

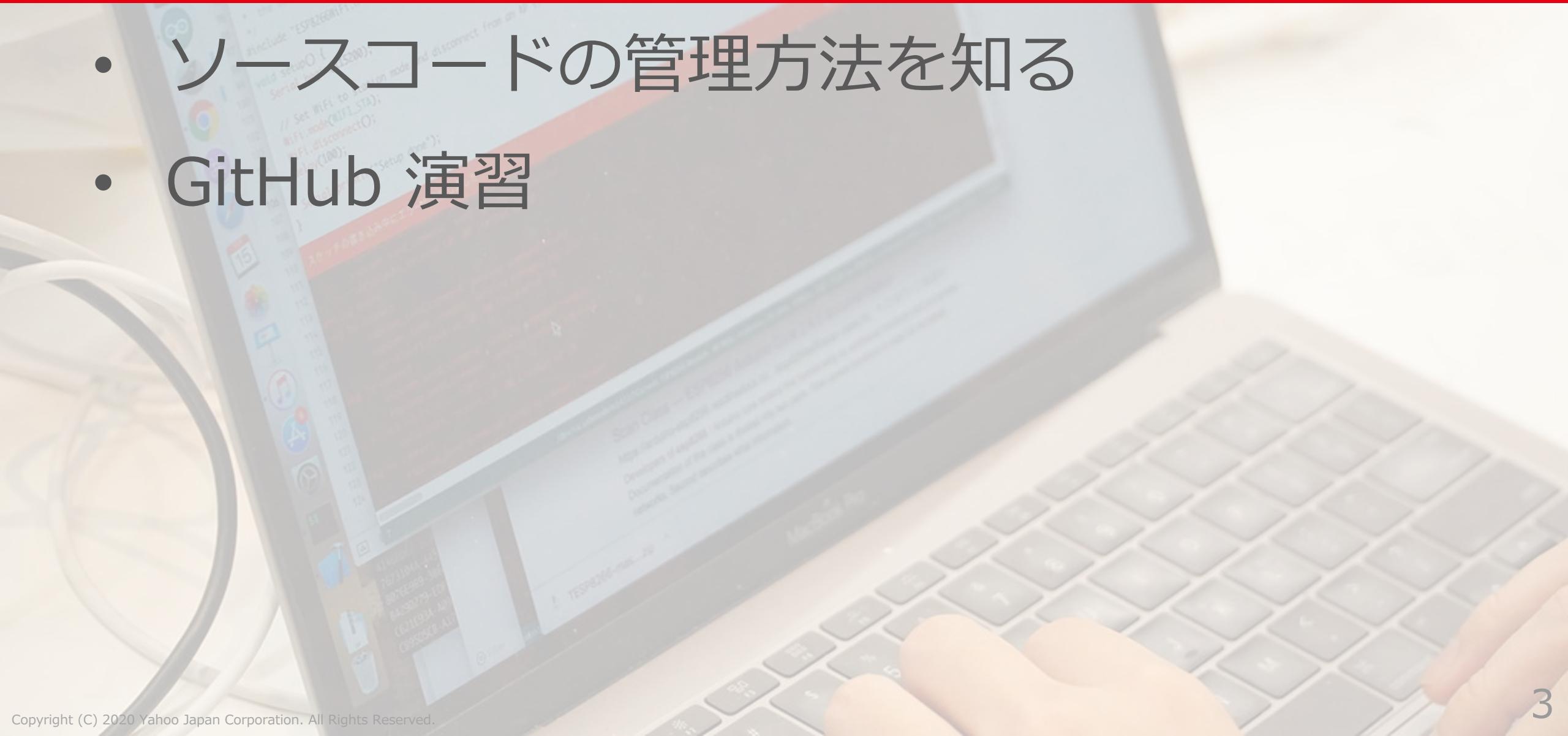
GitHub 入門編

CUI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. The commit history shows a single commit from 14 minutes ago:

- Commit message: 'Add index.html & src'
- Author: 'ユーザー名'
- Timestamp: '50cdc45 14 minutes ago'
- Commits: '1 commits'
- Files added:
 - 'src' (type: folder)
 - 'index.html' (type: file)
- Timestamp for files: '14 minutes ago'

A large blue callout box contains the text:

add/commit/push を行い
演習ファイルが上がっていればOK

Below the commit history, there is a message: "Help people interested in this repository understand your project by adding a README." with a "Add a README" button.

On the right side of the repository page, there are sections for "topics provided." and "Releases".

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 :**

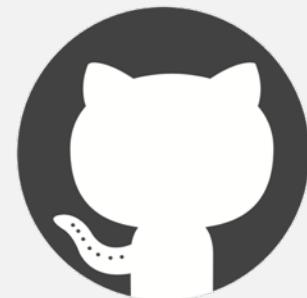
色々な人が共有できるようにしたリポジトリ
GitHub上にあるものはリモートリポジトリ

- **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

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

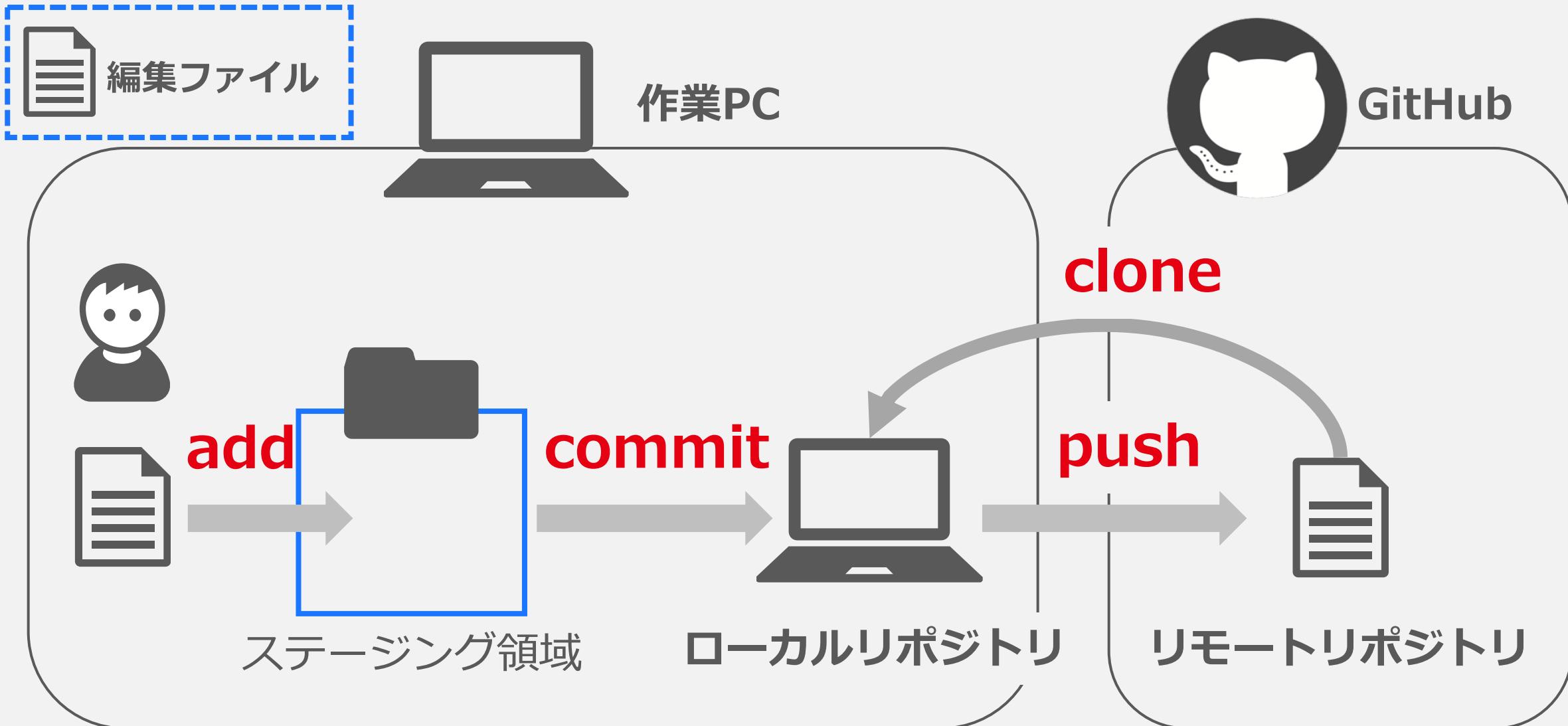
今回は コマンドラインで
Gitコマンド を使います

Step.1

GitHub の基本操作



GitHub の基本操作

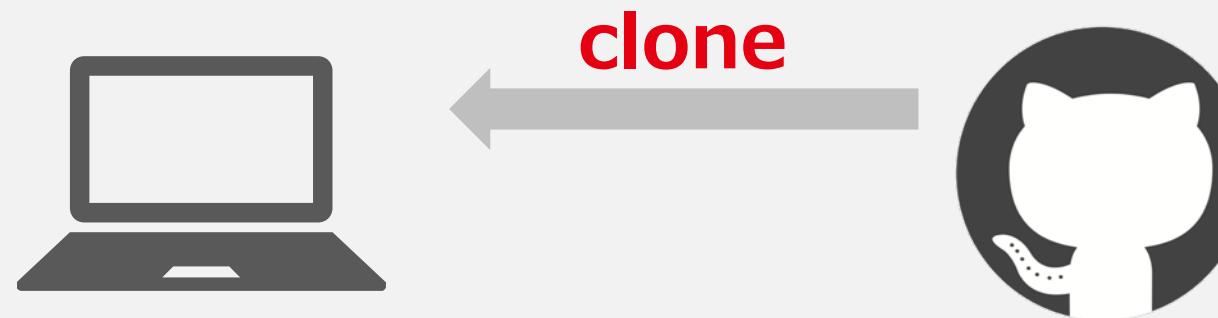


GitHub から自分のPCに持ってくる

- **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

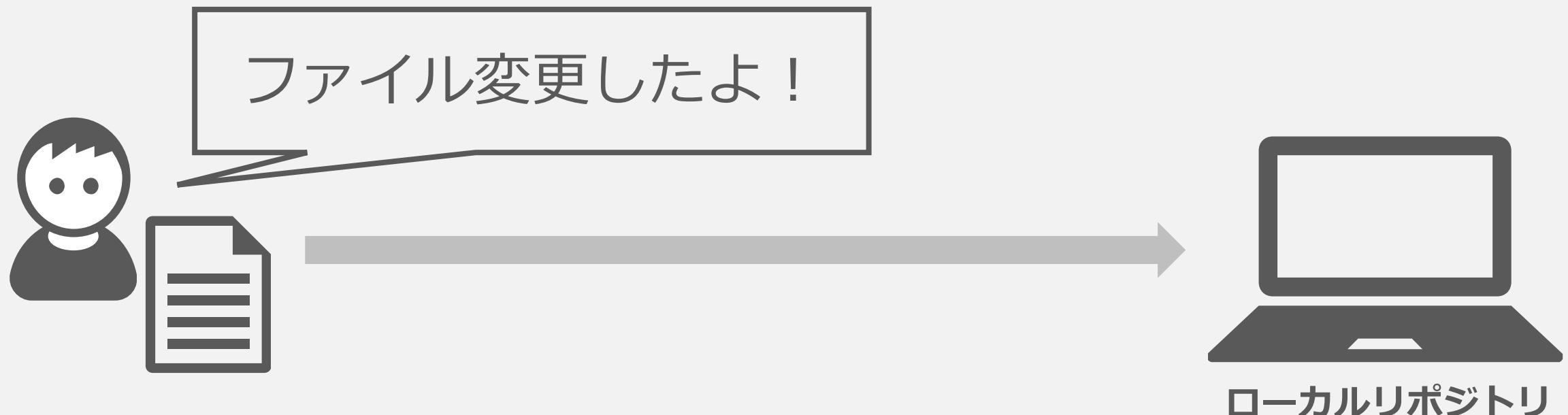


ローカルリポジトリ

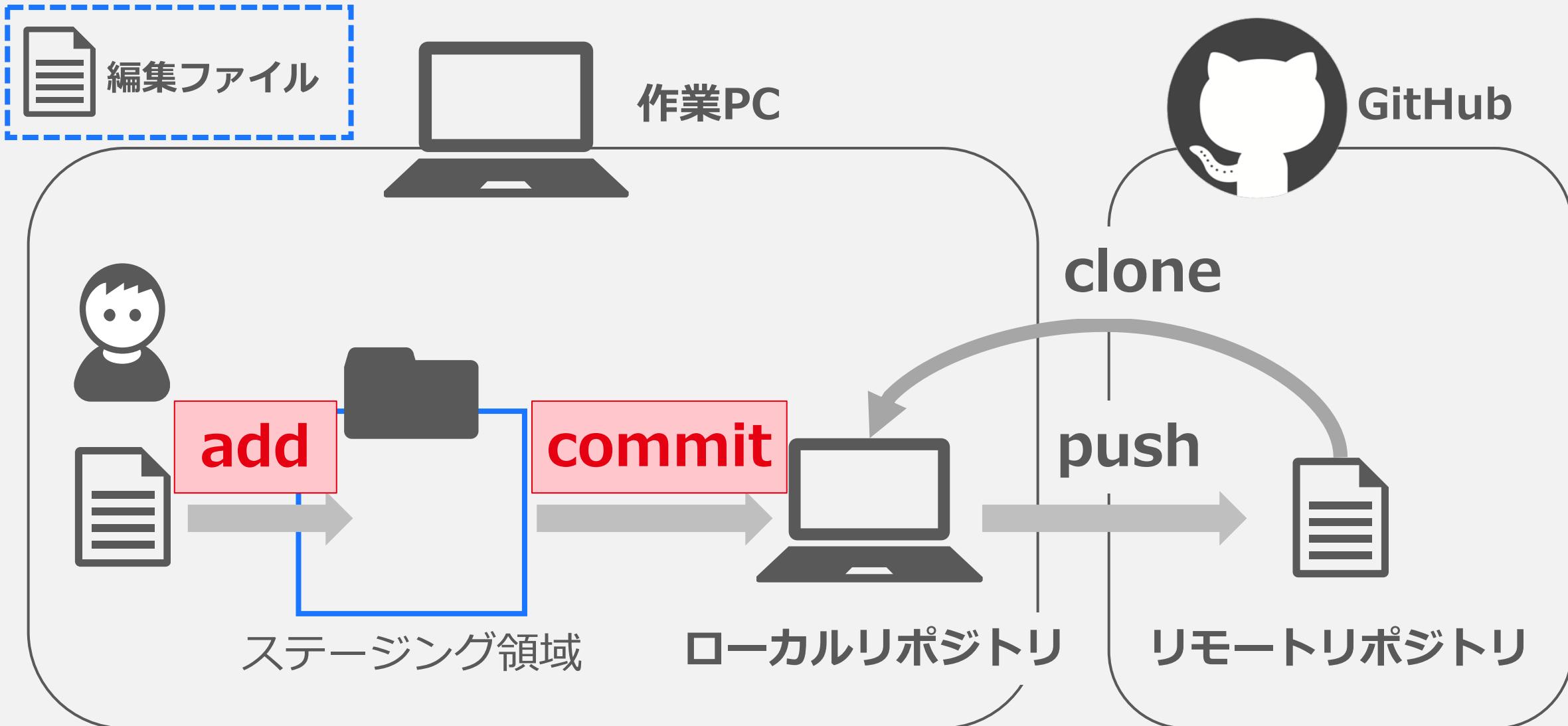
リモートリポジトリ

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



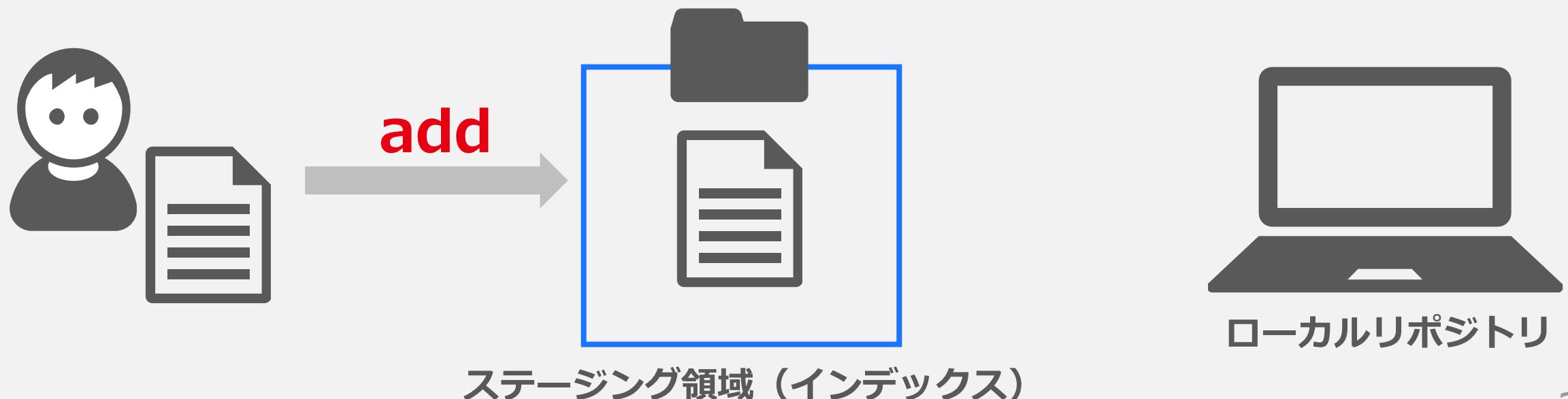
GitHub の基本操作



ファイルを修正してGitで扱う

Step.1

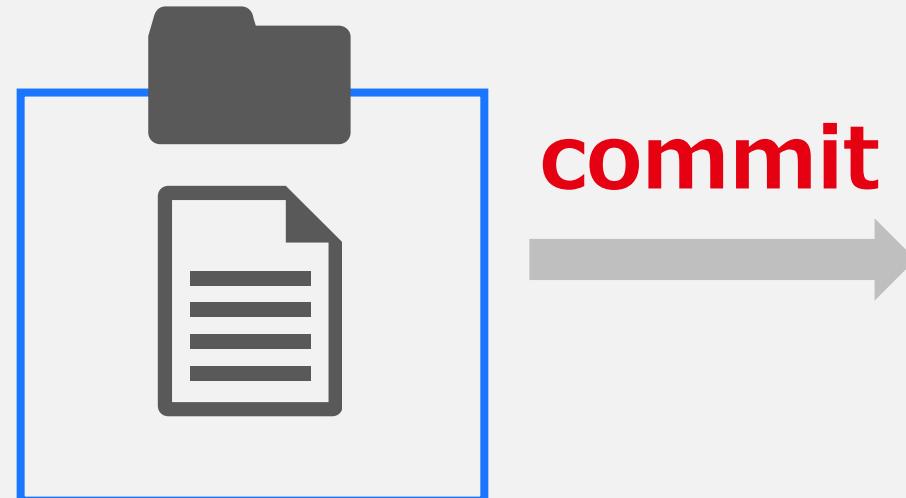
どのファイルの変更をしたかをステージング領域に一度登録する → **add** という



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

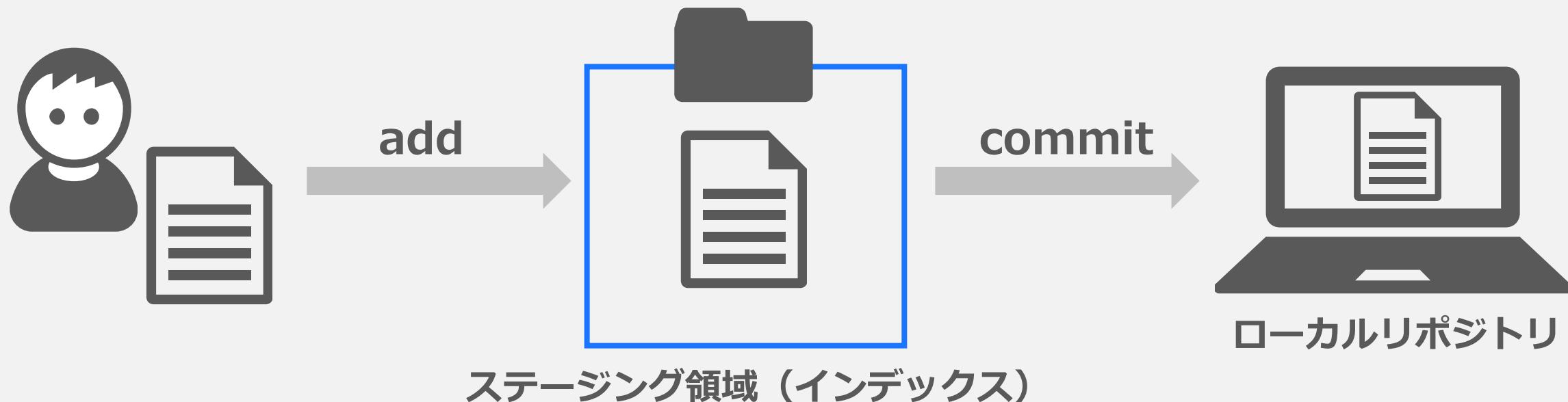


ステージング領域（インデックス）

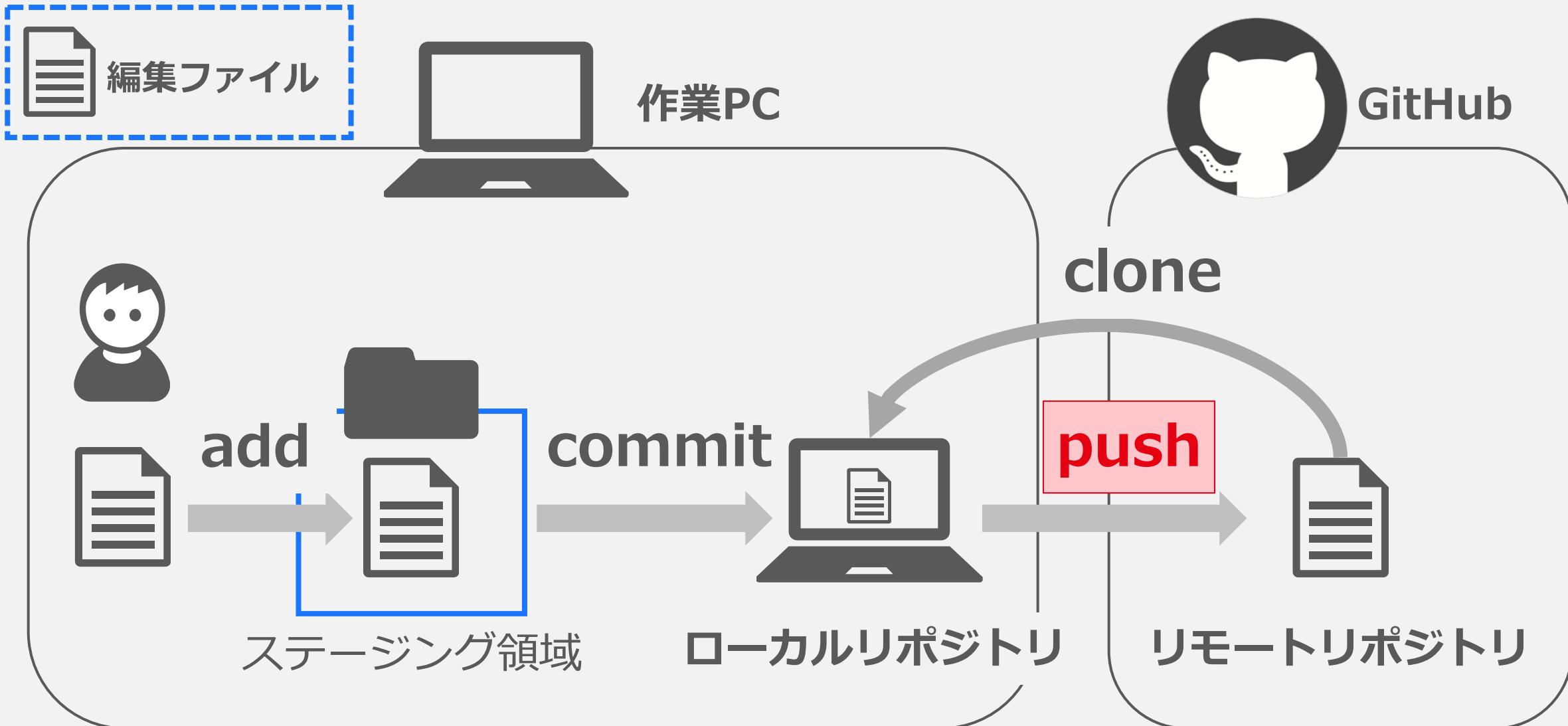


ファイルを修正してGitで扱う

- 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

Git CUIで
clone する

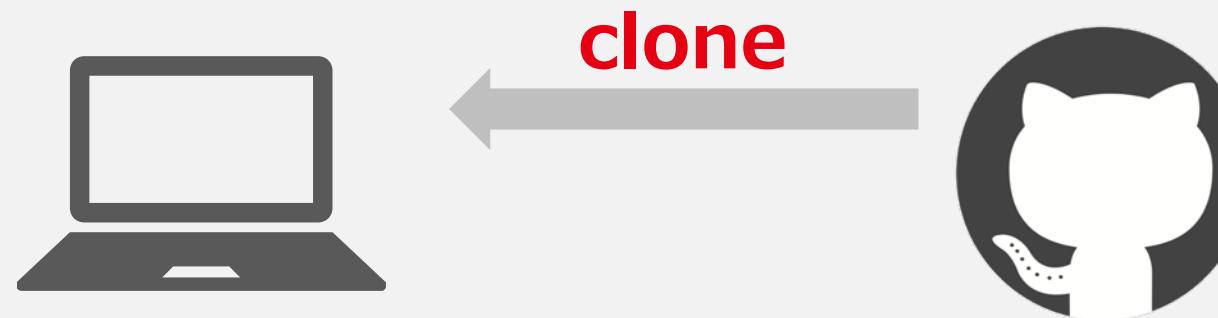


GitHub から自分のPCに持ってくる

- **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

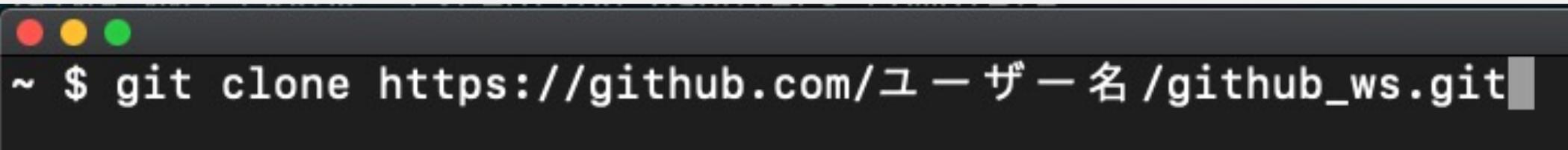


ローカルリポジトリ

リモートリポジトリ

clone

- ・ 適当なディレクトリに移動して、← この場所がローカルリポジトリ
- ・ 「git clone https://github.com/ユーザー名/github_ws.git」



```
~ $ git clone https://github.com/ユーザー名/github_ws.git
```

- ・ 「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

Git CUI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「git status」で変更されたファイルを確認できます。

```
~/github/github_ws $ git status
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)

    index.html
    src/

nothing added to commit but untracked files present (use "git add" to track)
```

ステージングにaddする

- 「git add .」（最後にピリオドが必要です）で addされます。「git status」でファイルごとの変更内容が出来ていることを確認してください。

```
[~/github/github_ws $ git add .
[~/github/github_ws $ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   index.html
    new file:   src/index.js
    new file:   src/styles.css
```

ファイルごとの
変更内容が表示され
ます

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

Git CUI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

- 「git commit -m “<任意のメッセージ>”」
でコミットされます。
- commit後は「git status」すると変更分は消えて
いることが確認できます。

```
[~/github/github_ws $ git commit -m "first commit"
[master d55fbdf] first commit
 3 files changed, 50 insertions(+)
  create mode 100644 index.html
  create mode 100644 src/index.js
  create mode 100644 src/styles.css
```

Git CUI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）

- ・ 「git push origin master」でpushします
(ブランチがmainの場合はmaster -> main。
commit時の表示を確認してください)
- ・ Githubのユーザー名とパスワードを入力する必要
があります。

```
~/github/github_ws $ git push origin master
```

GitHub に反映されているか確認

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

The repository details:

- Code: master branch, 1 branch, 0 tags
- Last commit: 50cdc45, 14 minutes ago, 1 commits
- Commits (from last commit):
 - src: Add index.html & src, 14 minutes ago
 - index.html: Add index.html & src, 14 minutes ago

Repository status:

- Unwatched (1 watch)
- Starred (0 stars)
- Forked (0 forks)

About section:

No description, website, or topics provided.

Releases section:

No releases published

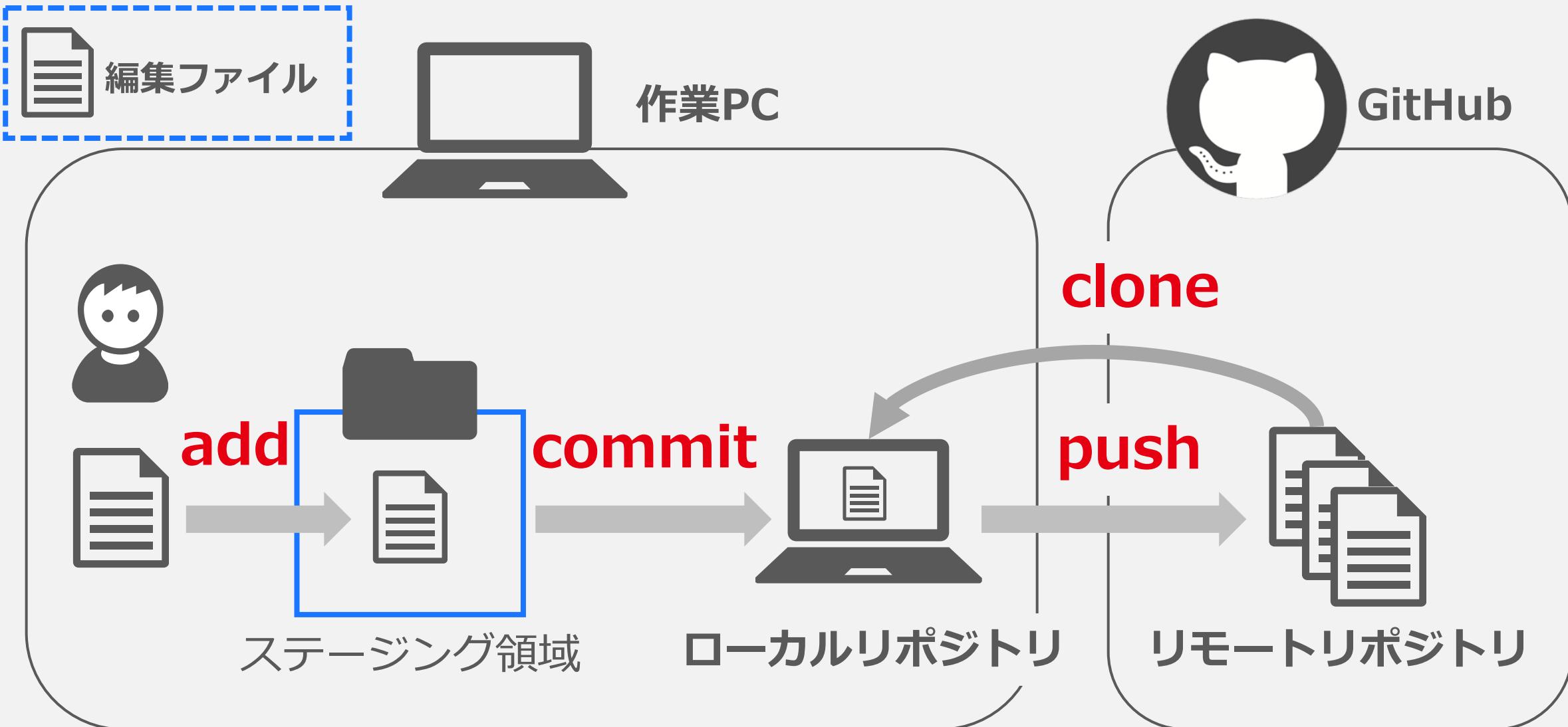
Add a README

Header navigation: Pull requests, Issues, Marketplace, Explore, Notifications, +, Profile.

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 の作成 / index.html を編集する
 - ファイルはメモ帳などのエディタで編集してください
 - 2. 作成/変更したファイルを add する
 - 変更した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1

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
 - コミットツリーを確認してみよう

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
- **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

ファイル差分の確認

```
[~/github/github_ws $ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   README.md
    modified:   index.html
```

新規ファイル(README.md)
更新ファイル(index.html)
add した際の違いを確認

ブレイクアウトルームへ移動します

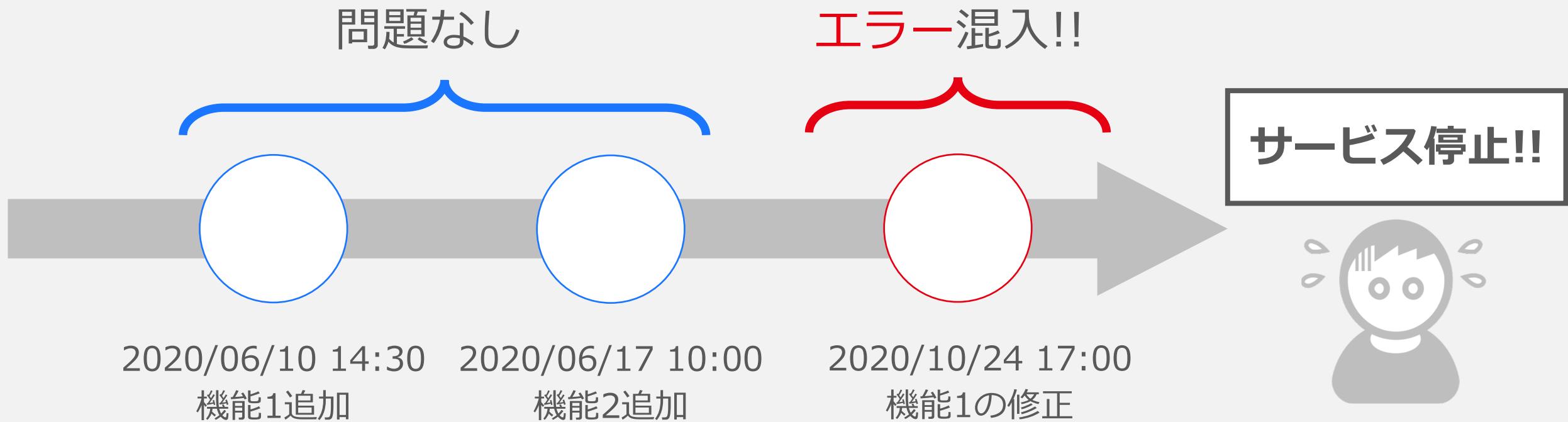


Step.2-1

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

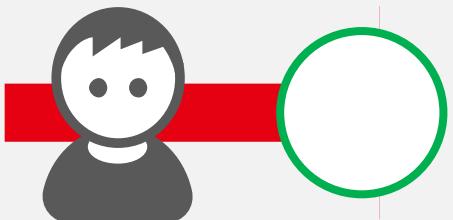
機能2追加

機能1の修正

止!!

ブランチ

最新の状態



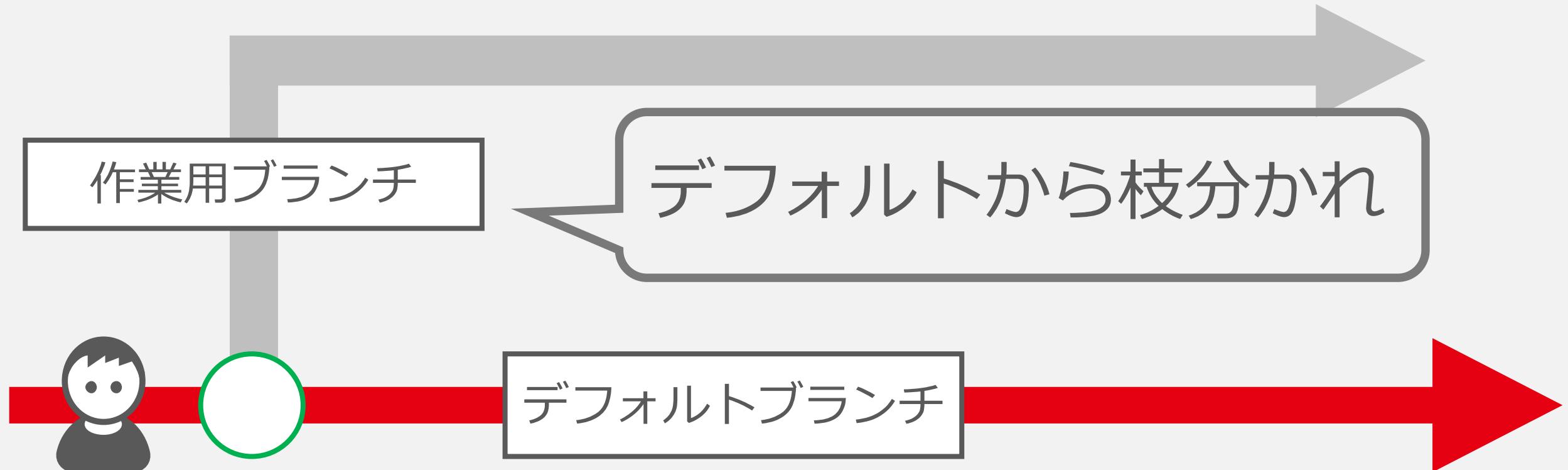
大元のブランチ

→ 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 `master` を使用

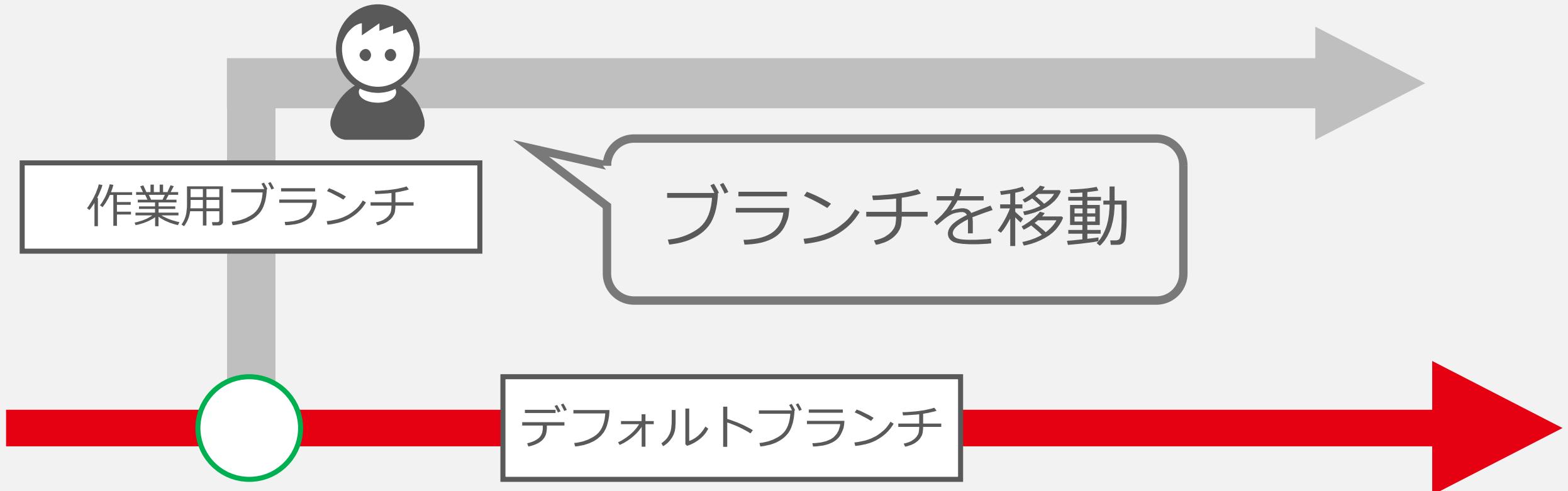
ブランチ

作業用ブランチを作成する



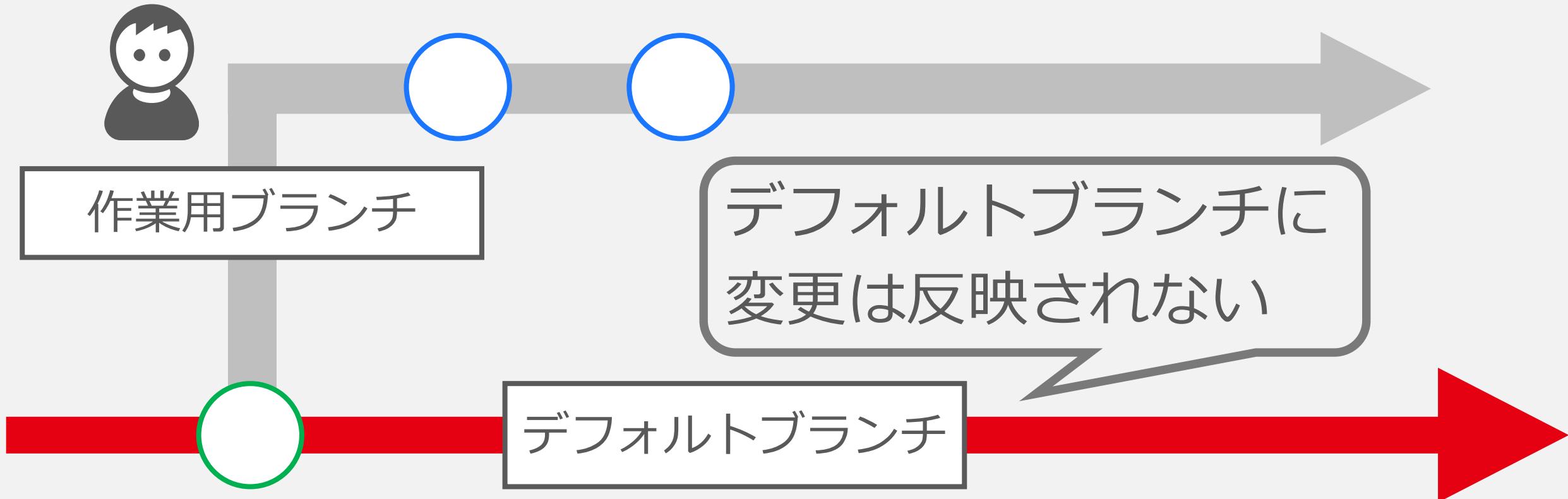
ブランチ

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



Sourcetreeで ブランチを作成する

YAHOO!
JAPAN

ブランチを作成する

- 「git branch <ユーザー名>」でブランチ作成

```
[~/github/github_ws $ git branch ユーザー名  
[~/github/github_ws $ git branch  
* master  
  ユーザー名
```

- 「git branch」コマンドで確認

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
 - この時リポジトリが master/main では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
 - リモートブランチに同じ名前で push
 - master ブランチに push しないように

ブランチを作成してファイルを Push

- 「git checkout ユーザー名」でブランチを変更

```
$ git checkout ユーザー名
```

- 「git branch」でブランチのチェックが変更されていることを確認

```
~/github/github_ws $ git branch
  master
* ユーザー名
```

任意ブランチに push する

- ・ 「git push origin ブランチ名」でpush
- ・ 今回はユーザーネーム

```
$ git push origin ユーザー名 █
```

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, Issues, Pull requests, Actions, Projects, Wiki, and Security. The 'Code' tab is highlighted with a blue border and a red underline. Below the navigation bar, there is a yellow banner with a user icon and the text 'ユーザ名 had recent pushes less than a minute ago'. To the right of the banner is a green button labeled 'Compare & pull request'. Underneath the banner, there are several status indicators: a dropdown menu showing 'master' (also highlighted with a blue border), '2 branches', '0 tags', and three buttons: 'Go to file', 'Add file', and 'Code'. A large blue callout box with white text is overlaid on the bottom right of the interface. The text inside the callout box is: 'Code → master をクリック' (Click on master) and 'Branches にアカウント名があればOK' (If the account name is in Branches, it's OK). On the left side of the main content area, there is a sidebar titled 'Switch branches/tags' with a search bar containing 'Find or create a branch...'. Below the search bar, there are two tabs: 'Branches' (which is selected and highlighted with a blue border) and 'Tags'. Under the 'Branches' tab, there is a list with a checked checkbox next to 'master' and another entry 'ユーザ名' (Account Name) which is also highlighted with a blue border. At the bottom of the sidebar, there is a blue button labeled 'View all branches'. At the very bottom of the page, there is a footer bar with the text 'README.md' and a small edit icon.

Code → master をクリック

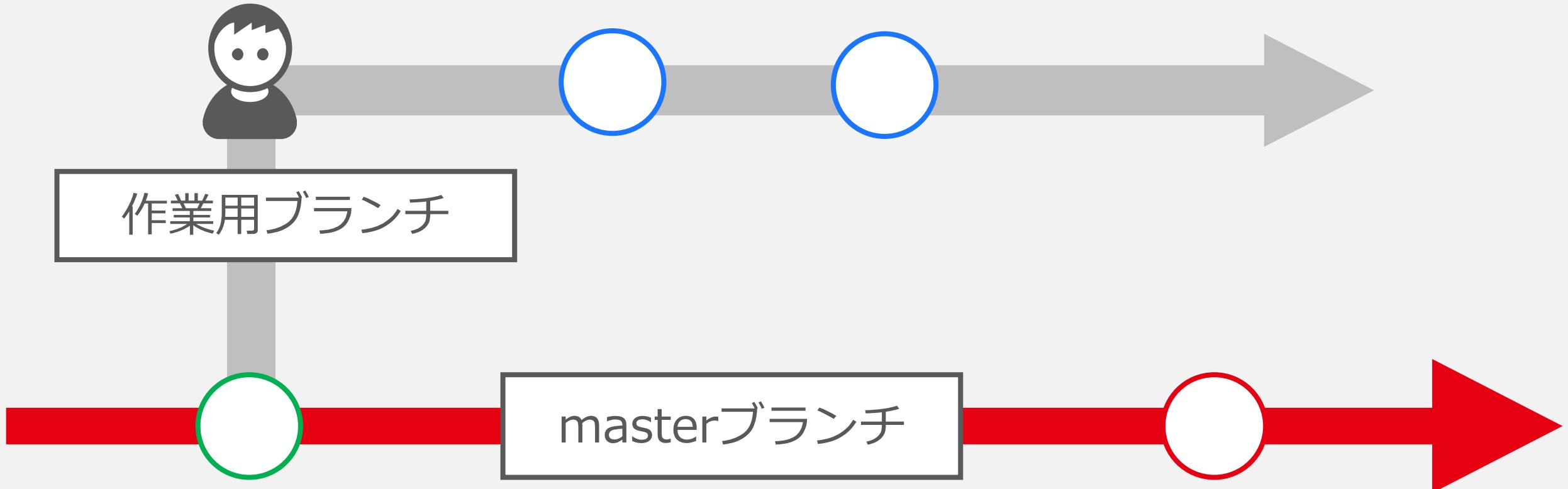
Branches にアカウント名があればOK

Step.2-2

masterに変更を反映する

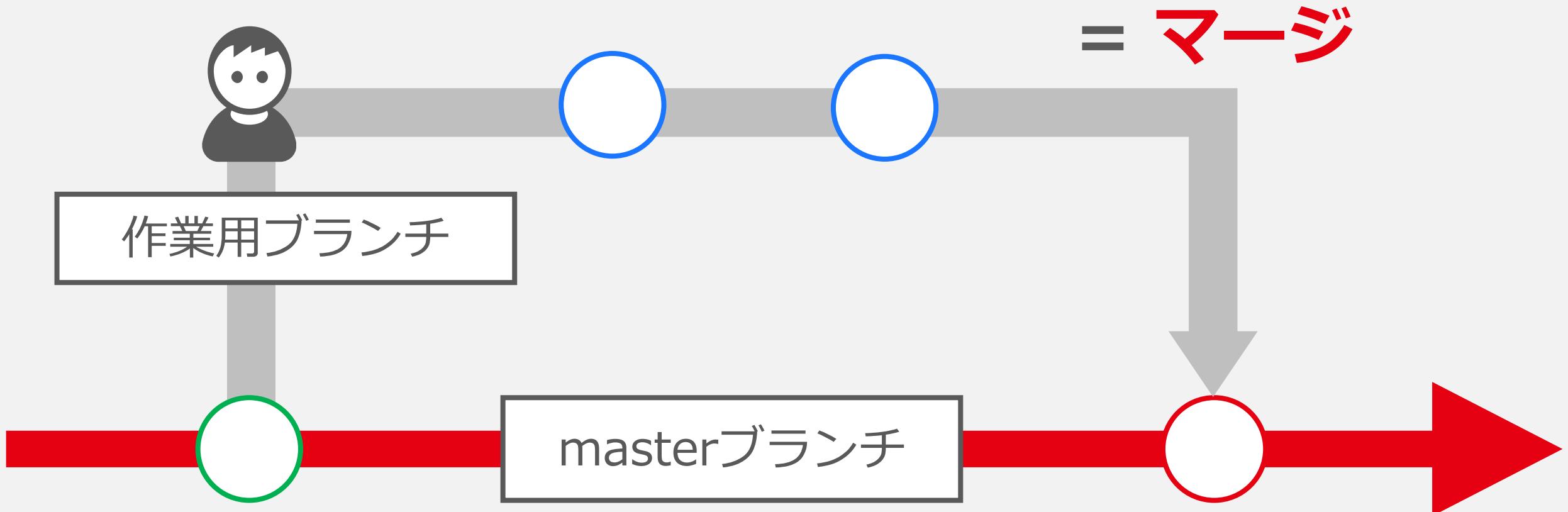
マージ

作業用ブランチの変更を master に反映する



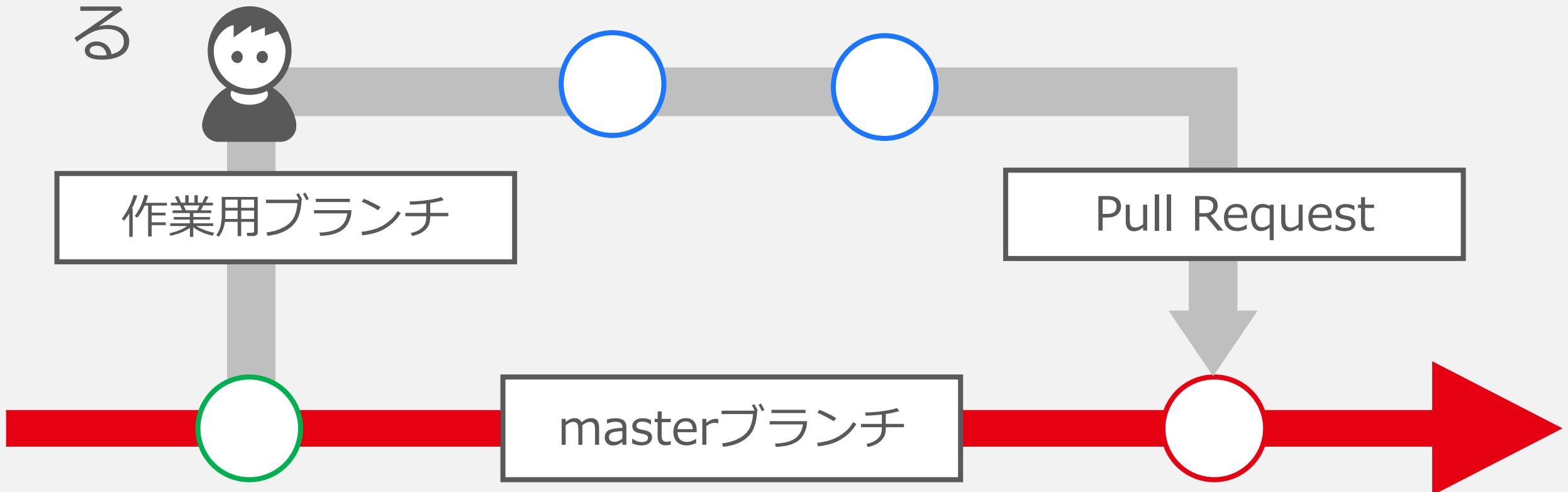
マージ

ブランチの変更を別のブランチに反映する



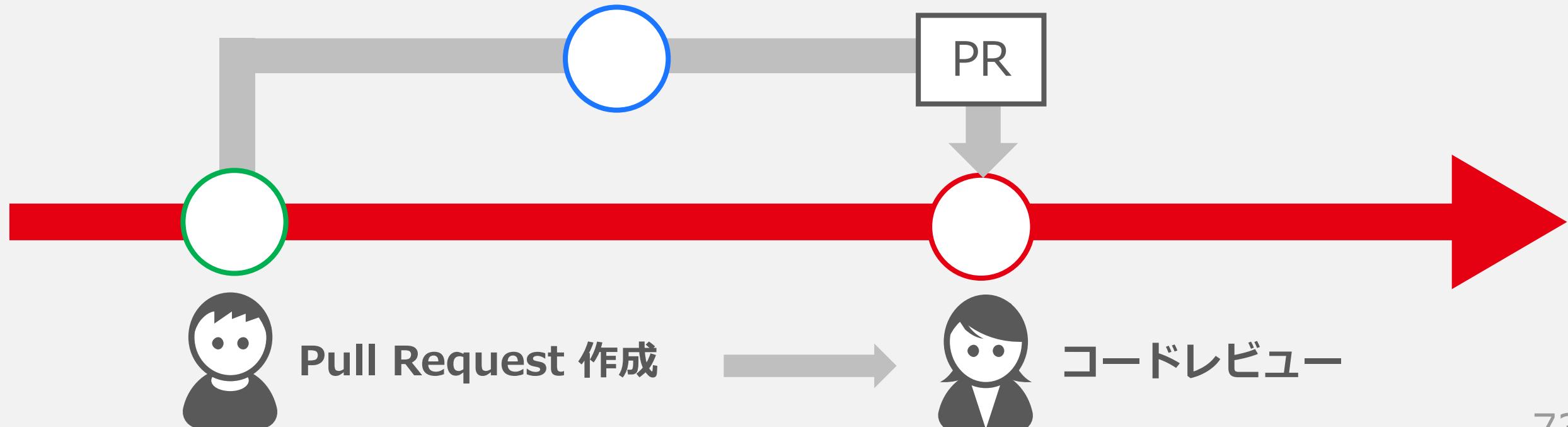
マージ

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

The screenshot shows a GitHub repository interface. At the top, there is a navigation bar with links: Code (highlighted), Issues, Pull requests, Actions, Projects, Wiki, and Security. Below the navigation bar, a yellow banner displays a message: "ユーザーネーム had recent pushes less than a minute ago" and a green button labeled "Compare & pull request". Underneath the banner, there are buttons for "master" (with a dropdown arrow), "2 branches", "0 tags", "Go to file", "Add file" (with a dropdown arrow), and "Code" (with a dropdown arrow). On the left side, there is a sidebar with a user icon and the text "ユーザーネーム Add R...", followed by a list of files: "src", "README.md", and "index.html". At the bottom of the sidebar, there is another "README.md" entry. A large blue callout bubble is overlaid on the right side of the screenshot, containing Japanese text: "GitHub にいくとこのような画面が出てきます (青枠注文)" and "※もし出でない場合は Pull requests のページから作成できます".

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

The screenshot shows a GitHub repository interface. At the top, there are navigation links: Code (highlighted with a red underline), Issues, Pull requests, Actions, Projects, Wiki, and Security. Below the navigation, a yellow banner displays a message: "ユーザ名 had recent pushes less than a minute ago" and a green button labeled "Compare & pull request". The main area shows a dropdown for the branch "master", a status bar indicating "2 branches" and "0 tags", and three buttons: "Go to file", "Add file", and "Code". A large blue callout box in the center says "Compare & pull request を押す". Below this, a list of recent commits is shown:

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

At the bottom, a file named "README.md" is listed with an edit icon.

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 interface for creating a pull request. 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 these are fields for 'PR タイトル' (Title), 'Write' (with a 'Preview' button), and '変更内容を記載' (Describe changes). A large blue callout box highlights the following terms:

- 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

Other visible fields include 'Attach files by dragging & dropping, selecting or pasting them.', a 'Create pull request' button, and sections for 'Projects', 'Milestone', and 'Linked issues'.

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 rectangular selection. 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 covers the middle section of the form, 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". At the bottom of the form, there's a "Create pull request" button.

PR タイトル

PR 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

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



Pull Request の作成方法

Pull Request の作成方法

The screenshot shows a GitHub pull request page with the following details:

- Title:** PR タイトル #1
- Status:** Open
- Merge Details:** ユーザー名 wants to merge 1 commit into master from ユーザー名
- Activity:** Conversation (0), Commits (1), Checks (0), Files changed (2)
- Comment:** ユーザー名 commented now: 変更内容を記載
- Commit:** Update index.html (commit hash: 78149a0)
- Instructions:** Add more commits by pushing to the ユーザー名 branch on ユーザー名 / github_ws.
- CI Status:** Continuous integration has not been set up. GitHub Actions and several other apps can be used to automatically catch bugs and enforce style.
- Branch Conflict:** This branch has no conflicts with the base branch. Merging can be performed automatically.
- Buttons:** Merge pull request, Open this in GitHub Desktop, 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

Changes from all commits ▾ File filter... ▾ Jump to... ▾ Review changes ▾

1 / 2 files viewed

Files changed に変更差分が掲載される

3 README.md

2 index.html

```
@@ -11,7 +11,7 @@ <h2>演習で自由に編集してください</h2>
 11 11      <!-- 変更箇所に困ったら下記のコメントアウトを外してください -->
 12 12      <div id="target">ここをクリックするとアラート発生</div>
 13 13
 14  -      <!--<div id="ok">GitHub完全に理解した</div>-->
 14  +      <div id="ok">GitHub完全に理解した</div>
 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 icon with a gear and a plus sign is shown next to the message.

The main review area contains two items:

- 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.

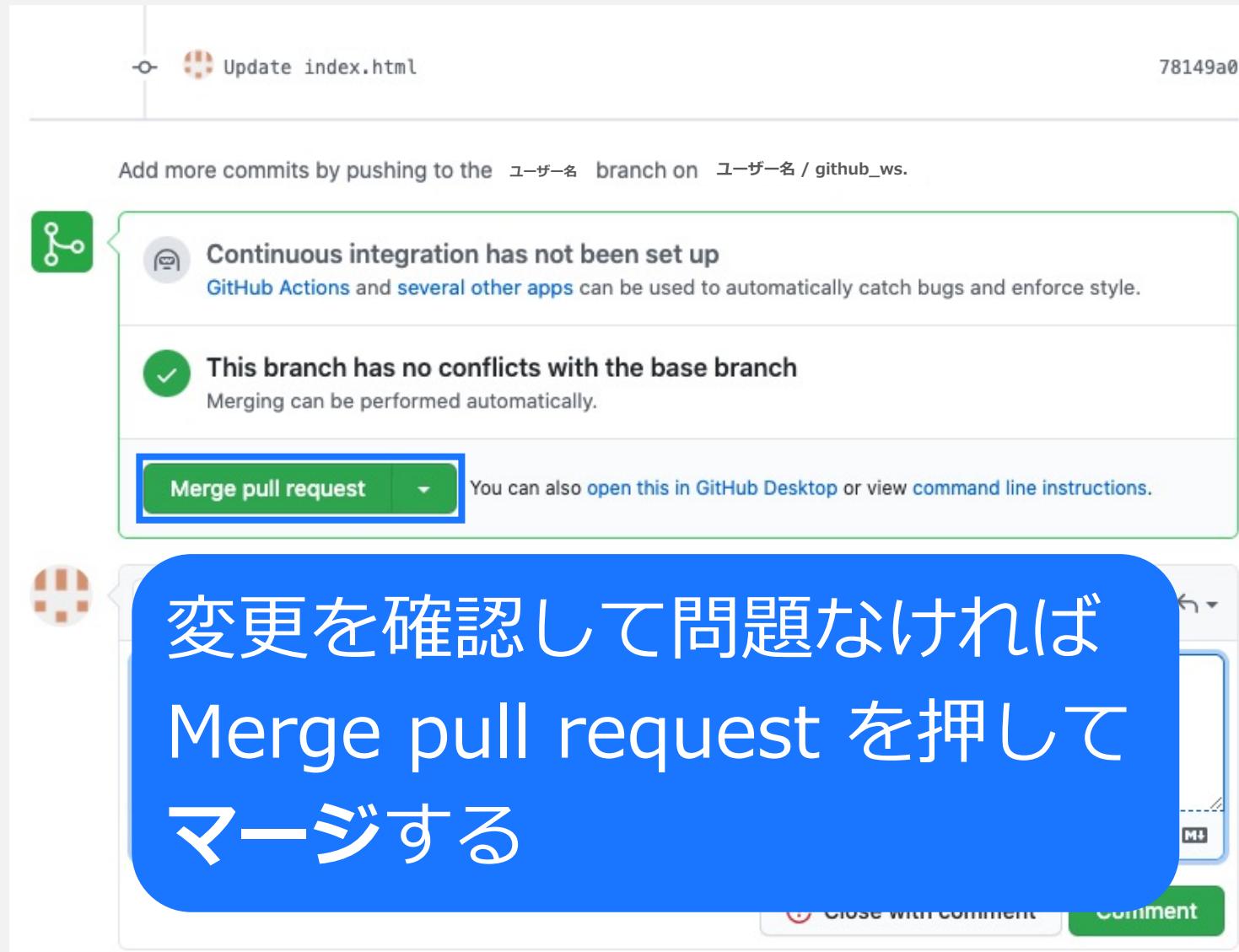
At the bottom of the review area, there's a "Merge pull request" button and a note: "You can also open this in GitHub Desktop or view command line instructions."

Below the review area is a comment input field with a blue border. The placeholder text in the field is "レビューや気になった内容はコメントできる". Below the input field, there's a note: "Attach files by dragging & dropping, selecting or pasting them." At the bottom of the comment input field, there are two buttons: "Close with comment" and "Comment".

Pull Request の確認



Pull Request をマージ



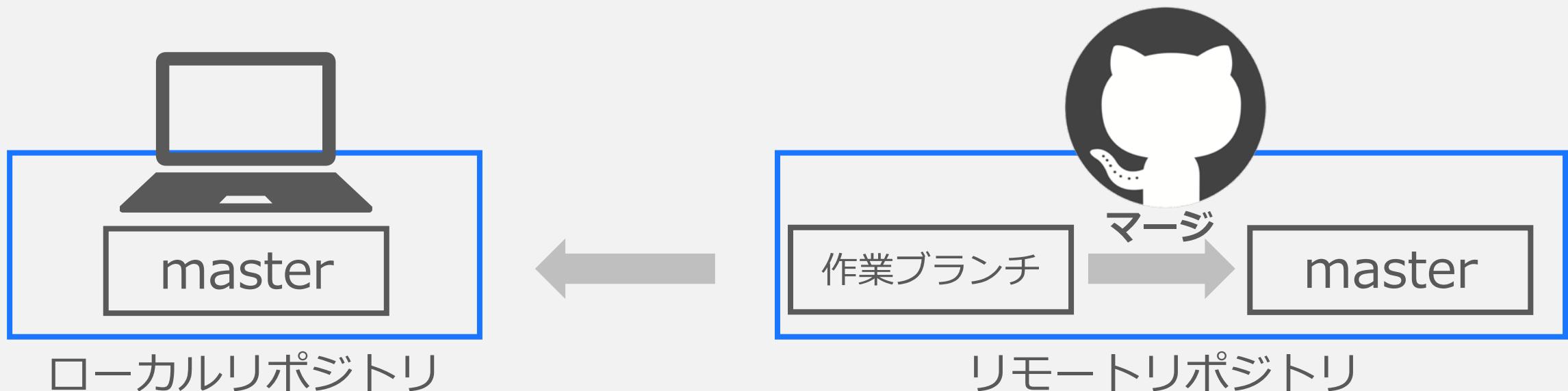
ローカルリポジトリに
pull する

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

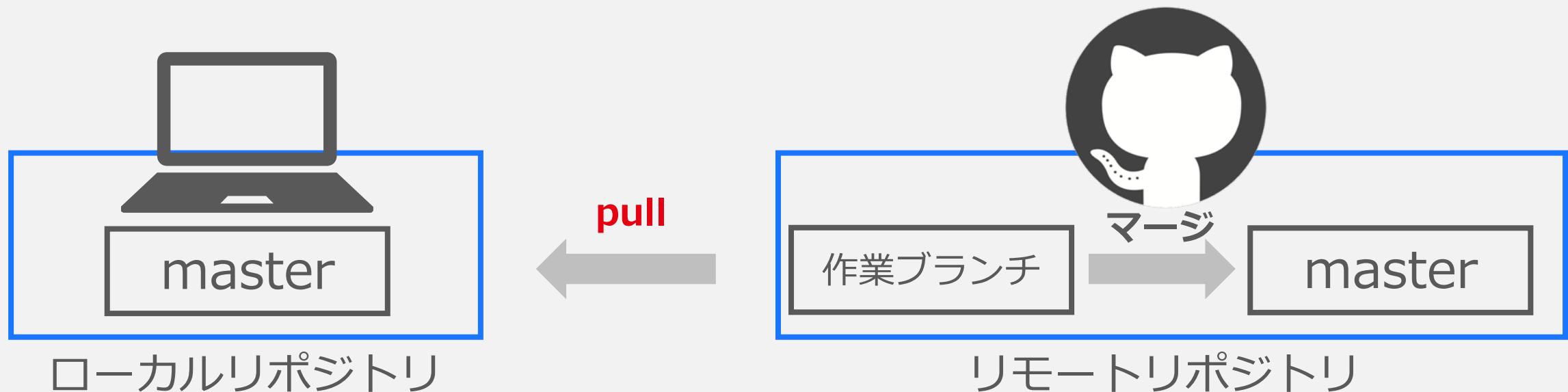
- ・ リモートの master には反映される
- ・ ローカルの master には反映されていない



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



Git CUIで
pull する



ローカルに pull する

- 「git checkout master」でmasterに切り替える

```
$ git checkout master
```

- 「git branch」でmaster/mainになっていることを確認

```
[~/github/github_ws $ git branch  
* master  
ユーザー名
```

ローカルに pull する

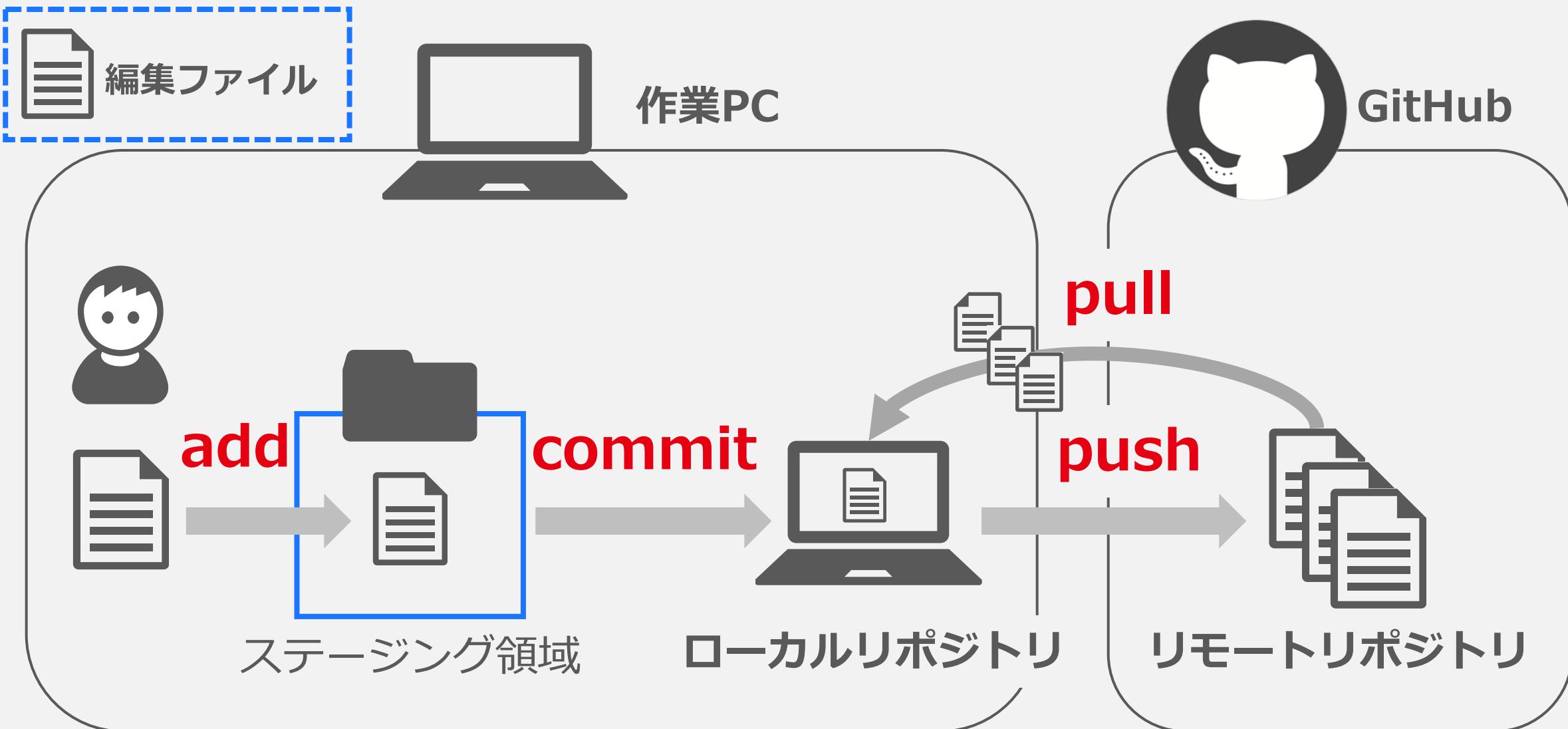
- ・ 「git pull」でpullする

```
$ git pull
```

- ・ ローカルの該当ファイルを開く

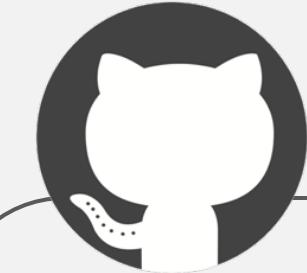
→ ファイルの変更箇所を確認

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 master 反映
 - 1. Pull Request を作成
 - こだわった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 master に pull
 - 変更が反映されてるかファイル内容を確認しよう

ブレイクアウトルームへ移動します

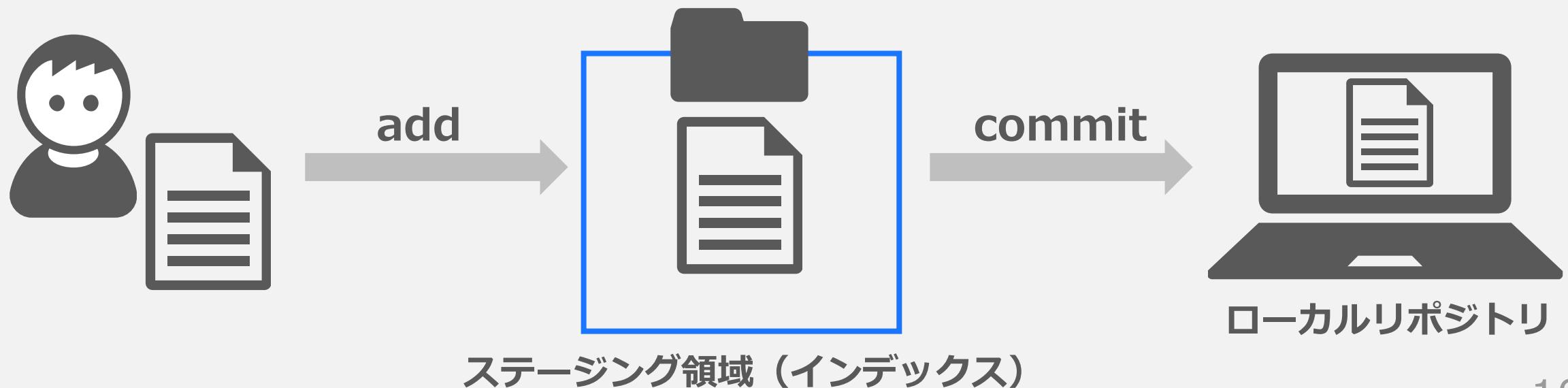


おさらい

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**

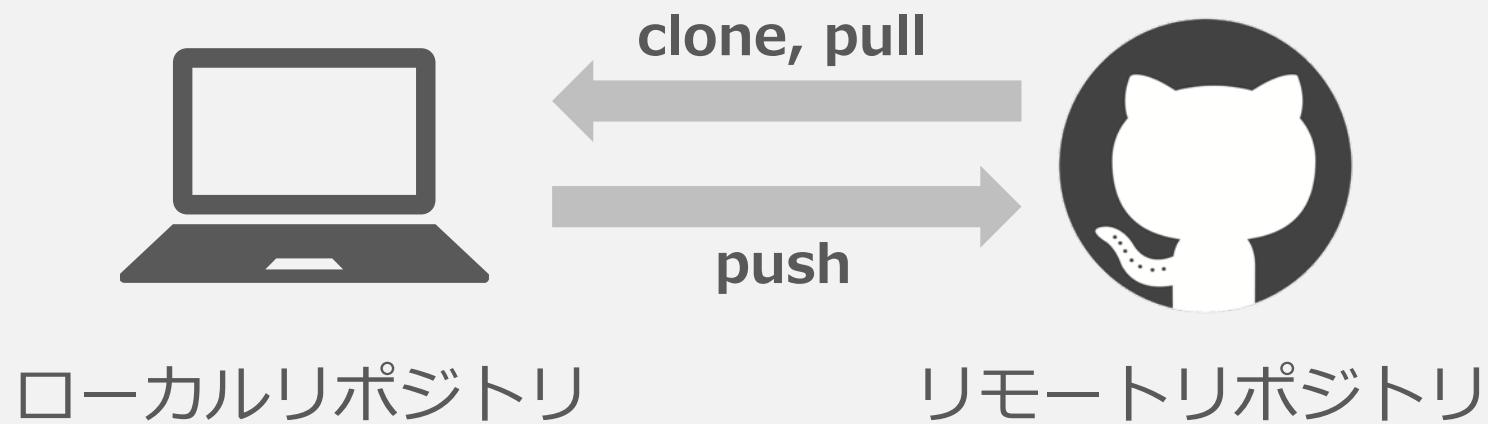


【おさらい】ローカルとリモートの関係

clone : リモートをローカルにコピー

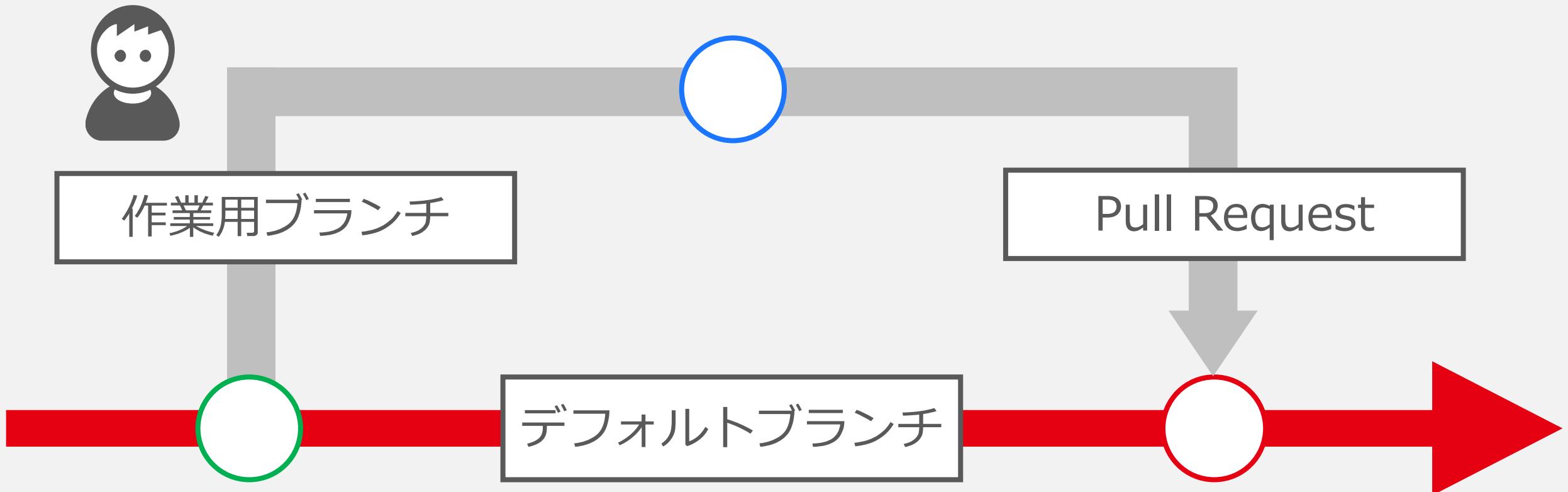
pull : リモートの変更をローカルに反映

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。

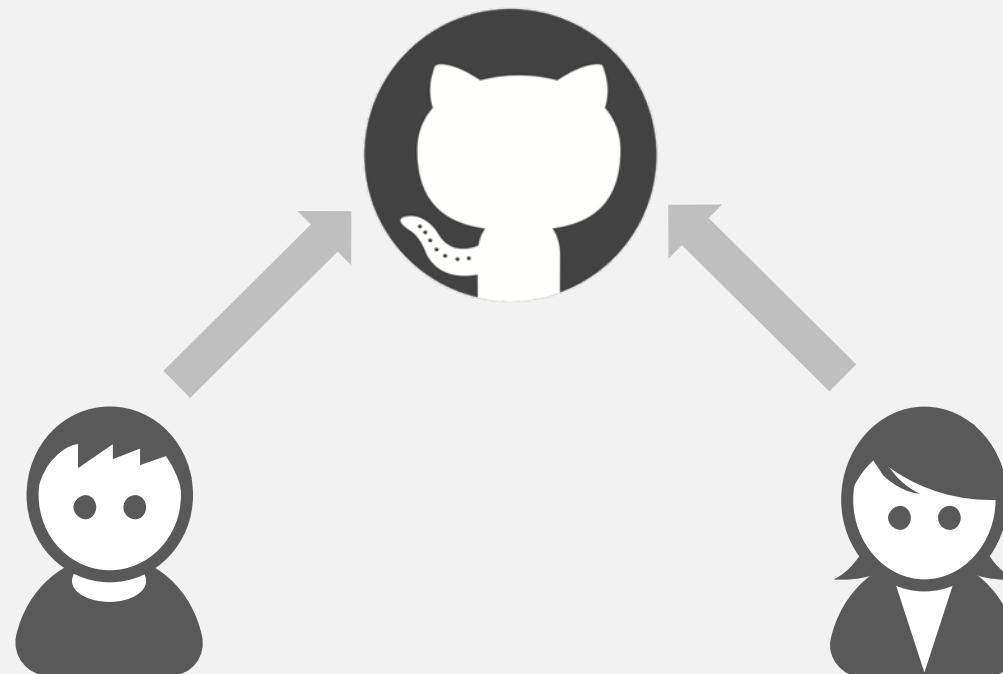
まずは基本的な使い方に慣れる。
使っていく中で調べながら使う。

GitHub を複数人で使う

YAHOO!
JAPAN

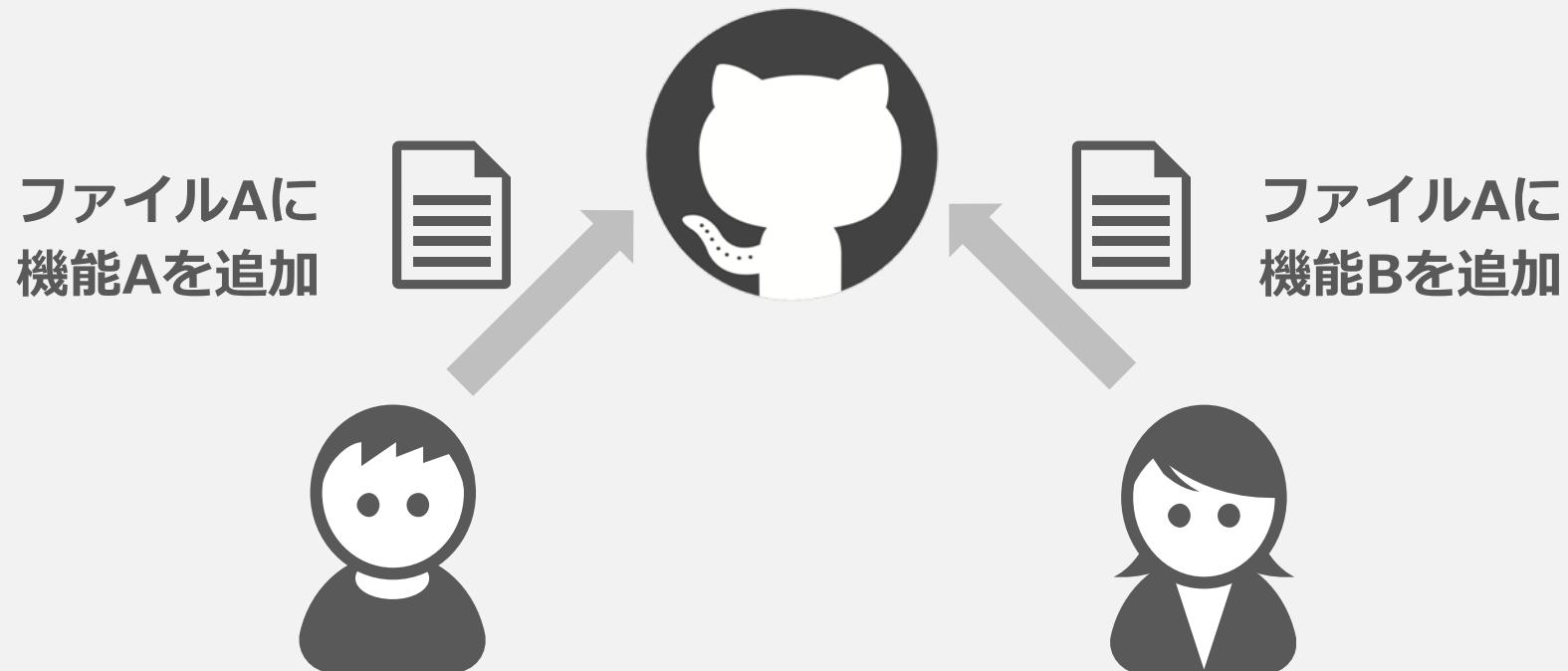
チーム開発では

- 1つのリポジトリに対して複数人で作業する



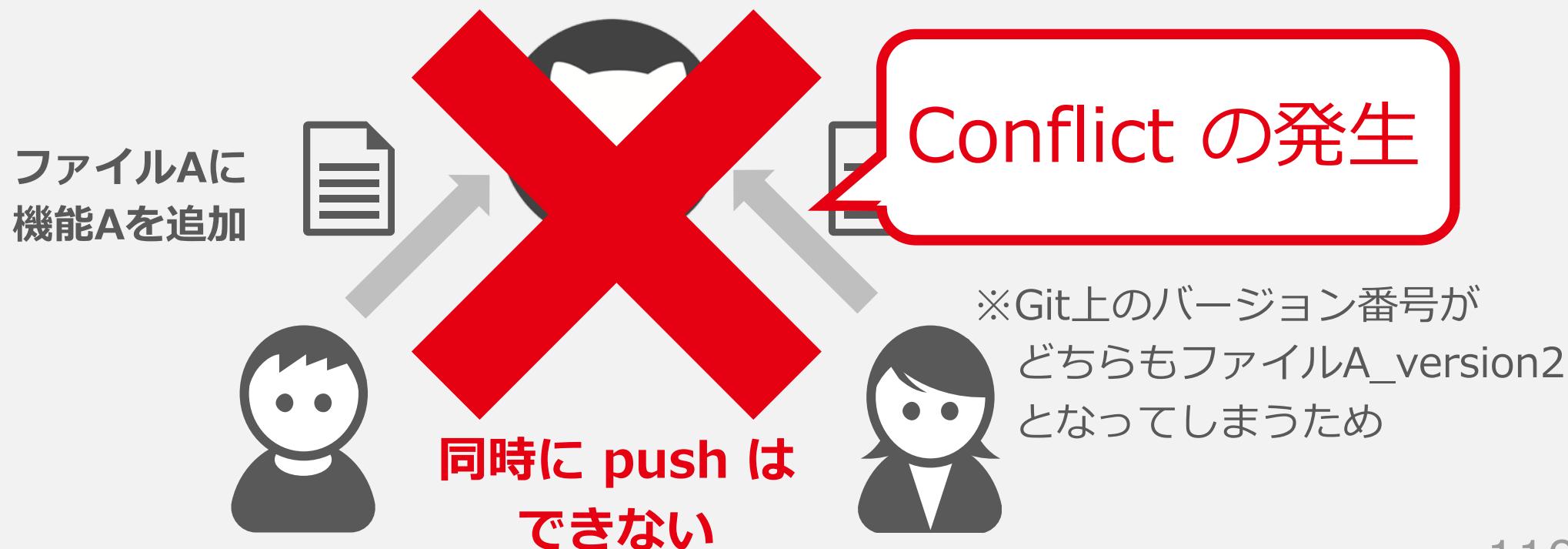
チーム開発では

- ・ 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



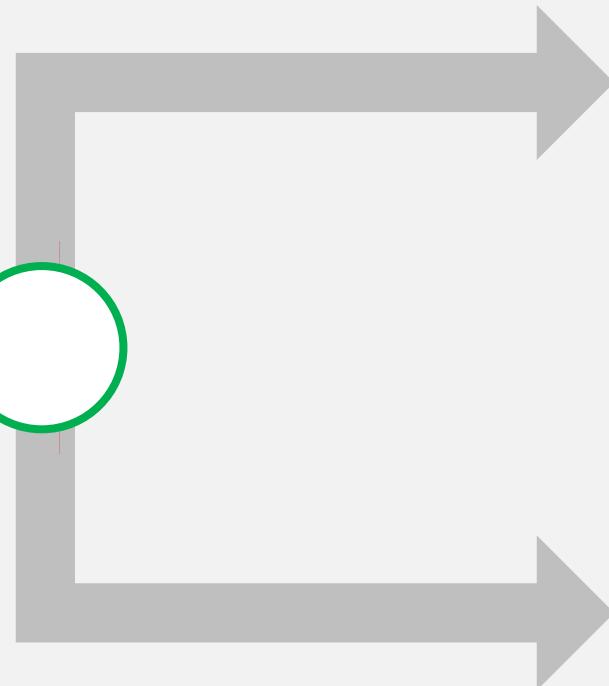
同時に push は
できない



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

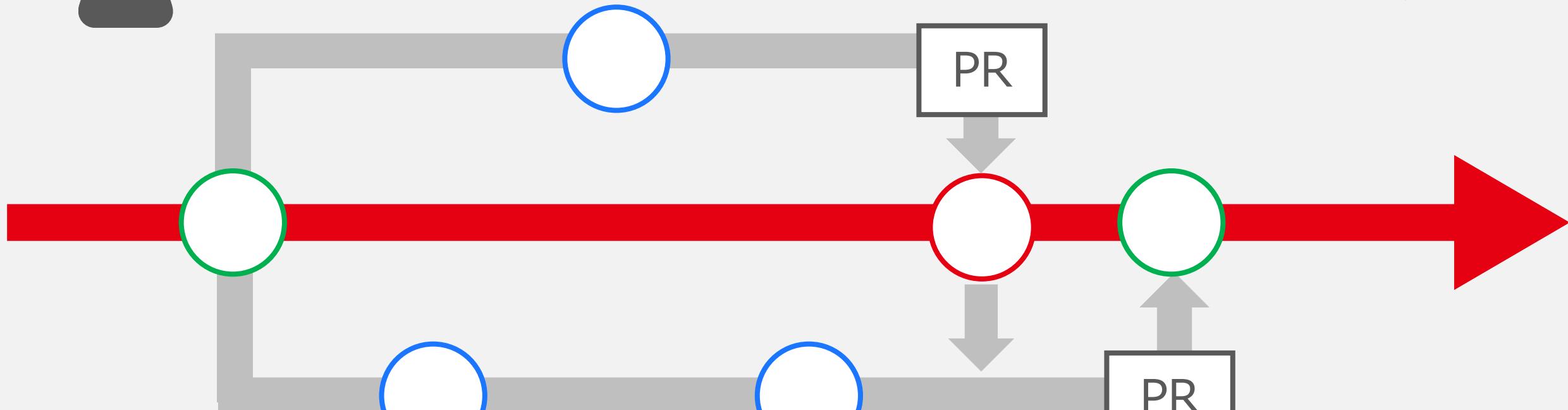
作業者がそれぞれブランチを作成

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



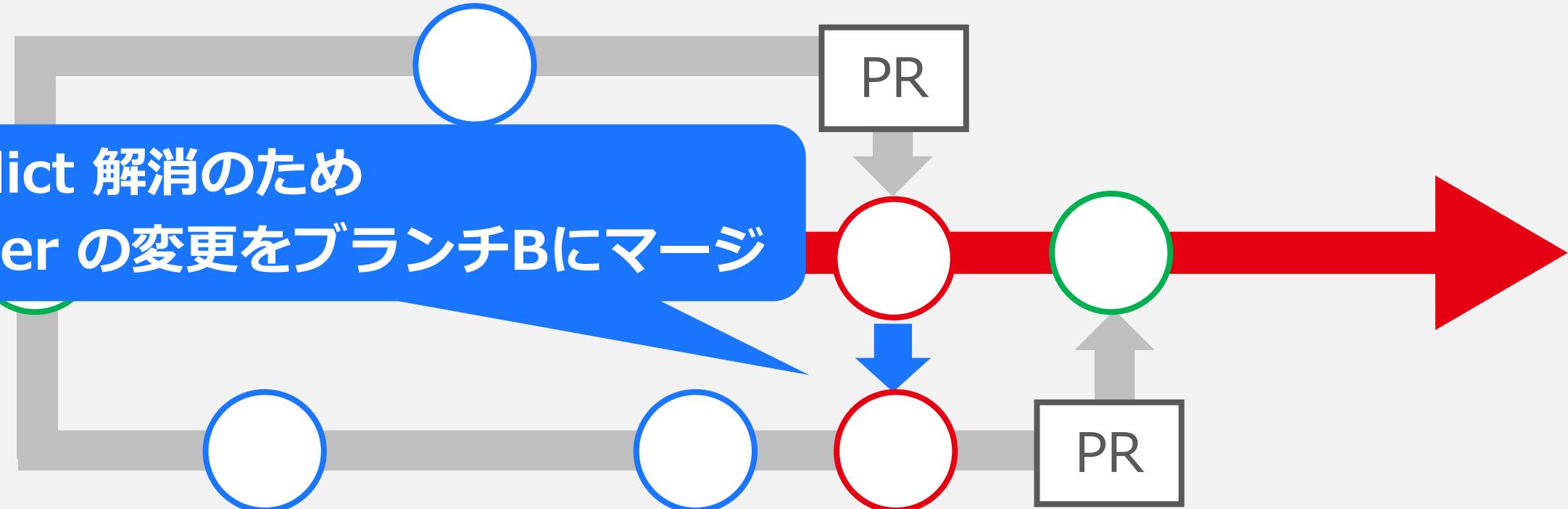
Aさんの作業ブランチ「A」

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
master 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

アンケート記入
お願いします



記念撮影

YAHOO!
JAPAN

懇親会

YAHOO!
JAPAN

懇親会

- 興味あるテーマを選びます
 - ブレイクアウトルームに移動後、好きな部屋へ移動してください
- group_1:** イベントの内容、エンジニアの技術について
- group_2:** 就活・採用
- group_3:** フリーテーマ

アンケート記入が済んでいればいつ退出してもOKです

クロージング

YAHOO!
JAPAN

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
```