしてブランチを移動
2. 「1↓0↑」のように表示されているとき更新ボタンを押すとpullされます
(状況により表示されない場合もあります)



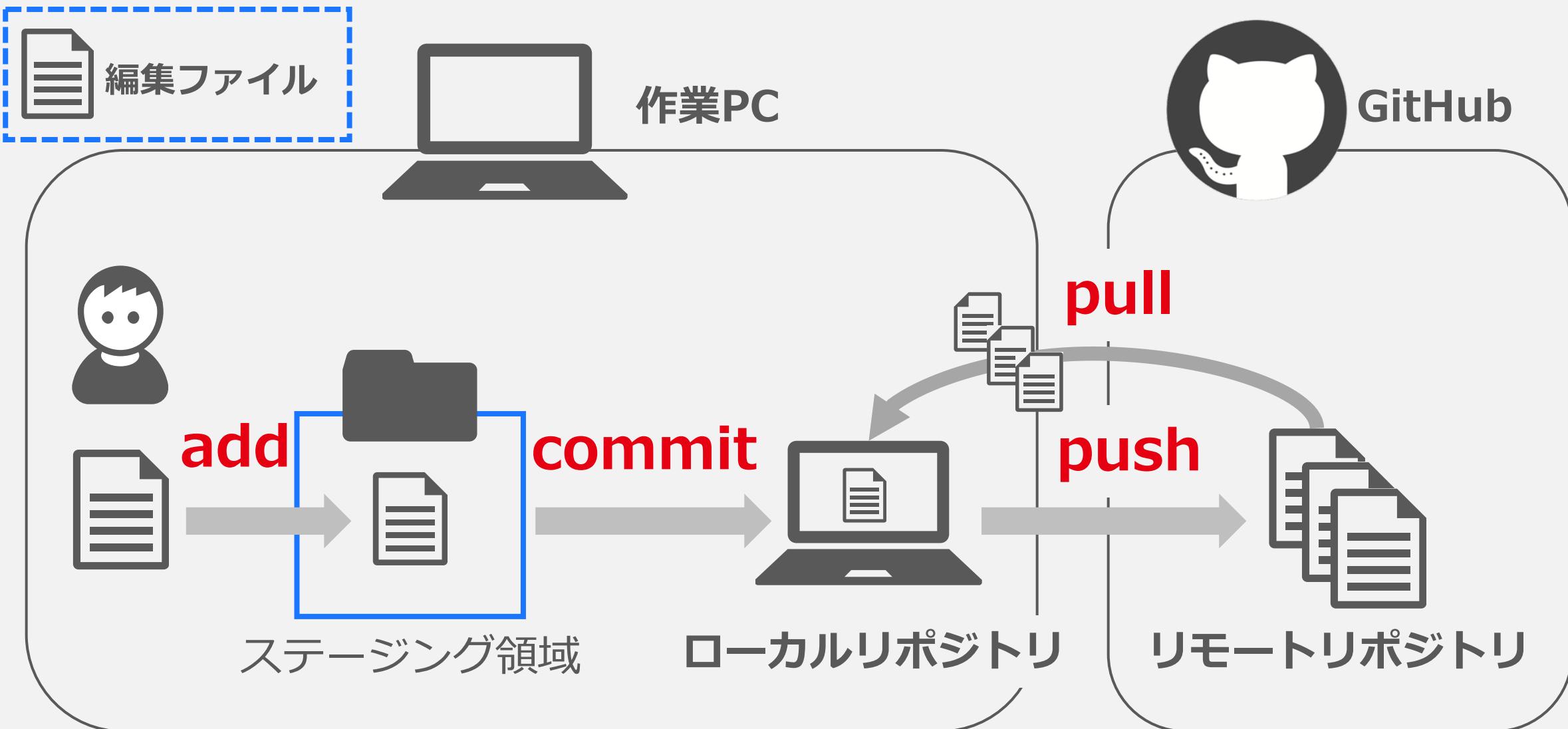
3. ローカルファイルが更新されていくことを確認

ローカルにpullする

- 更新ボタンでのpushがうまく行かないときは、下図の「…」から「Pull」を選択してください

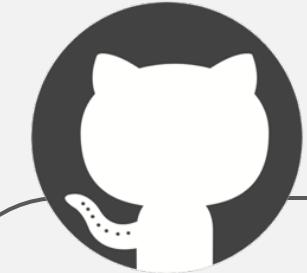


ローカルリポジトリの更新



ローカルリポジトリの更新

clone : リモートをローカルにコピー
pull : リモートの変更をローカルに反映



GitHub



Step.2

ハンズオン

ハンズオンの作業内容 その1

- 新たにブランチを作成し、ファイルをpush
 1. 別のブランチを作成
 2. ファイルを編集して作成したブランチにpushしてみよう
 3. GitHubで作成したブランチとpushしたファイルを確認

ハンズオンの作業内容 その2

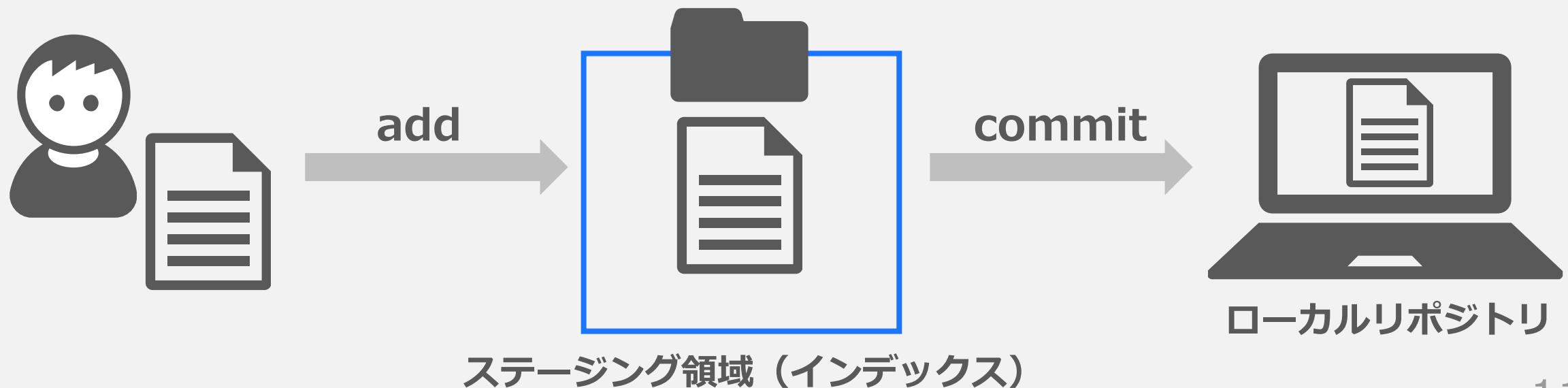
- Pull Requestを作成してmain反に映
 - 1. Pull Requestを作成
 - 変更点をコメントしよう
 - 他人のPRにコメントしよう
 - 2. Pull Requestをマージ
 - 3. ローカルのmainにpull
 - 変更が反映されてるかファイル内容を確認しよう

おさらい

YAHOO!
JAPAN

【おさらい】ローカルリポジトリの作業

- ・ ファイルの変更は **add** して **commit**
- ・ addするファイルの内容に注意
- ・ commitしたあとは**必ずコメントを残す**

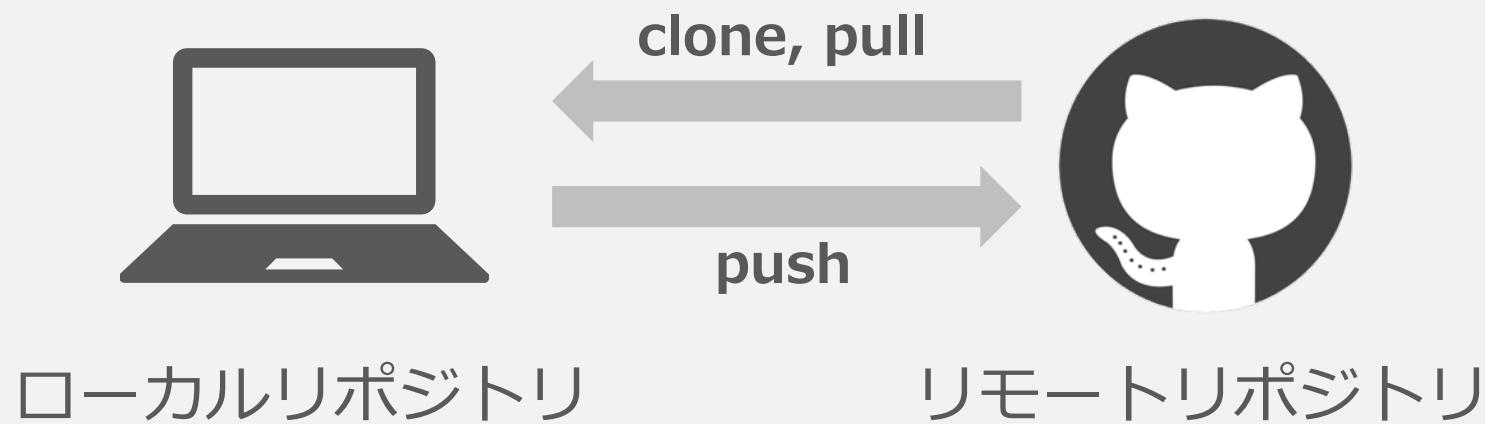


【おさらい】ローカルとリモートの関係

clone : リモートをローカルにコピー

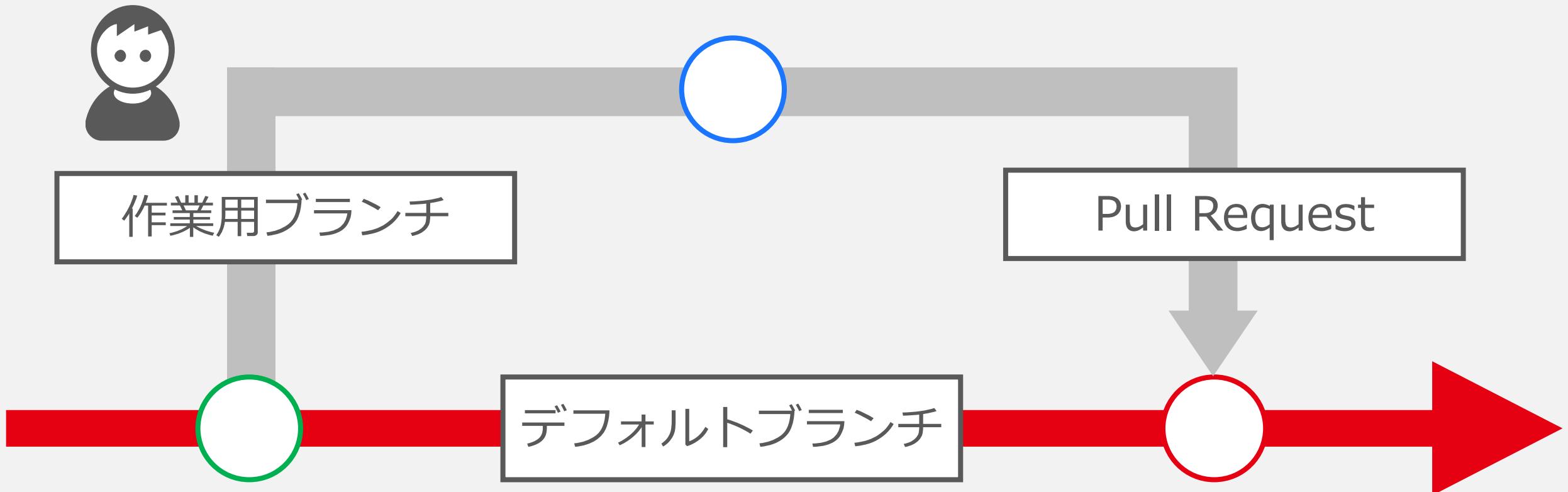
pull : リモートの変更をローカルに反映

push : ローカルの変更をリモートに反映



【おさらい】 ブランチ

作業ブランチを作つて Pull Request をする



まとめ

- ・ この資料で扱った機能は基礎中の基礎
- ・ Git と GitHub にはこれ以外にも便利な機能があるので、必要な場合は検索してみましょう。

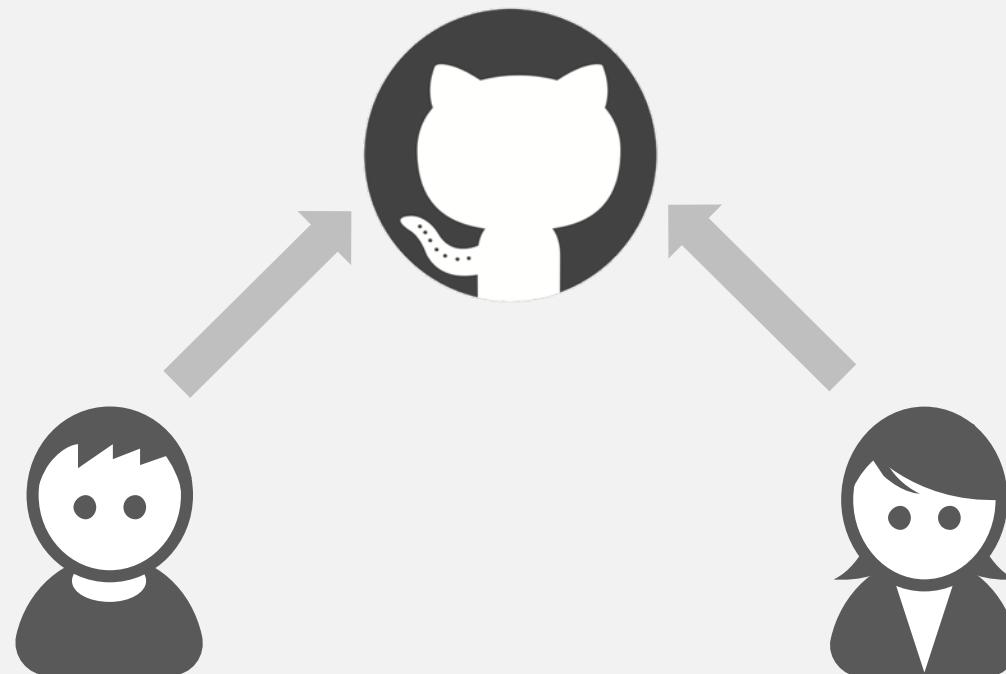
まずは基本的な使い方に慣れる。
使っていく中で調べながら使う。

GitHubを複数人で使う



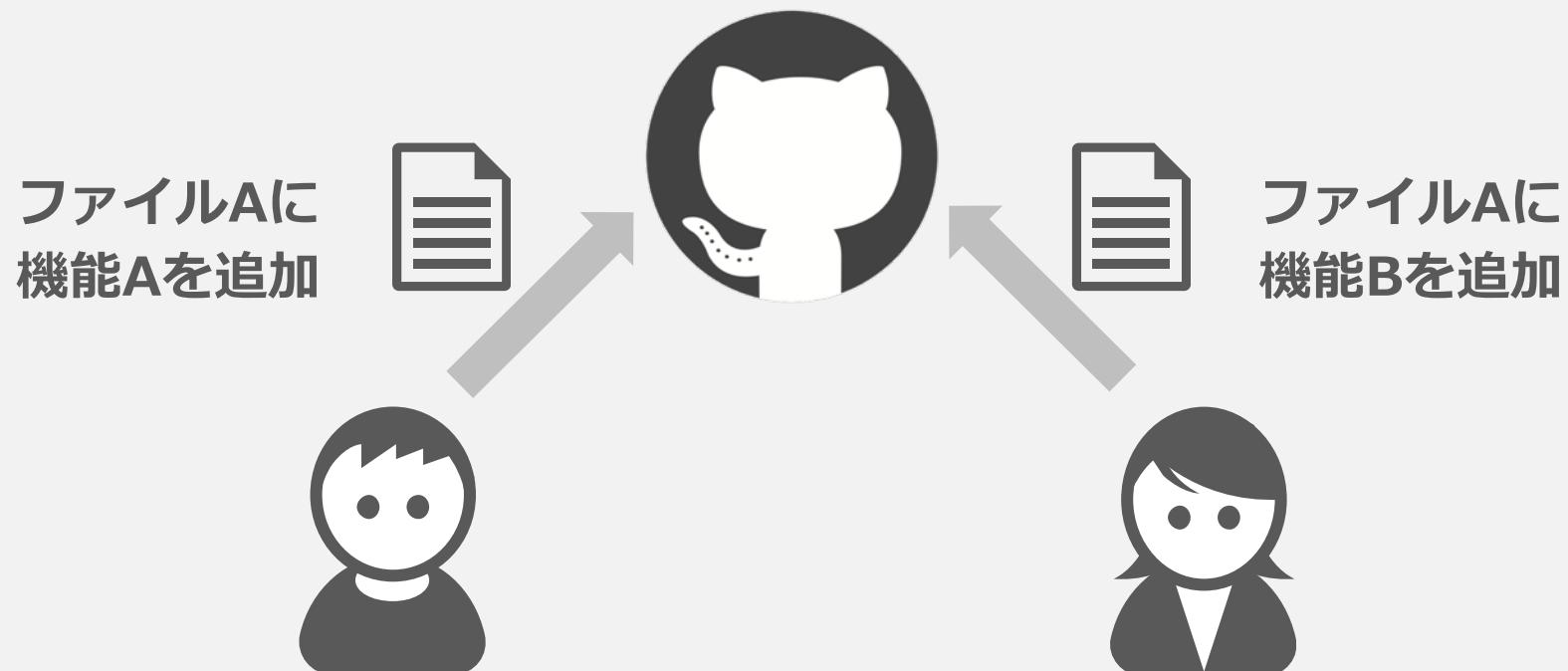
チーム開発では

- 1つのリポジトリに対して複数人で作業する



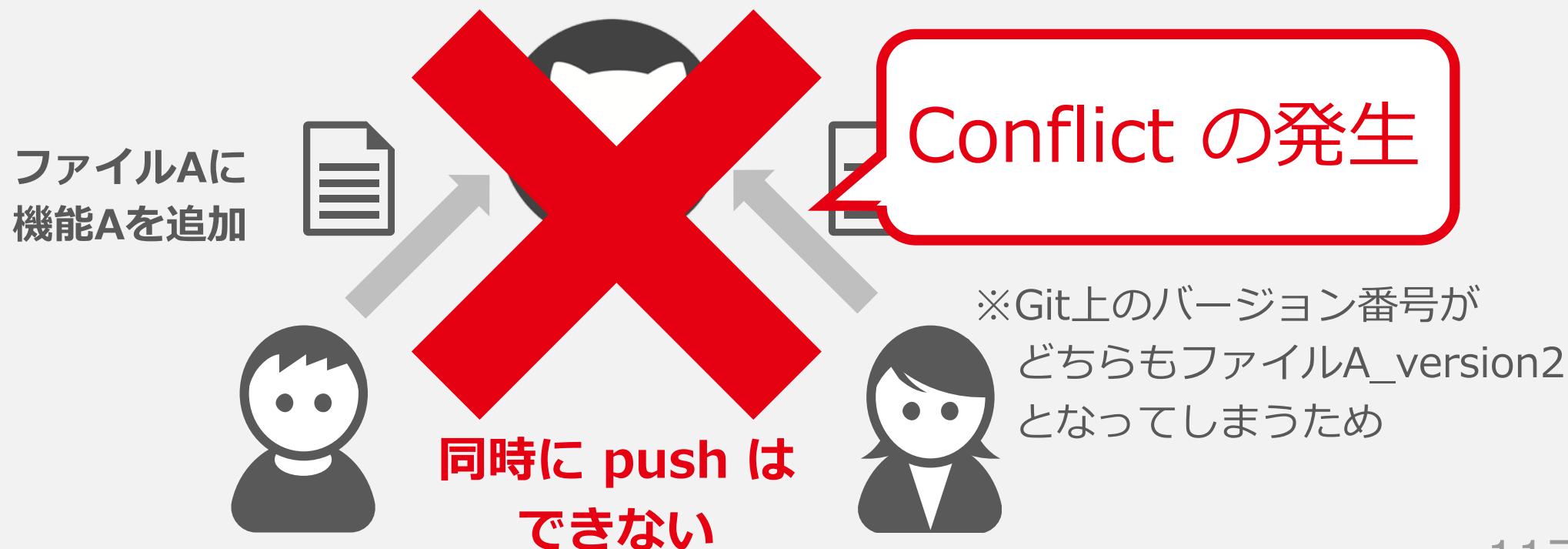
チーム開発では

- ・ 1つのリポジトリに対して複数人で作業する
- ・ 同じファイルを同時に修正してしまう可能性も



チーム開発では

- ・ 1つのリポジトリに対して複数人で作業する
- ・ 同じファイルを同時に修正してしまう可能性も



チーム開発では

- ・ 1つのリポジトリに対して複数人で作業する
- ・ 同じコードを同時に修正してしまう可能性もある

Conflict を避けるため
作業者ごとに
ブランチを分けましょう



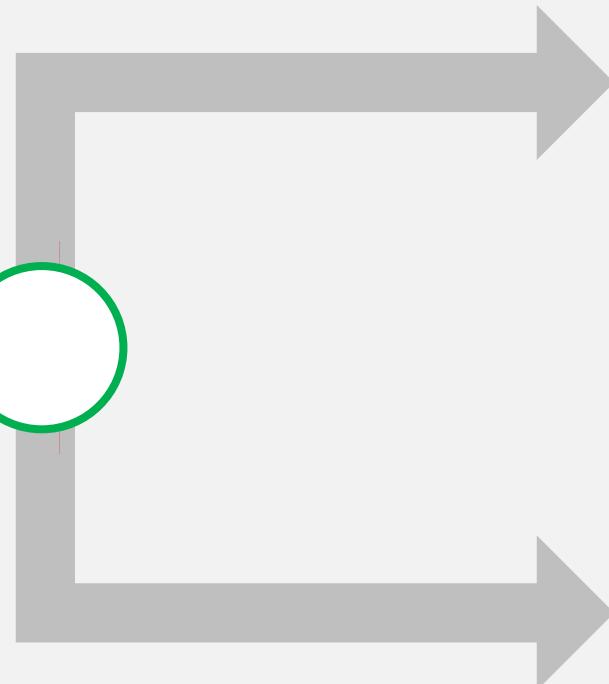
同時に push は
できない



チーム開発では



Aさんの作業ブランチ



Bさんの作業ブランチ

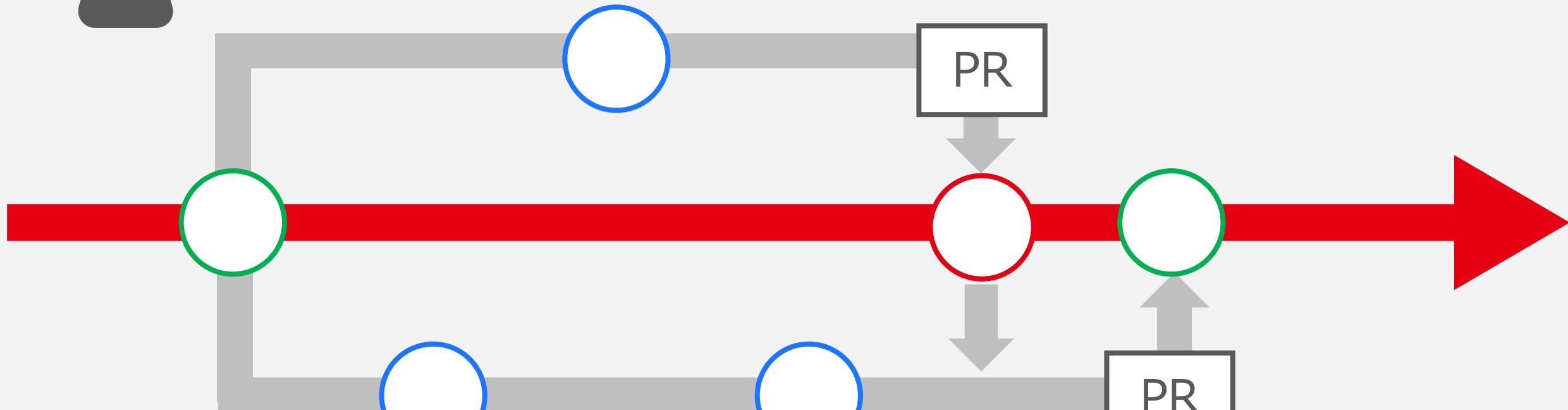
作業者がそれぞれブランチを作成

チーム開発では



Aさんの作業ブランチ「A」

作業者がそれぞれブランチを作成
→ PRでコードレビューをしてマージ



Bさんの作業ブランチ「B」

チーム開発では



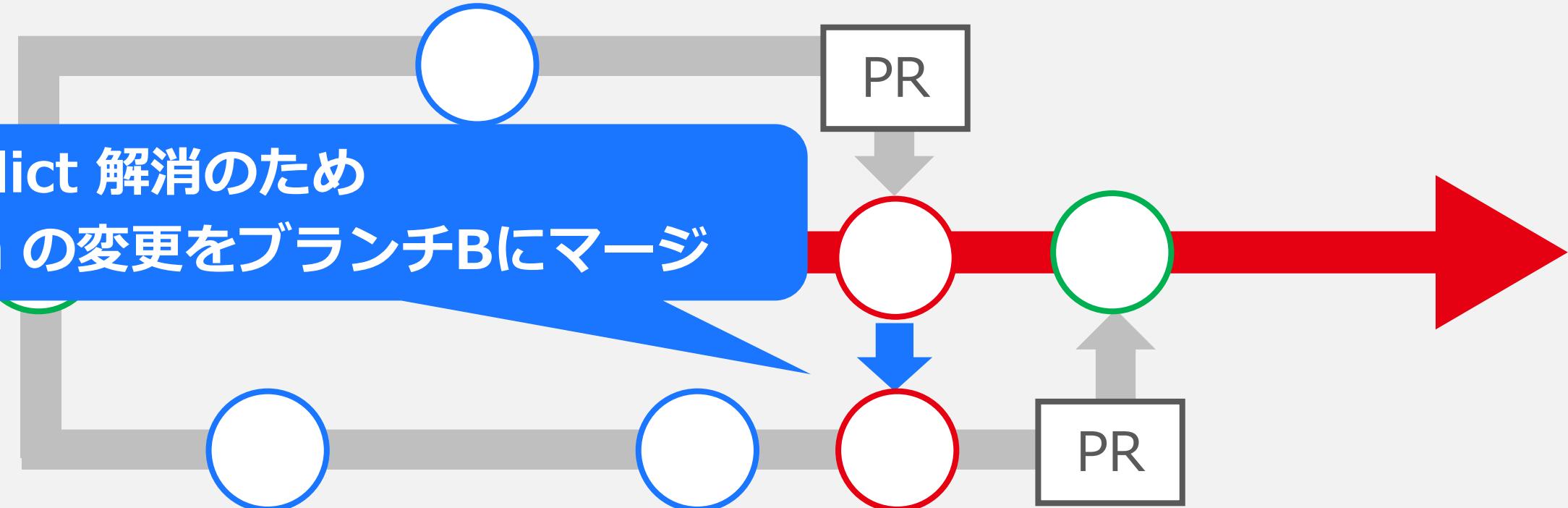
Aさんの作業ブランチ「A」

作業者がそれぞれブランチを作成
→ PRでコードレビューをしてマージ

Conflict 解消のため
main の変更をブランチBにマージ



Bさんの作業ブランチ「B」



GitHub入門編

以上



おまけ

- **Hack U をフォローしよう**

1. <https://github.com/hackujp> にアクセス
2. hackujp を Follow
3. hackujp/github_tutorial を Watch & Star

※どちらも任意です

最新情報は **connpass/Twitter(@hackujp)** をフォロー

Appendix



Gitコマンドでの操作



Gitコマンドでの操作

- add

```
$ git add ファイルパス
```

- commit

```
$ git commit -m "commitコメント"
```

Gitコマンドでの操作

- ブランチ作成
\$ git branch ブランチ名
- 今のブランチを確認
\$ git branch
- ブランチの切り替え
\$ git checkout ブランチ名
- ブランチの作成 + 切り替え
\$ git checkout -b ブランチ名

Gitコマンドでの操作

- clone

```
$ git clone リポジトリURL
```

- pull

```
$ git pull origin pullしたいブランチ名
```

- push

```
$ git push origin pushしたいブランチ名
```

Gitコマンドでの操作

- “origin” とは？
 - リモートリポジトリのURLを指している
 - 毎回打つのが面倒なので“origin”という略称で代用できるように設定されている

例) これらは同じ内容を表す

```
$ git pull origin master
```

```
$ git pull git@github.com:ユーザ名/リポジトリ名.git master
```