

# 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。



# イベントの流れ

- ・ 導入：セットアップの確認 / Gitとは
- ・ 第一部：GitHubの基本操作（講義 + ハンズオン）
- ・ 第二部：ブランチの作成とマージ（講義 + 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. There is 1 branch and 0 tags. A commit history is shown:

- 50cdc45 14 minutes ago 1 commits  
- src: Add index.html & src (14 minutes ago)  
- index.html: Add index.html & src (14 minutes ago)

A large blue callout box covers the top right portion of the repository page. It contains the text:

add/commit/push を行い  
演習ファイルが上がっていればOK

At the bottom left of the repository page, there is a message: "Help people interested in this repository understand your project by adding a README." with a "Add a README" button. At the bottom right, there is a "Releases" section with the message: "No releases published Create a new release".

# Gitとは



# Git とは

- Gitとは「バージョン管理システム」である
  - バージョン管理とは  
「ファイルの変更を記録しておくこと」
- いつ
  - 誰が
  - どのファイルを
  - どう変更したか

# Git / GitHub とは

Git	バージョン管理システム
GitHub	<ul style="list-style-type: none"><li>• Gitでバージョン管理をしやすくする <b>Webサービス</b></li><li>• ブラウザから変更履歴やコードを共有 することができる</li><li>• コミュニケーションツールの側面も ある</li></ul>

# リポジトリとは

**リポジトリ**：ファイルの変更を記録する場所



APIサーバ  
リポジトリ



Androidアプリ  
リポジトリ

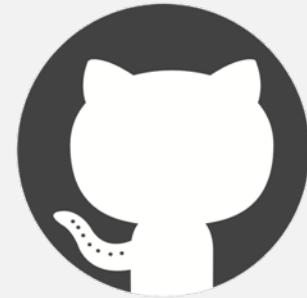


iPhoneアプリ  
リポジトリ

# リポジトリとは

- **リモートリポジトリ**

GitHub上（Web上）に存在するもの  
複数人で共有することができる



- **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



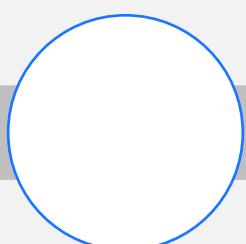
# バージョン管理とは

リポジトリ単位でのバージョン管理を行う

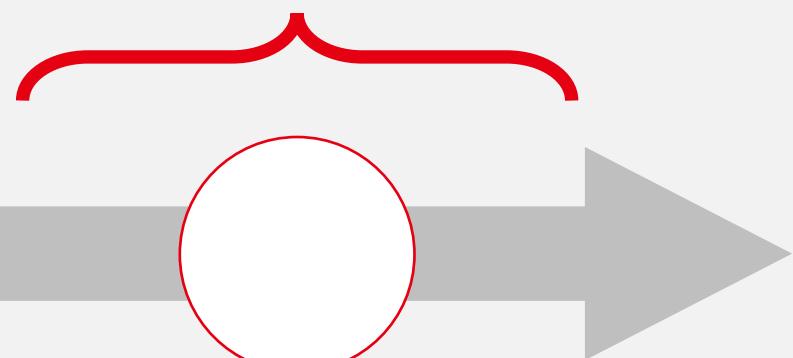
過去の状態



2022/04/10 18:30  
Aさんが  
背景画像を変更



最新の状態



2022/05/01 00:00  
Aさんが  
ロゴを変更

2022/04/30 10:00  
Bさんが  
タイトルを変更

# 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



# Git を使うには

主に以下の 2 つの方法がある

- ・ 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

## 第一部

# GitHub の基本操作



# 第一部で扱う内容

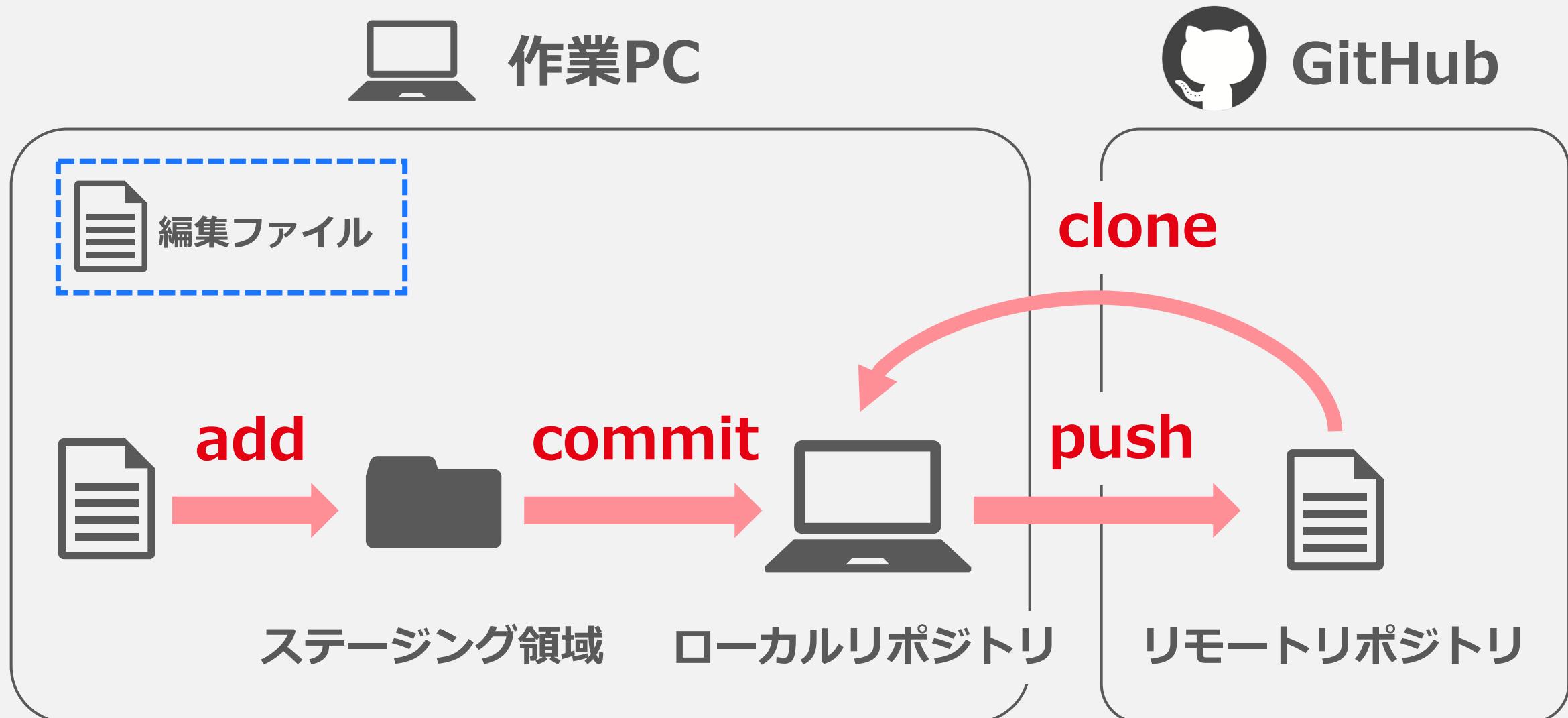
## 講義：GitHubの基本操作

- clone の操作
- add, commit の操作
- push の操作

## ハンズオン

- ファイルを追加/変更して、リモートにpush

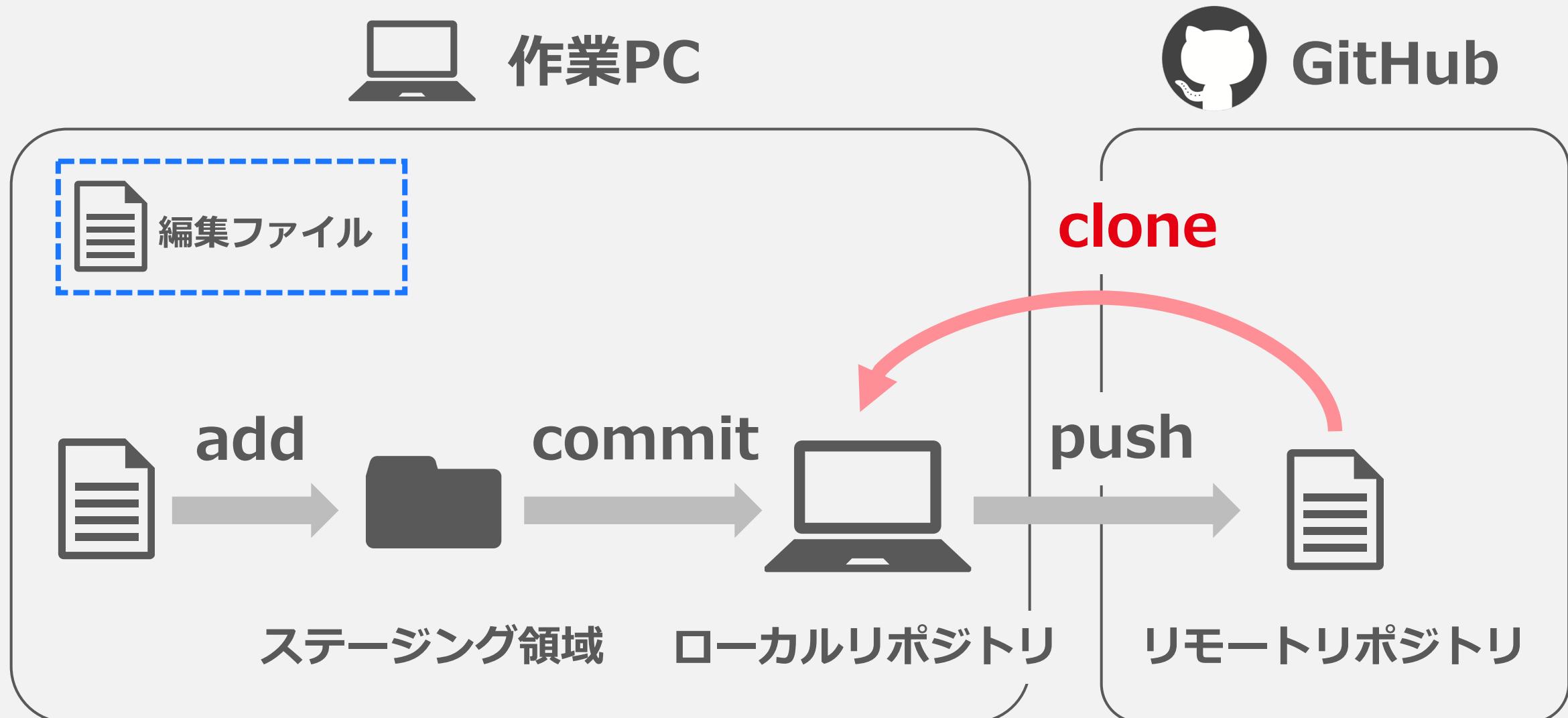
# GitHub の基本操作



# clone の操作

YAHOO!  
JAPAN

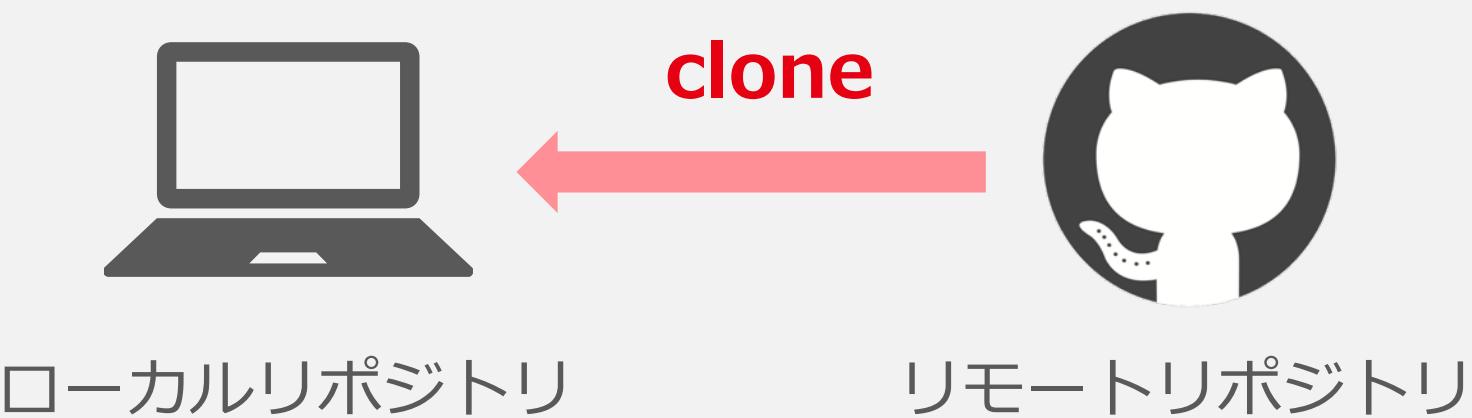
# GitHub の基本操作



# GitHub から自分のPCに持ってくる

## ● clone

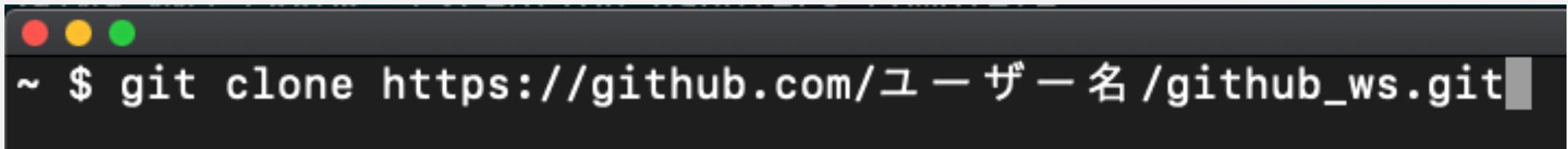
リモートリポジトリを自分のローカルリポジトリに  
コピーする（この操作は初回のみ行う）



# GUIコマンド - clone

1. 適当なディレクトリに移動する  
※ 「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
2. clone のコマンドを打つ

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



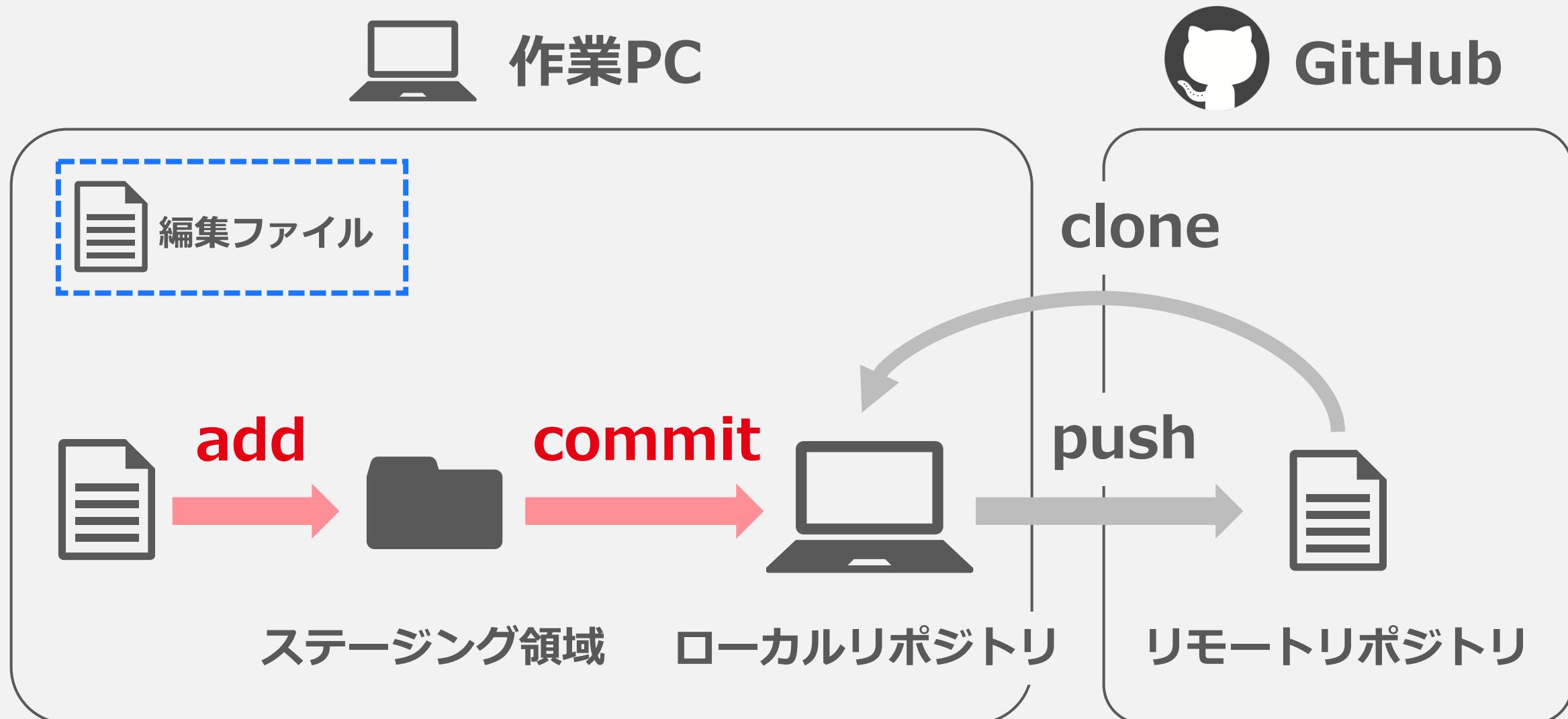
A screenshot of a macOS terminal window. The window has a dark background with light-colored text. At the top left are three colored window control buttons (red, yellow, green). The main area contains a command line interface. The command `$ git clone https://github.com/ユーザー名/github_ws.git` is visible, with the part `ユーザー名` replaced by a placeholder ellipsis. The cursor is positioned at the end of the command.

この場所にローカルリポジトリを作成する

# add, commit の操作



# GitHub の基本操作



# ファイルを修正してGitで扱う

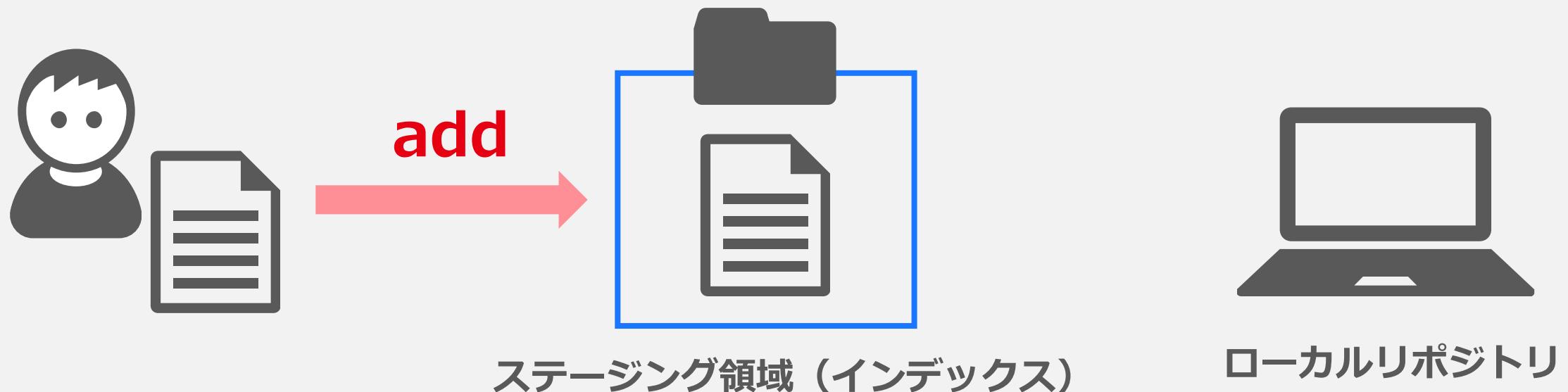
ファイルの変更をローカルリポジトリに登録するには**add**と**commit**の2つのステップが必要になります



# ファイルを修正してGitで扱う

- **add**

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



# ファイルを修正してGitで扱う

## ● commit

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



ステージング領域（インデックス）

ローカルリポジトリ

# GUIコマンド - add

- addする前に `$ git status` で変更されたファイルを確認します。

```
~/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)
```

# GUIコマンド - add

- `$ git add .` (最後にピリオドが必要です) でadd
- `$ git status` でファイルごとの変更内容が出てい  
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

- 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**
- 以下のような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
  - 個人情報
  - パスワード
  - API Key などの鍵情報

# GUIコマンド - commit

- \$ 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
```

# Commit する際のポイント

- commitのポイント

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

「OOを変更した」



**commit**



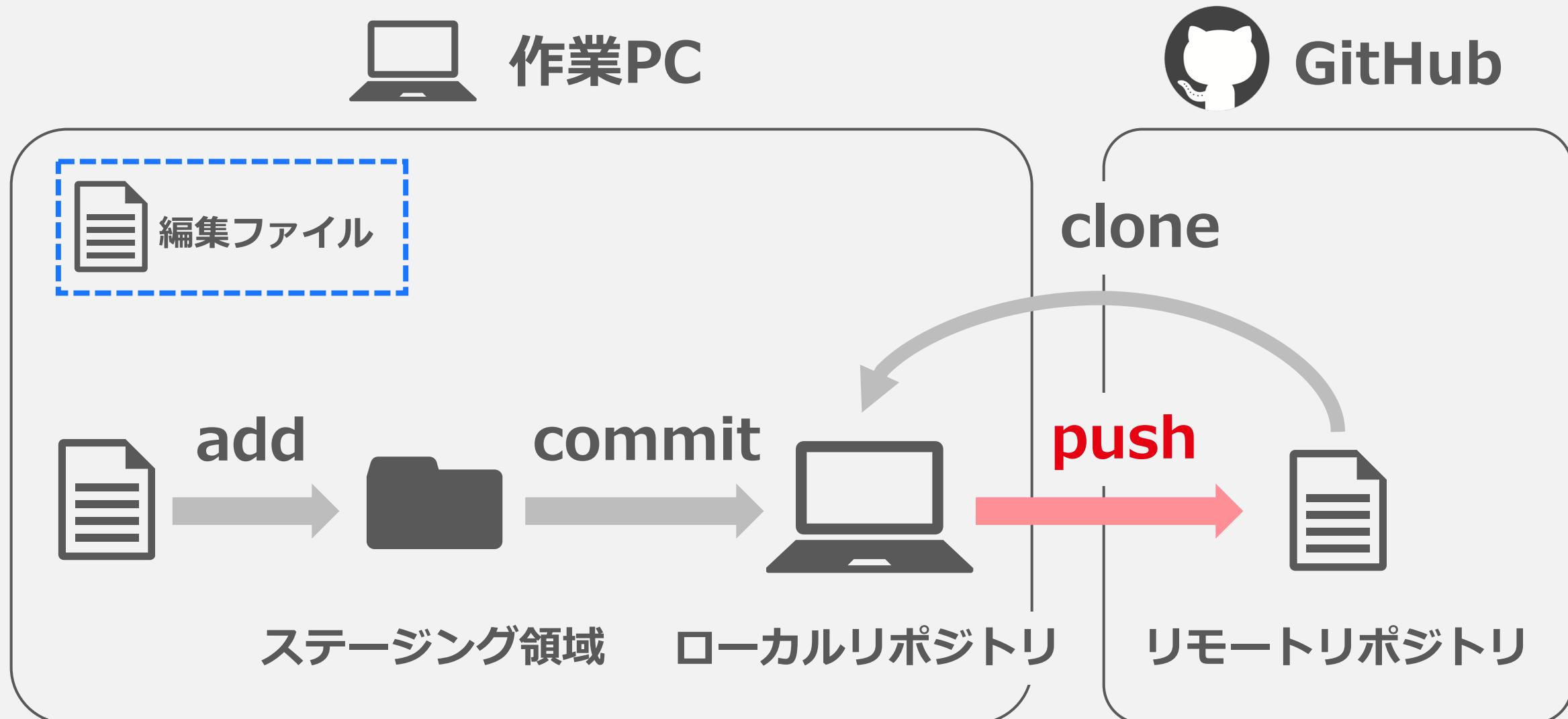
ステージング領域 (インデックス)

ローカルリポジトリ

# push の操作

YAHOO!  
JAPAN

# GitHub の基本操作



# GitHub から自分のPCに持ってくる

- **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

# GUIコマンド - push

- \$ git push origin master でpushします

※ ブランチ名がmainの場合はmaster -> main。  
commit時の表示を確認してください

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

- 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 に反映されているか確認

Screenshot of a GitHub repository page for 'github\_ws'.

The repository details:

- Code: master branch, 1 branch, 0 tags
- Last commit: 50cdc45, 14 minutes ago, 1 commits
- Commits:
  - src: Add index.html & src, 14 minutes ago
  - index.html: Add index.html & src, 14 minutes ago
- Help message: Help people interested in this repository understand your project by adding a README.
- Add a README button.

Repository statistics:

- Unwatch: 1
- Star: 0
- Fork: 0

About section:

No description, website, or topics provided.

Releases section:

No releases published

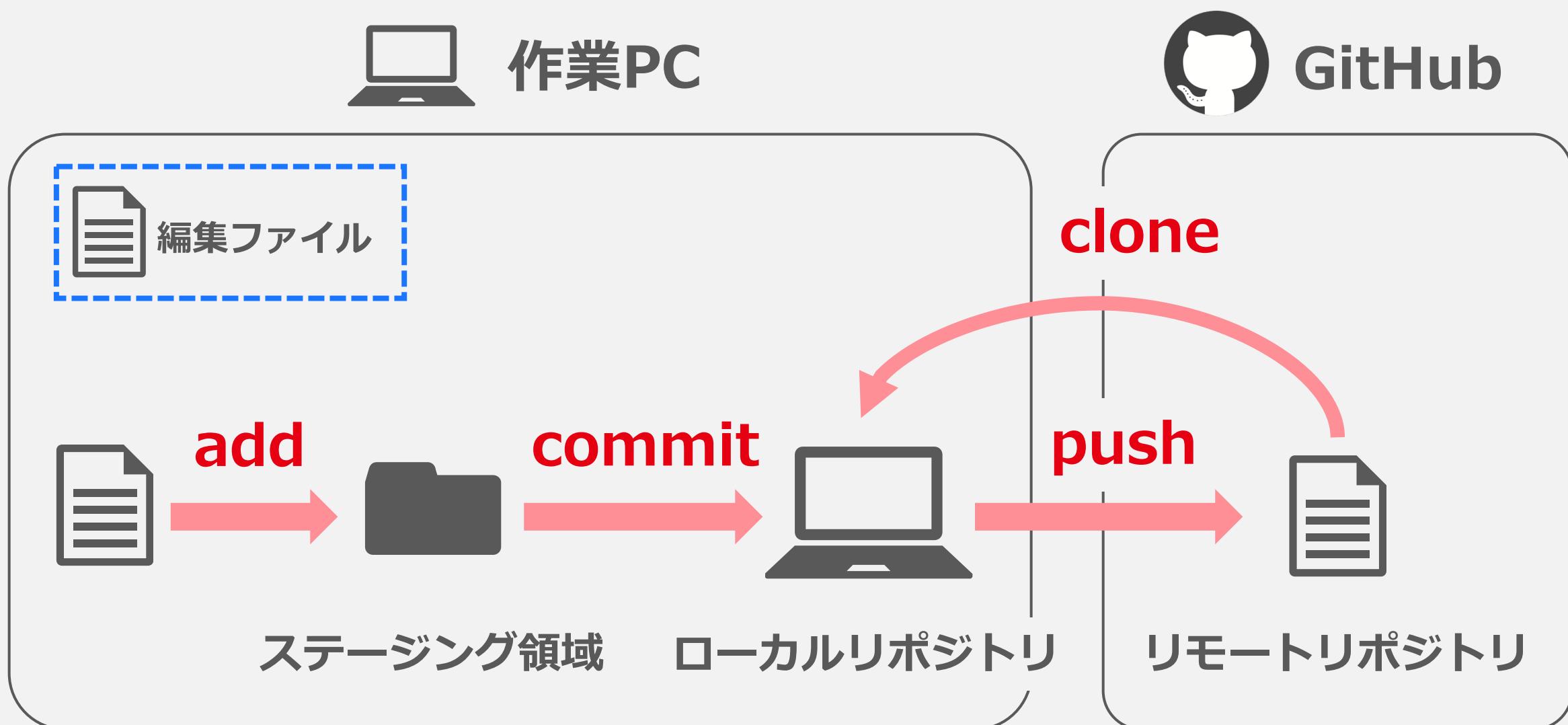
Create a new release

# Pushがうまくいかない時

- 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



# 第一部

# ハンズオン

YAHOO!  
JAPAN

# 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

## 手順

1. README.md の作成 / index.html を編集する
  - ファイルはVSCode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てみよう

# 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

# 補足

- 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、変更履歴などを記載する。  
Markdown記法で記述する。

ハンズオンでは自分のリポジトリの説明を追加  
しましょう。

# 補足

ファイル差分の確認は \$ git status コマンドを使う

```
~/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

## 第二部

# ブランチの作成とマージ

# 第二部で扱う内容

## 講義：ブランチの作成とマージ

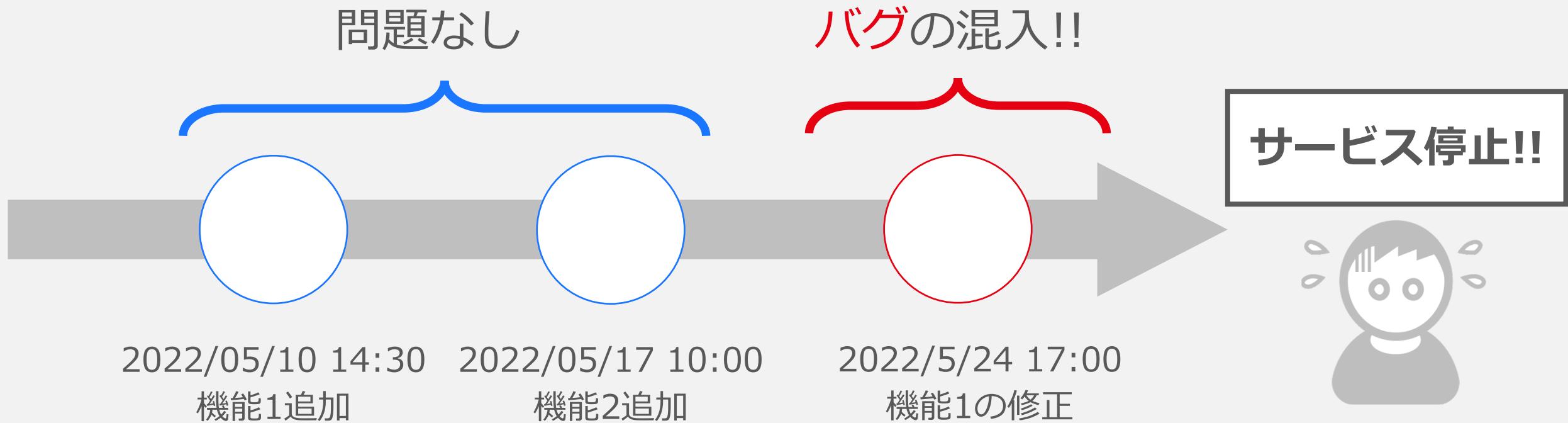
- ・ ブランチの作成、移動、ブランチからのpush
- ・ マージ / Pull Request
- ・ ローカルリポジトリへのpull

## ハンズオン

- ・ 新たにブランチを作成し、ファイルを push
- ・ Pull Request を作成して master 反映

# バグの混入

リポジトリに動かないコードが上がってしまうと・・・



# バグの混入

リポジトリに動かないコードが上がってしまうと・・・

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

2022/05/10 14:30

機能1追加

2022/05/17 10:00

機能2追加

2022/5/24 17:00

機能1の修正

!!

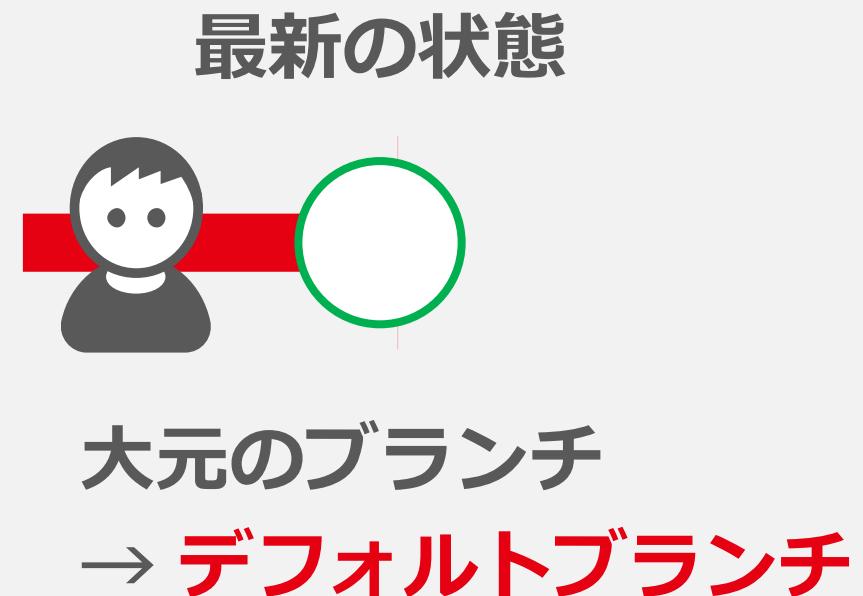
# ブランチの作成

YAHOO!  
JAPAN

# ブランチ

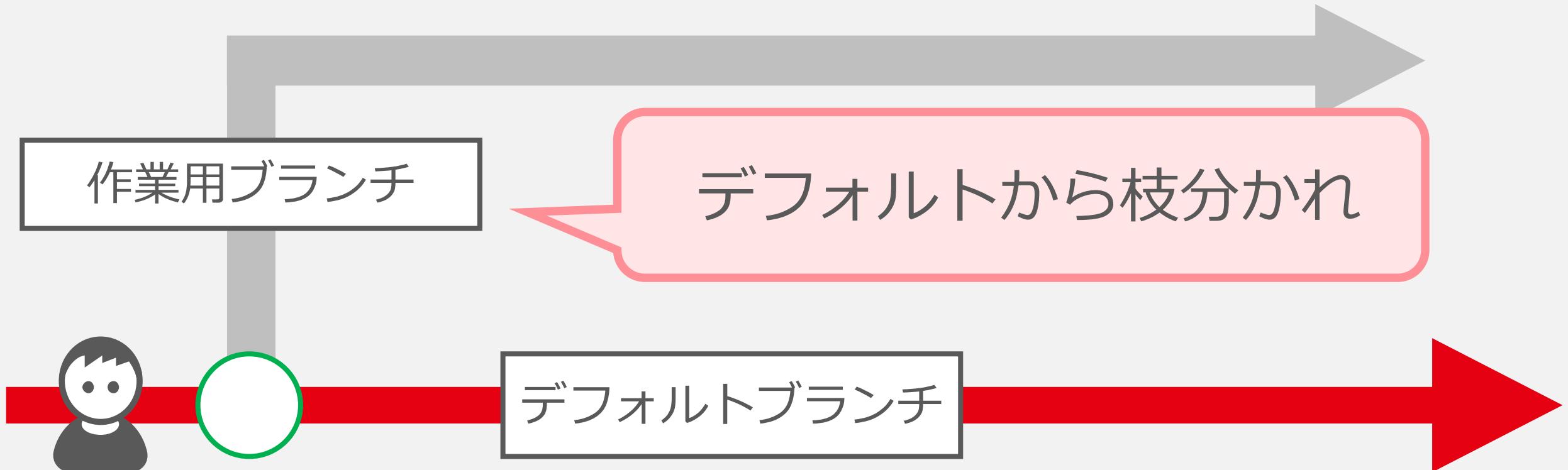
- ブランチ (=枝)

- 作業環境を分けるもの
- 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



# ブランチを作成する

- \$ git branch <ユーザー名> でブランチ作成
- \$ git branch コマンドで現在のブランチを確認

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

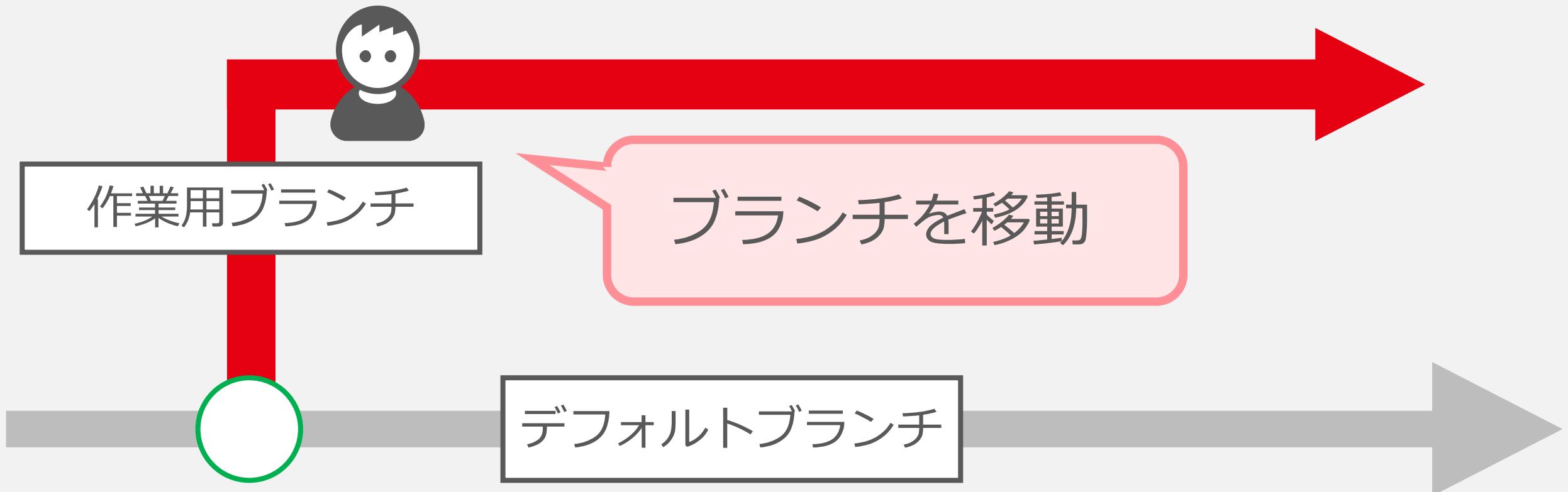
ブランチ名は自分で好きに決められますが、今回は新規  
ブランチ名をGitHubアカウントと同じにしましょう。

# ブランチの移動

YAHOO!  
JAPAN

# ブランチの移動

作業用ブランチに移動 (=チェックアウト) する



# ブランチを移動する

- \$ git checkout ブランチ名 でブランチを変更

```
$ git checkout ユーザー名
```

- \$ git branch  
でブランチが変更されていることを確認

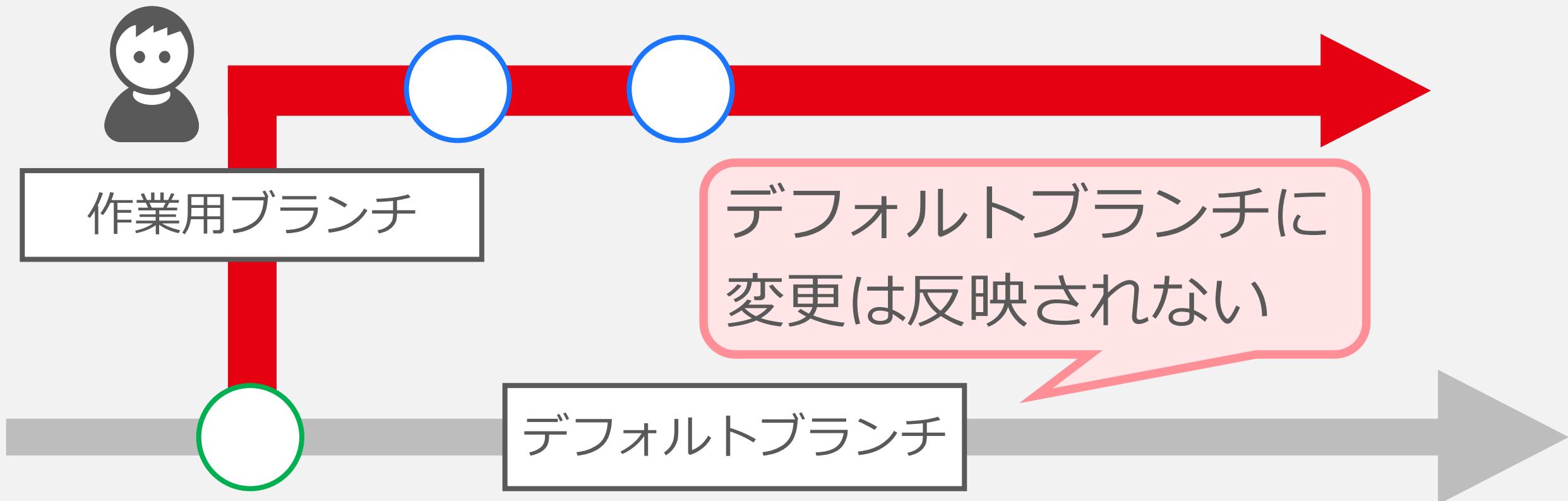
```
~/github/github_ws $ git branch  
  master  
* ユーザー名
```

# 作業ブランチからのpush

YAHOO!  
JAPAN

# 作業ブランチからの push

作業ブランチで add/commit/push する



# 作業ブランチからの 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
- \$ git push origin ブランチ名 でpush
  - 今回はブランチ名 = ユーザ名

```
$ 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 links: Code, Issues, Pull requests, Actions, Projects, Wiki, and Security. The 'Code' link is highlighted with a red box. 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 buttons for "master" (with a dropdown arrow), "2 branches", "0 tags", "Go to file", "Add file", and "Code". A dropdown menu titled "Switch branches/tags" is open, showing a search bar with "Find or create a branch..." and tabs for "Branches" and "Tags". The "Branches" tab is selected and highlighted with a red box. It lists "master" (marked with a checkmark) and "ユーザ名" (User Name). A blue callout bubble points to the "ユーザ名" entry with the text "Code → master をクリック" (Click master to switch). Another blue callout bubble points to the "ユーザ名" entry with the text "Branches にユーザー名があればOK" (If the user name is in the branches, it's OK).

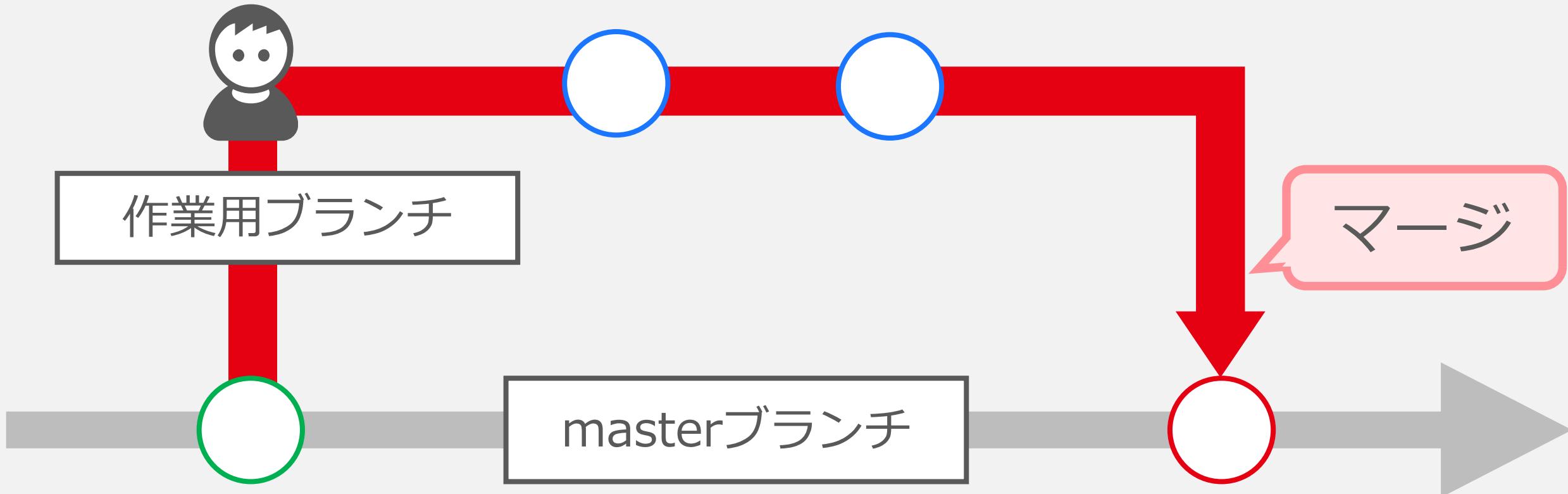
Code → master をクリック

Branches にユーザー名があればOK

# マージ / Pull Request

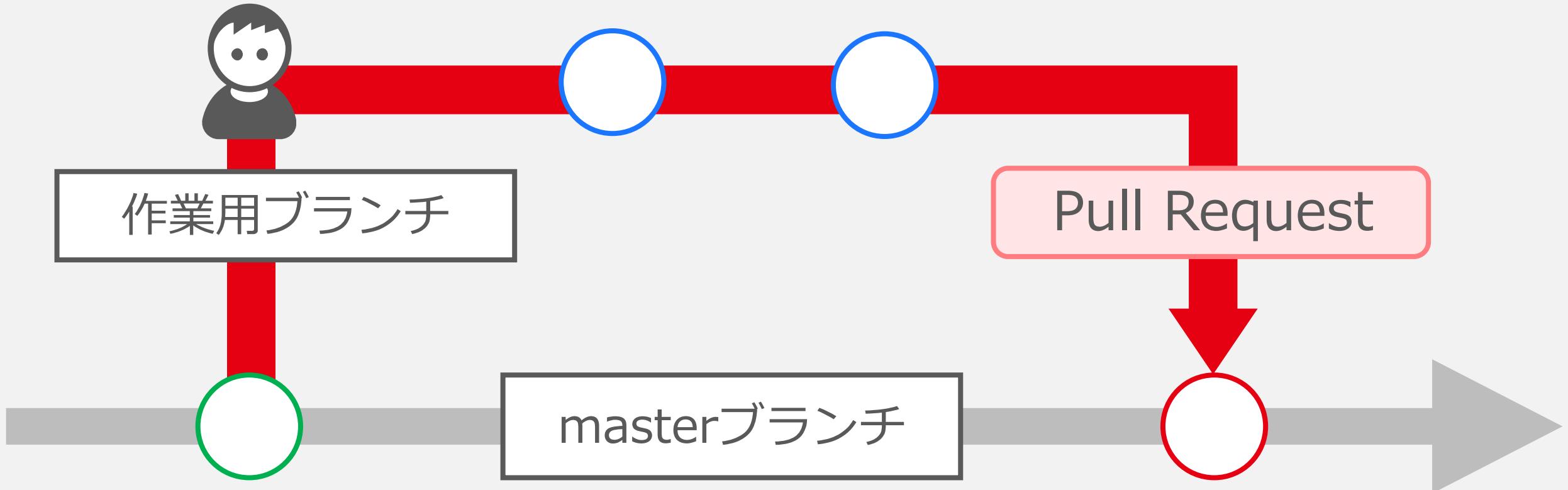
# マージ

マージ：ブランチの変更をmasterブランチに反映する



# Pull Request (PR)

マージする前に Pull Request (PR) を作成する



# Pull Request (PR)

## ● Pull Request (PR)

- ・ マージしてOKかコードレビューするためのもの
- ・ GitHubのコミュニケーションツールとしての側面



# 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

ユーザーネーム Add README.md

src

README.md

index.html

README.md

作業ブランチをpushした後、GitHubにいくとこのような画面が出てきます（青枠）

※もし出てない場合は 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 dropdown menus for "master", "2 branches", and "0 tags", along with buttons for "Go to file", "Add file", and "Code". A large blue callout box with white text says "Compare & pull request を押す". The main content area shows a list of recent commits:

	ユーザーネーム Add README.md	
	src Add index.html & src	1 hour ago
	README.md Add README.md	39 minutes ago
	index.html Add README.md	39 minutes ago

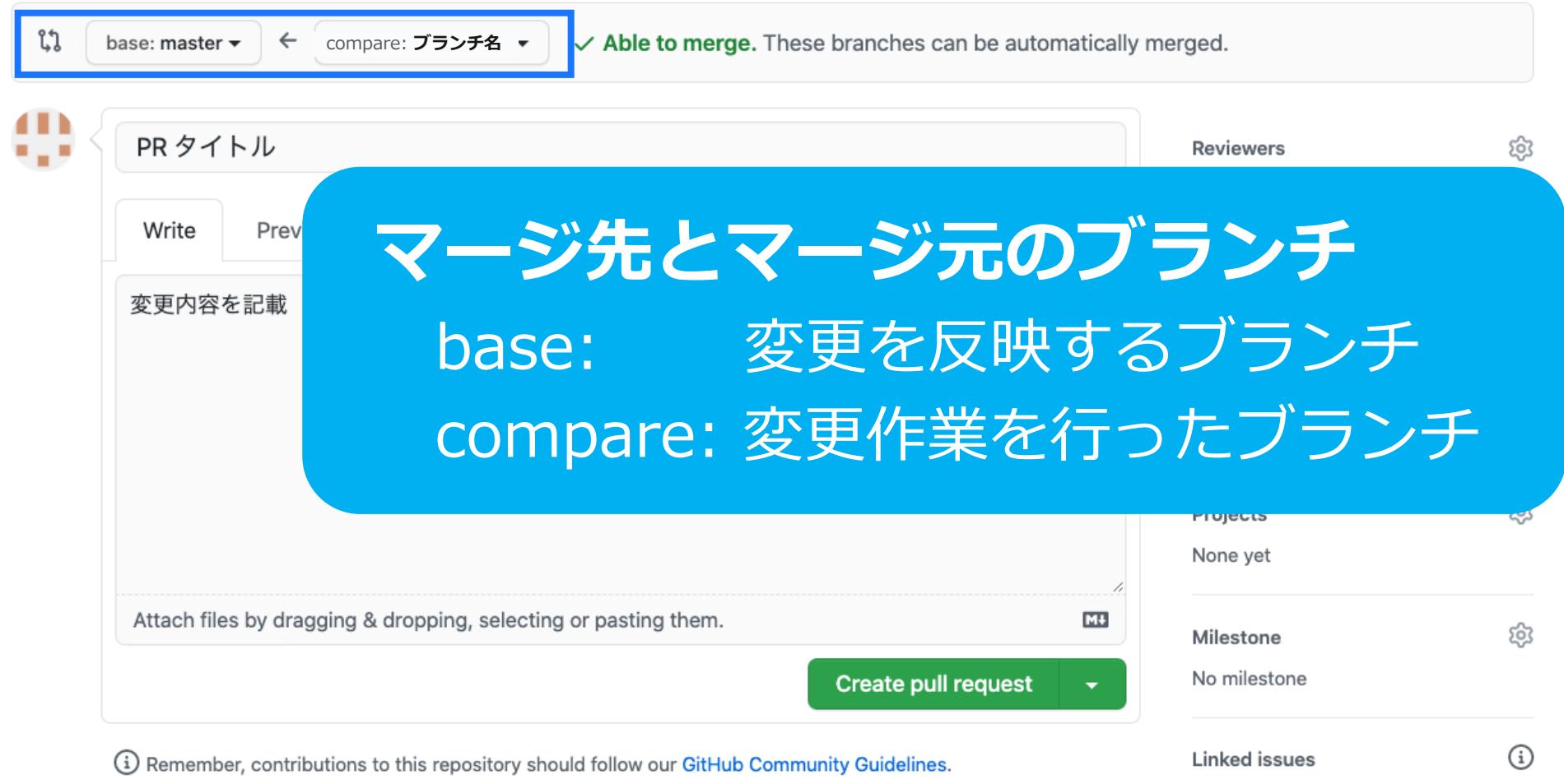
At the bottom, there is a single commit for "README.md" with a edit icon.

# 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 this, there's a large blue callout box containing Japanese text explaining the 'base' and 'compare' parameters.

マージ先とマージ元のブランチ

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

PR タイトル  
Write Prev  
変更内容を記載  
Attach files by dragging & dropping, selecting or pasting them.  
Create pull request

Reviewers  
Projects  
None yet  
Milestone  
No milestone  
Linked issues

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

Linked issues



# 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. A large blue callout box covers the middle section, containing the text "Pull Request のタイトル" and "→ 概要がひと目でわかるように". To the right of the title field are buttons for "Write" and "Preview", and a toolbar with various text format icons. Below the title field is a placeholder "Attach files by dragging & dropping, selecting or pasting them." At the bottom is a green "Create pull request" button. On the right side of the screen, there are sections for "Reviewers" (No reviews), "Projects" (None yet), "Milestone" (No milestone), and "Linked issues". At the very bottom, a note says "Remember, contributions to this repository should follow our [GitHub Community Guidelines](#)".

PR タイトル

PR 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

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 displays the base branch as 'master' and the compare branch as 'ブランチ名'. A green checkmark indicates that the branches are 'Able to merge'. The main area contains a title field ('PR タイトル') and a rich text editor ('Write' tab selected) with 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), and 'Milestone' (No milestone). A large blue button at the bottom is labeled 'Pull Request作成ボタン' (Create pull request). A note at the bottom left reminds users to follow GitHub Community Guidelines.

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

PR タイトル

Reviewers

No reviews

Assignees

No one—assign yourself

Labels

None yet

Milestone

No milestone

Linked issues

Pull Request作成ボタン

Create pull request

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

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

# Pull Request の作成方法

The screenshot shows a GitHub repository interface. At the top, there's a navigation bar with links: Code, Issues, Pull requests 1 (which is highlighted), Actions, Projects, Wiki, Security, Insights, and Settings. Below the navigation bar, a pull request card is displayed with the title "PR タイトル #1". A green "Open" button is visible. The card indicates a merge from a user's branch into the "master" branch. It shows 0 conversations, 1 commit, 0 checks, and 2 files changed. A comment from a user named "ユーザー名" is shown, stating "commented now" and "変更内容を記載". A blue callout bubble with the text "Pull Requestが作成された" (Pull Request was created) is overlaid on the right side of the card. Below the card, there's a note: "Add more commits by pushing to the ユーザー名 branch on ユーザー名". At the bottom, there's a green "Merge pull request" button and a note: "You can also open this in GitHub Desktop or view command line instructions." A green box highlights the "Continuous integration has not been set up" section, which mentions GitHub Actions and other apps for catching bugs and enforcing style. Another green box highlights the "This branch has no conflicts with the base branch" section, noting that merging can be performed automatically.

PR タイトル #1

Open ユーザー名 Wants to merge 1 commit into master from ユーザー名

Conversation 0 Commits 1 Checks 0 Files changed 2

ユーザー名 commented now Owner 😊 ...

変更内容を記載

Update index.html

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

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.

Pull Requestが作成された

# 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 に変更差分が表示される

Line	Change Type	Content
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	+	<div id="target">ここをクリックするとアラート発生</div>
13	-	<!--<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	-	<script src=".src/index.js"></script>
16	+	<script src=".src/index.js"></script>
17	-	</body>

# Pull Request の確認

The screenshot shows a GitHub Pull Request interface. At the top, there's a commit message: "Update index.html" with hash "78149a0". Below it, a note 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 two hands is next to a box containing CI status: "Continuous integration has not been run by GitHub Actions and several other apps". Below that, a green checkmark icon indicates "This branch has no conflicts with your base branch". A green button says "Merge pull request". A blue callout bubble points to this button with the Japanese text "気になる点がある場合はコメント欄から記入する". At the bottom, there's a "Write" tab, a "Preview" tab, and a large text area with placeholder text "レビューや気になった内容はコメントできる". Below this is a file attachment section with placeholder text "Attach files by dragging & dropping, selecting or pasting them.". At the very bottom are two buttons: "Close with comment" and a larger green "Comment" button.

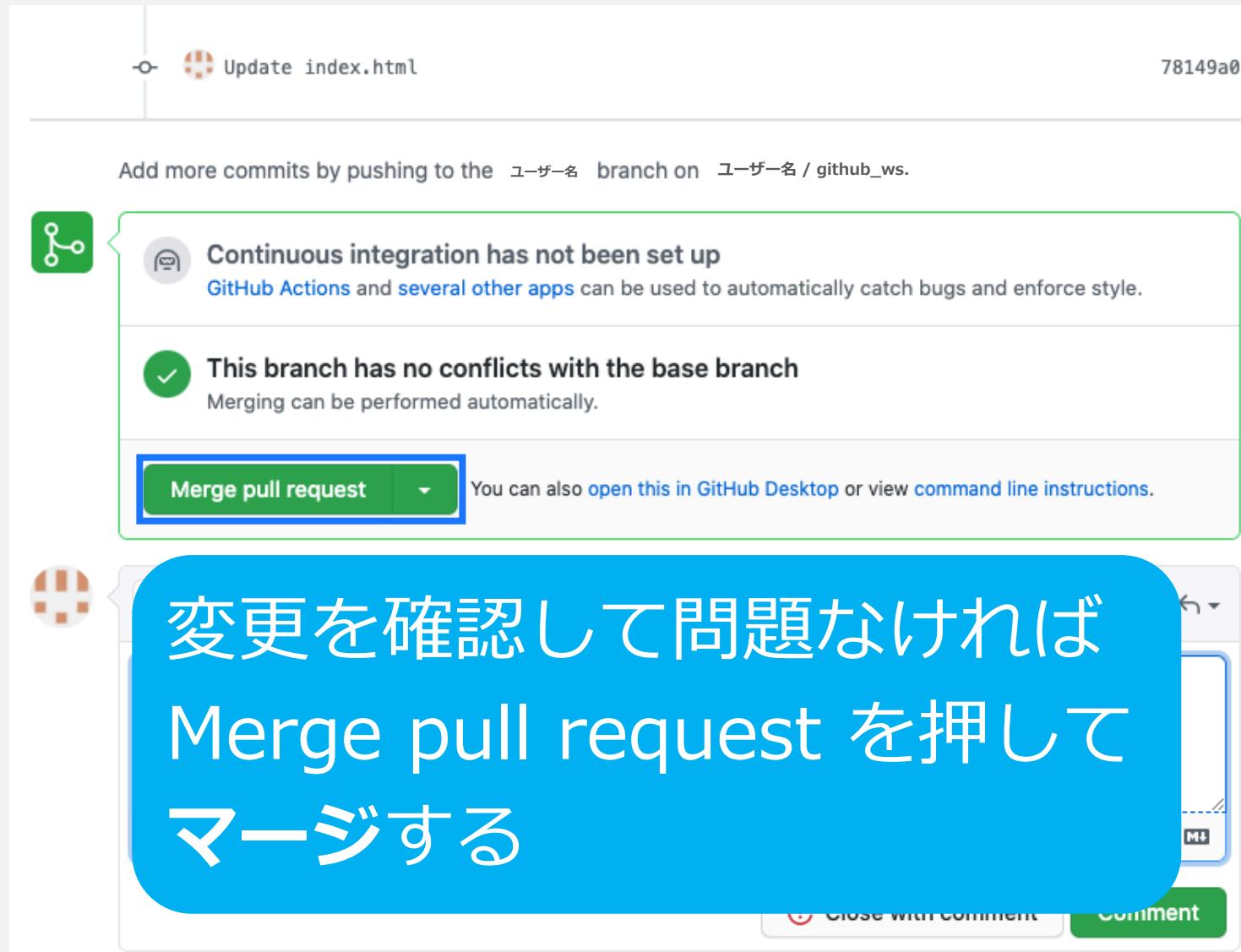
気になる点がある場合は  
コメント欄から記入する

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 をマージ



# ローカルリポジトリへのpull

YAHOO!  
JAPAN

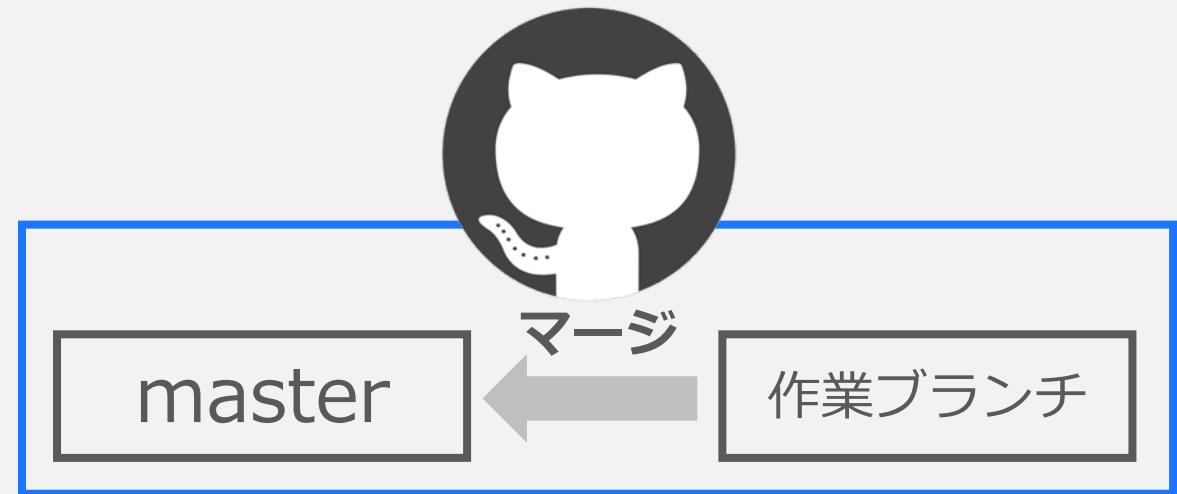
# マージされたものをローカルに反映

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



ローカルリポジトリ

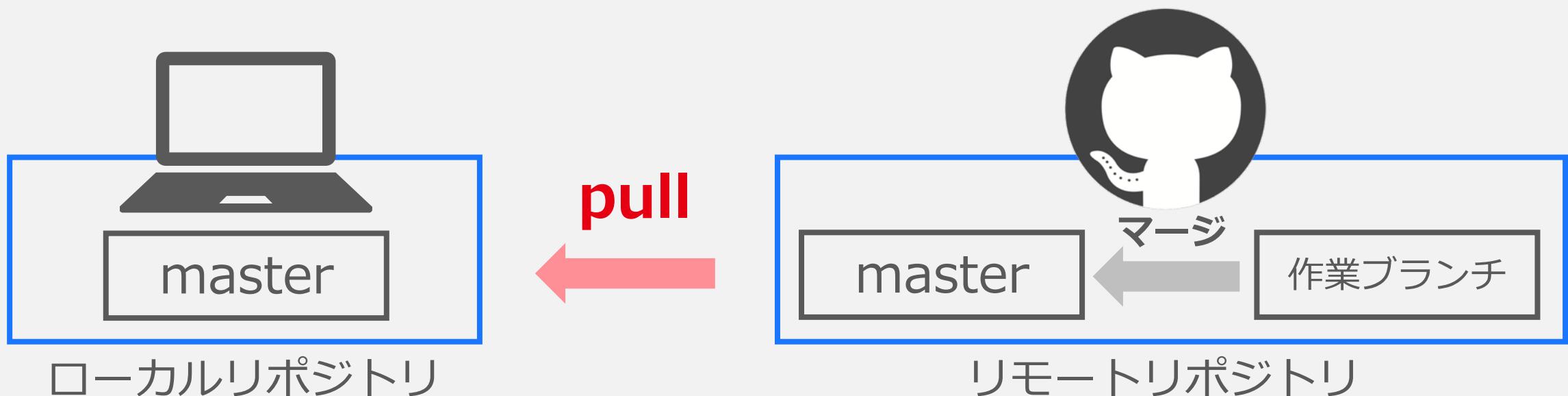


リモートリポジトリ

# マージされたものをローカルに反映

- **pull**

リモートの master の内容をローカルの master に反映



# GUIコマンド - pull

- \$ 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
```

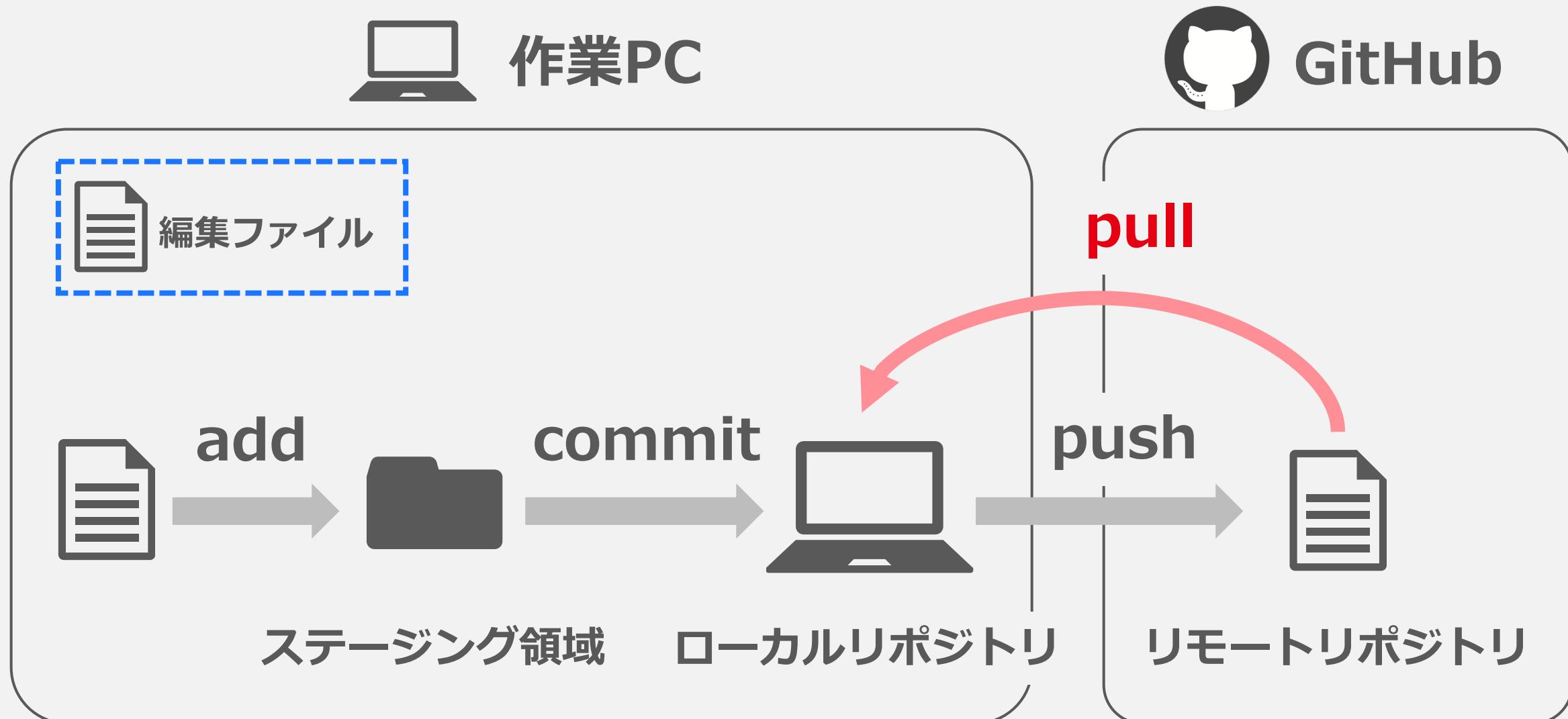
# GUIコマンド - pull

- \$ git pull でリモートリポジトリの変更を  
ローカルリポジトリに反映

```
$ git pull
```

- ローカルの該当ファイルを開く  
→ ファイルが変更されていることを確認

# ローカルリポジトリの変更



# ローカルリポジトリの変更

**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

**pull**

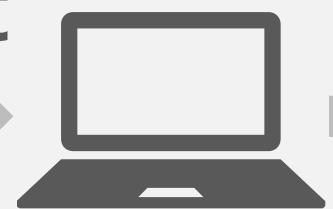
**push**



**add**



**commit**



ステージング領域

ローカルリポジトリ



リモートリポジトリ

# 第二部

# ハンズオン

YAHOO!  
JAPAN

# 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

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 するファイルに注意（**秘密情報を漏らさない**）
- ・ commit 時には必ずコメントを残す

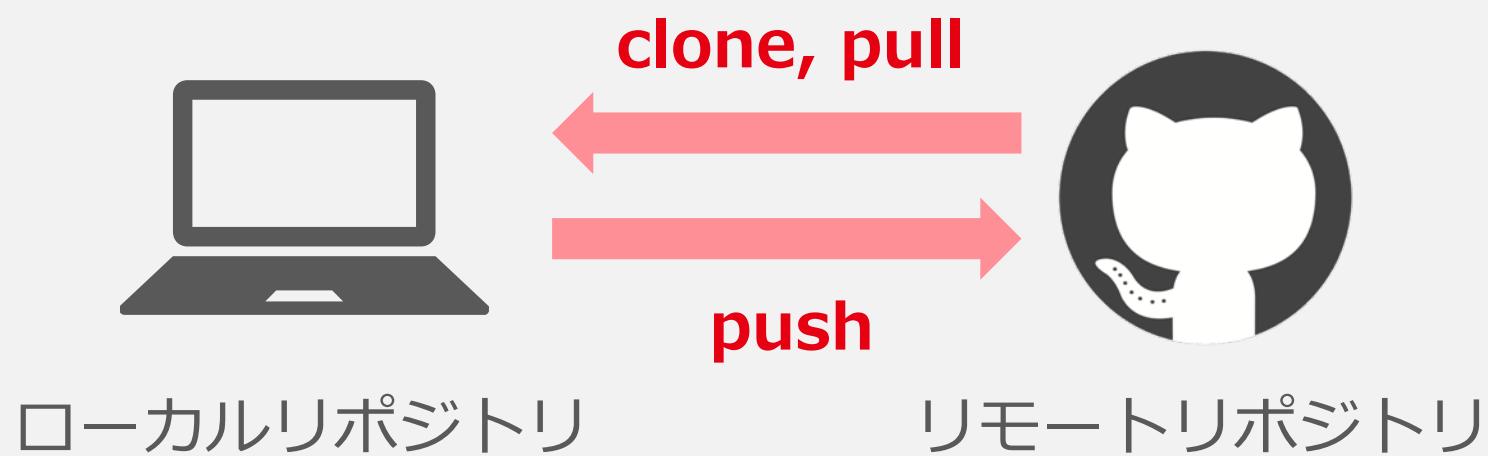


# 【おさらい】ローカルとリモートの関係

**clone** : リモートをローカルにコピー (初回のみ)

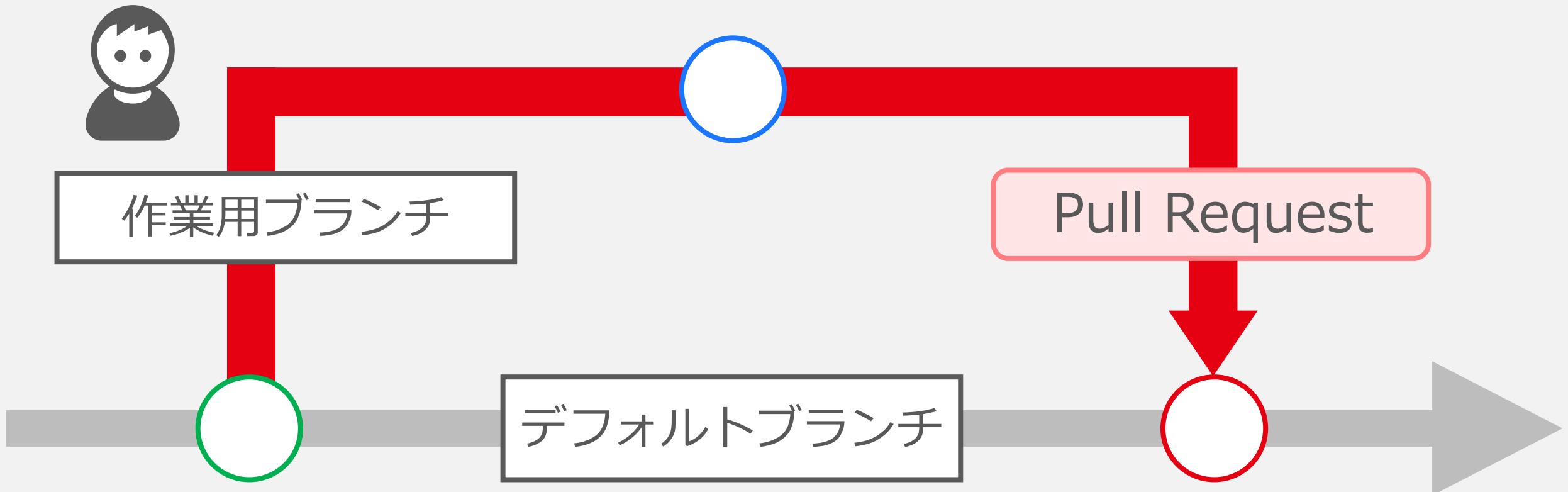
**pull** : リモートの変更をローカルに反映

**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入門レベル
- ・ 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。

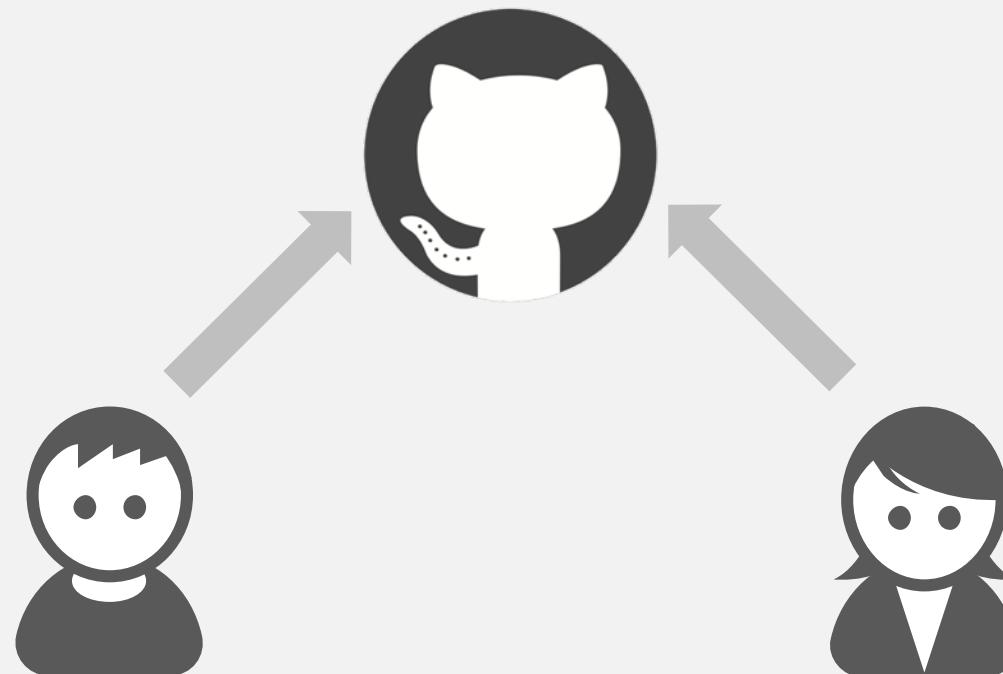
まずは基本的な使い方に慣れる。  
使っていく中で調べながら使う。

# GitHub を複数人で使う

YAHOO!  
JAPAN

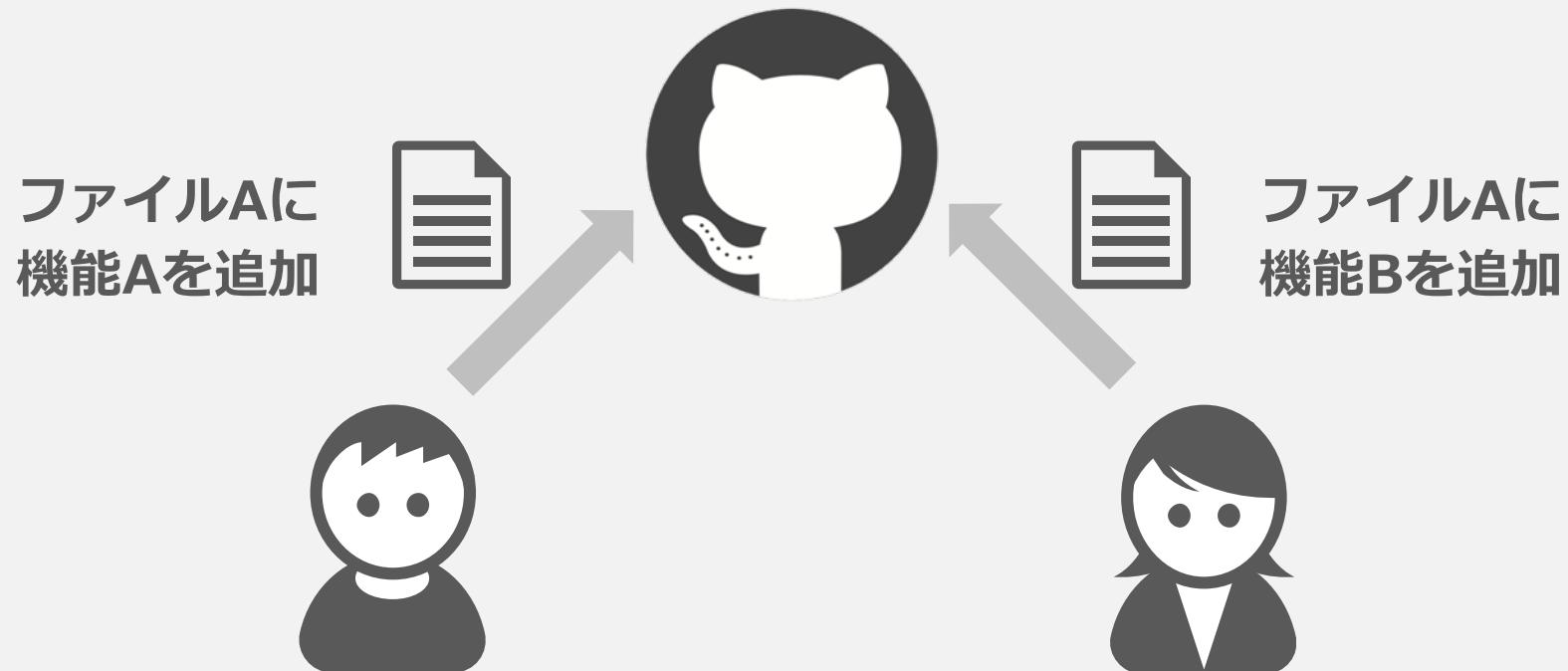
# チーム開発では

- 1つのリポジトリに対して複数人で作業する



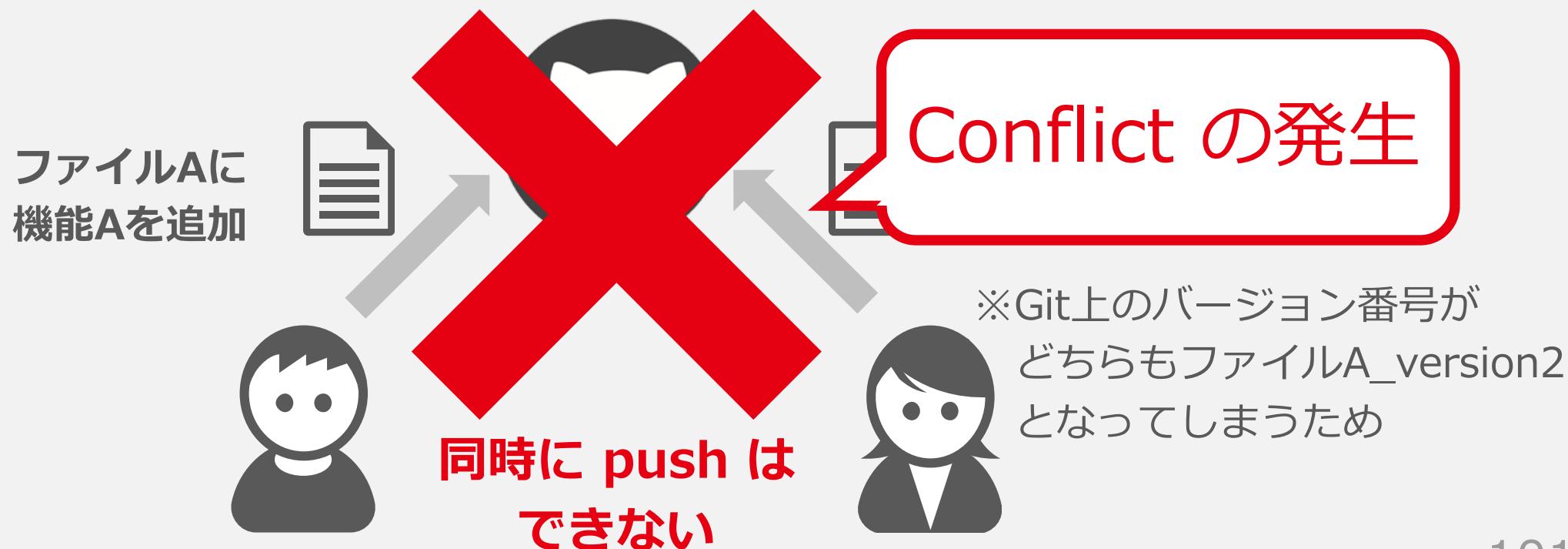
# チーム開発では

- ・ 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

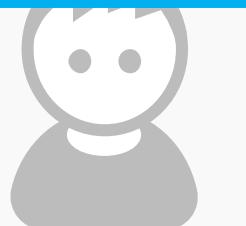
- ・ 1つのリポジトリに対して複数人で作業する
- ・ 同じファイルを同時に修正してしまう可能性も



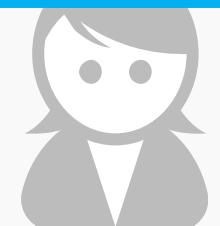
# チーム開発では

- ・ 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



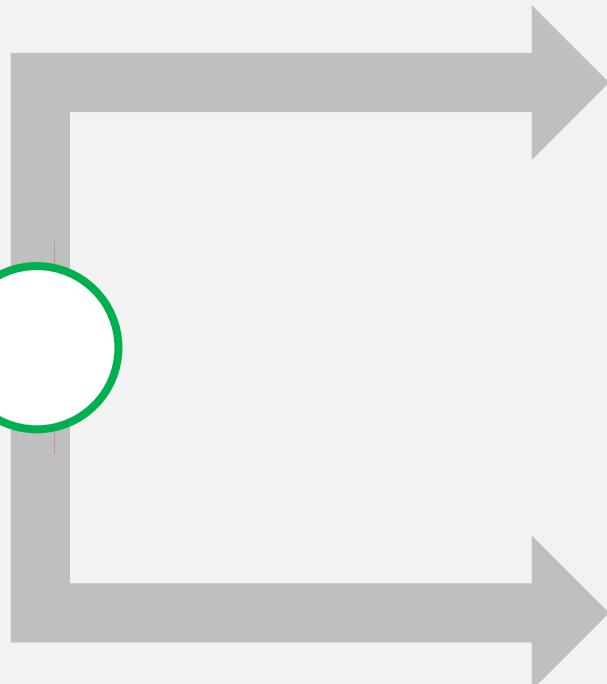
同時に push は  
できない



# 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

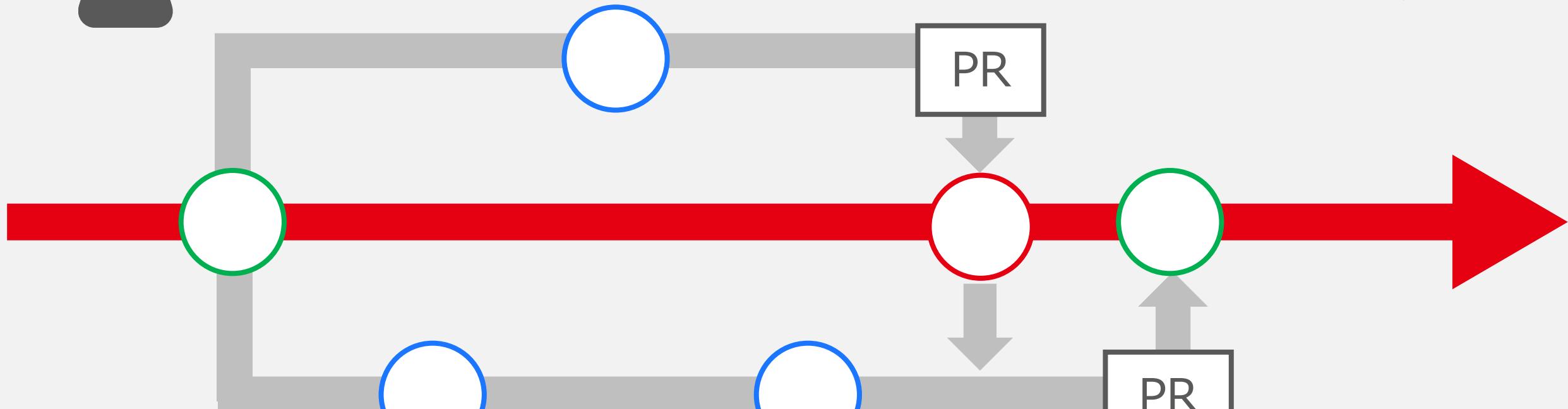
作業者がそれぞれブランチを作成

# 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」

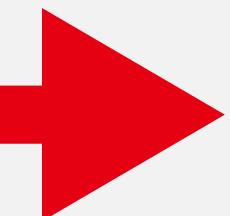
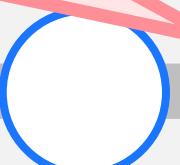
作業者がそれぞれブランチを作成  
→ PRでコードレビューをしてマージ



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

# 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
```