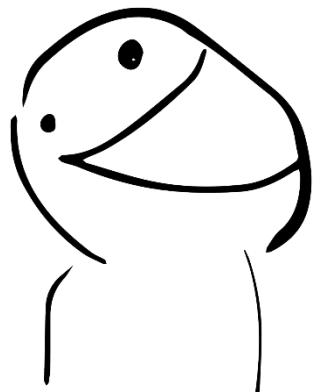


講習会 第2回 git&GitHub編

担当:谷澤悠太

スライドはGitHubに上がってるので確認してください
https://github.com/Tohoku-University-Takizawa-Lab/study_session/tree/main/2023/2nd_git_GitHub/slides



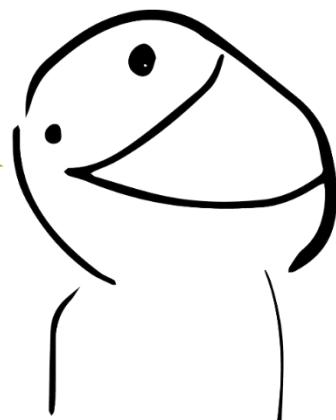
目次

- **Introduction**
- **What is git?**
- **Environment**
- **Tutorial git (by yourself)**
- **Tutorial git (cooperation)**

少し難しそうな日本語にはdeepL英語訳を紫文字で付けました
多少ニュアンスが違っても温かい目で見てください

I've added a deepL English translation in **purple letters** for the
Japanese that looks a little difficult.

Please look at it warmly, even if the nuance is a little different.



Introduction

こんな経験、ありませんか？

Have you ever had this experience?

- **CASE I : ファイルを編集**

- しばらくするとファイルが増えすぎてどれがなんのファイルかわからなくなる.....

- there are so many files that you don't know which one is what...

名前
120525_ドキュメント_最新.txt
120602_ドキュメント.txt
120604_ドキュメント.txt
120605_ドキュメント_修正版.txt
120605_ドキュメント_江口.txt
120605_ドキュメント_最新 のコピー.txt
120605_ドキュメント_最新.txt
120605_ドキュメント.txt
ドキュメント_会議用.txt



もうどれが何だか
わからんばい...

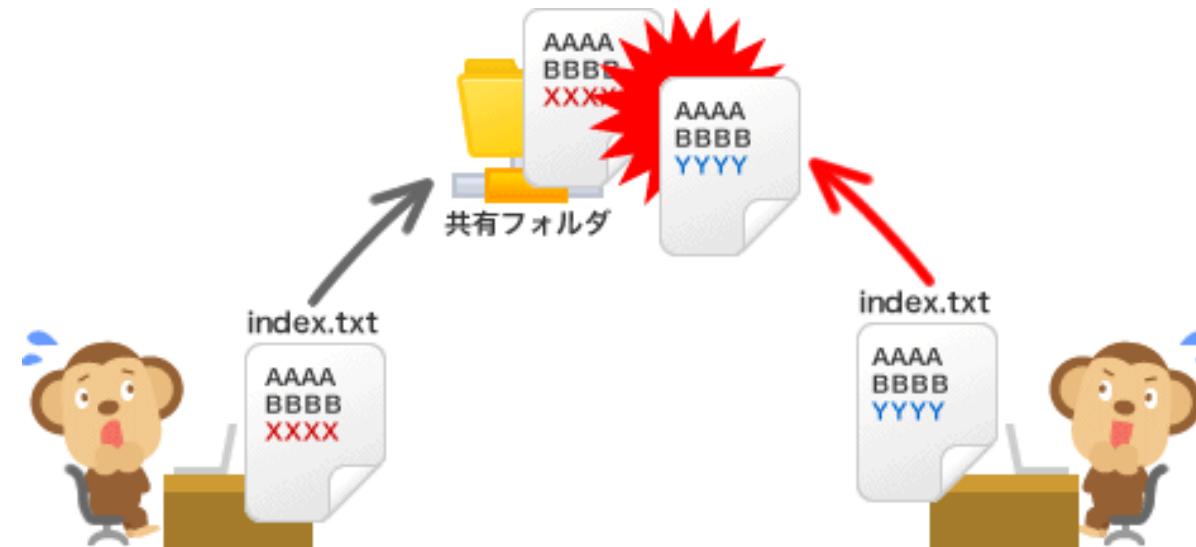
こんな経験、ありませんか？

Have you ever had this experience?

- CASE II : 共有フォルダ内の編集

誰かと編集が競合して、わけが分からなくなる.....

Editing conflicts with someone else, and it gets confusing...



こんな経験、ありませんか？

Have you ever had this experience?

- CASE II : 共有フォルダ内の編集

誰かと編集が競合して、わけが分からなくなる.....

Editing conflicts with someone else, and it gets confusing...

“git” can solve these problems!



Today's goal

Goals of the 2nd STUDY SESSION

gitとは?

- バージョン管理
version control
- 複数人での共同作業
Collaboration with multiple people

これらを簡単にするためのツール

tools to make these easy

・目標 Goal

- gitを使ったバージョン管理ができるようになる！
Learn to use git for version control.
- GitHubをつかって複数人での共同作業ができるようになる
To be able to collaborate with multiple people using GitHub

What is Git?

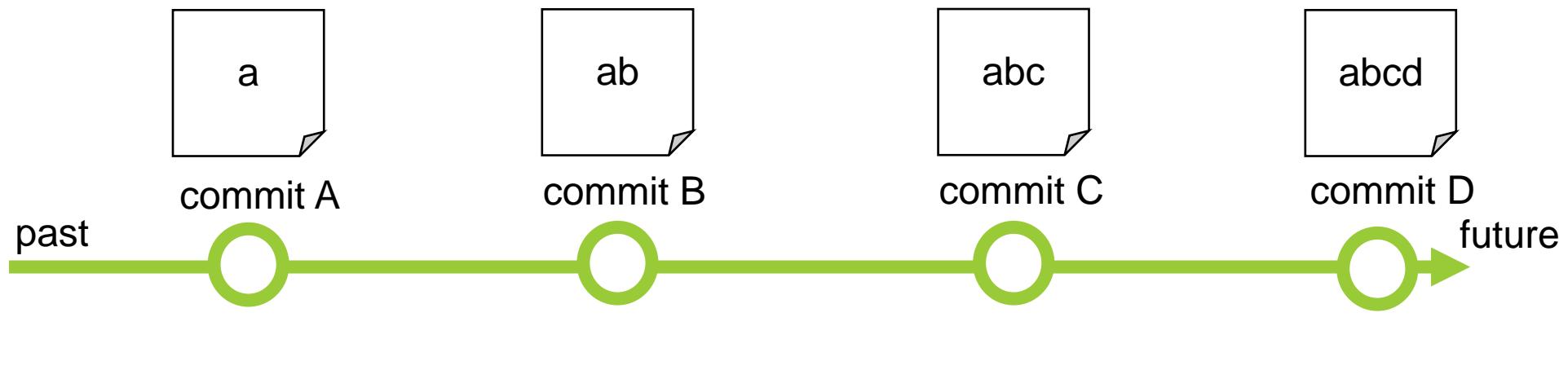
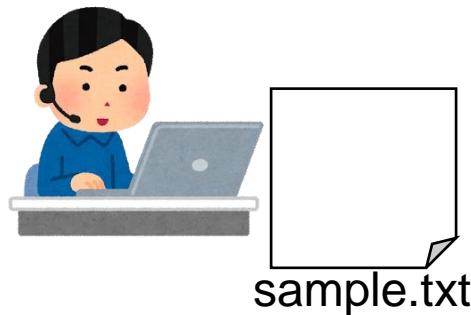
単語 : commit

- 過去の情報を保存 Save past information

- セーブポイント=commit
- 少し変えるたびにcommitするといい

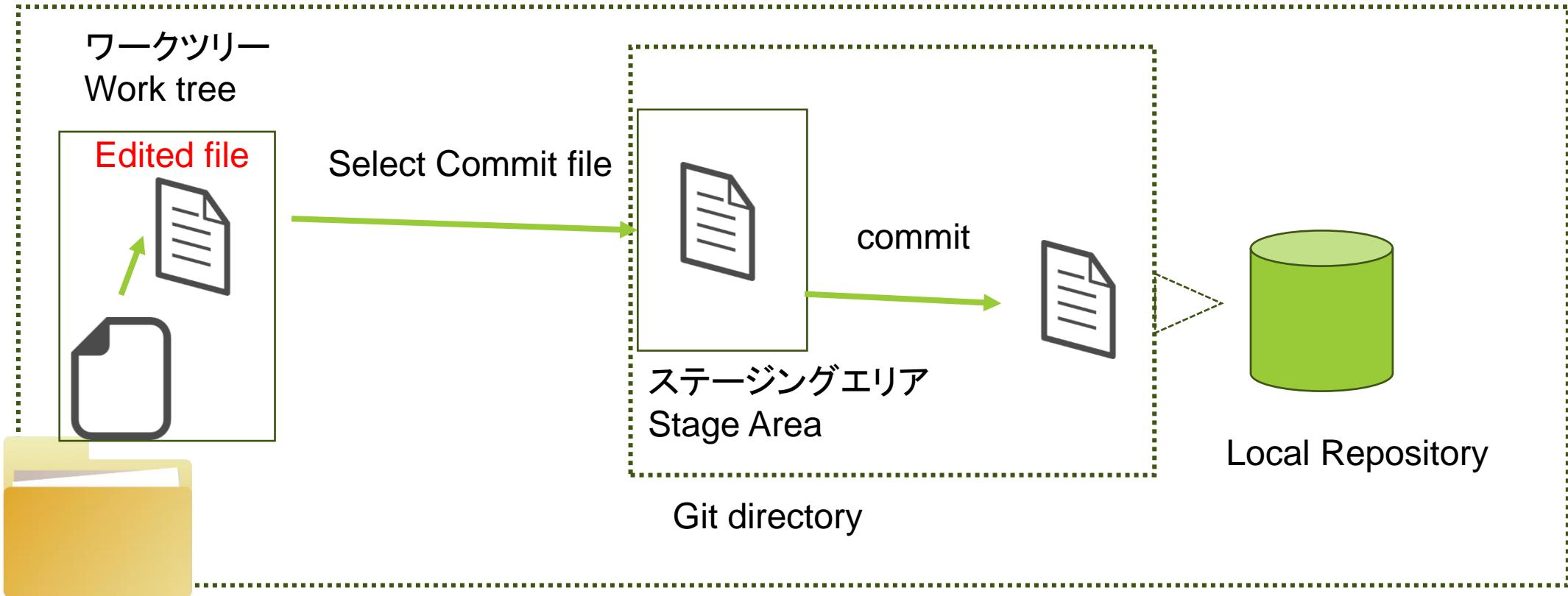
You should commit every time you change it a little.

セーブする -> コミットする
セーブしたポイント -> コミット



ローカルリポジトリに対する操作

Operations on local repository



Gitで管理しているフォルダ
Folders managed by Git

リモートリポジトリに対する操作

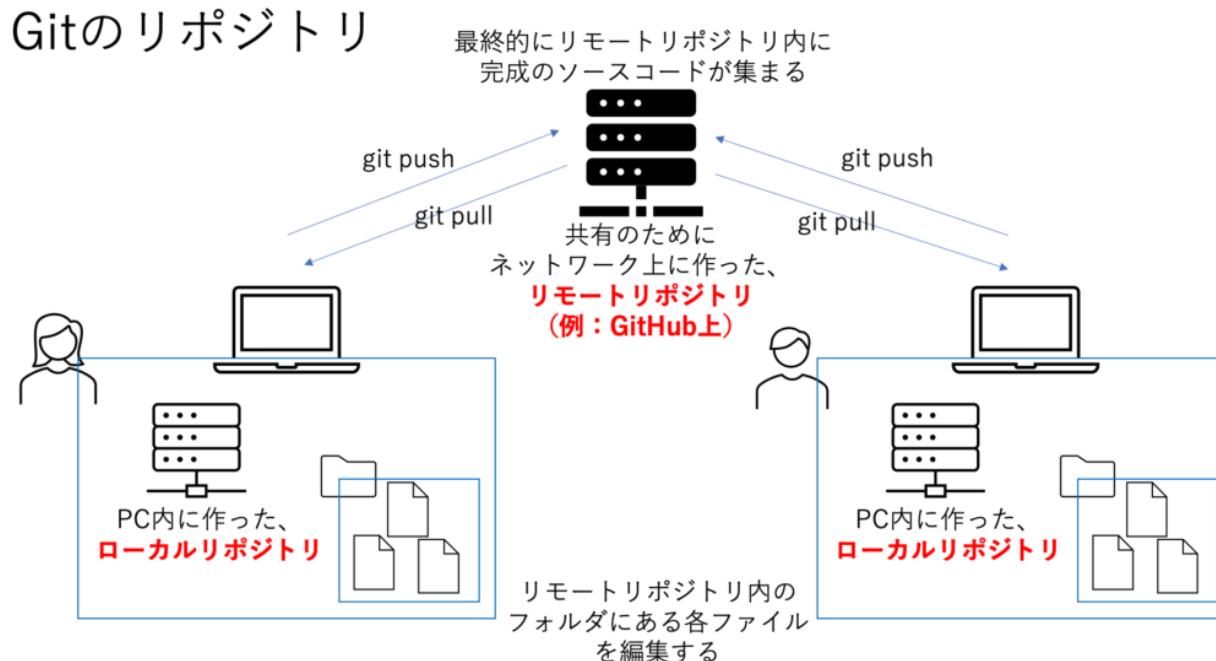
Operations on remote repository

- **gitで管理するデータの保存場所**
 - gitはこの中のファイルを管理してくれる
- **共有リポジトリとローカルリポジトリ**

Where to store the data managed by git

git will manage the files in the repository

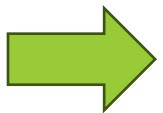
There is a shared repository and a local repository.



Environment

Git Graph install

Open VSCode



①

② select Git Graph

The screenshot shows the Visual Studio Code interface. On the left, the 'Extensions' sidebar is open, displaying the 'Market...' tab. A red box highlights the 'Git Graph' extension by mhutchie, version 1.30.0, which is described as 'View a Git Graph of your repository, and perform Git actions from the graph.' Below the extension details, there is a large green 'Install' button. Another red box highlights this 'Install' button. To the right of the extensions list, the main workspace shows the 'Git Graph' extension installed. It displays a git log graph for a repository named 'demo - Git Graph'. The graph visualizes commit history, branches, and merges. The bottom of the screen shows the standard VSCode navigation bar with tabs for PROBLEMS, OUTPUT, TERMINAL, PORTS, and DEBUG CONSOLE.

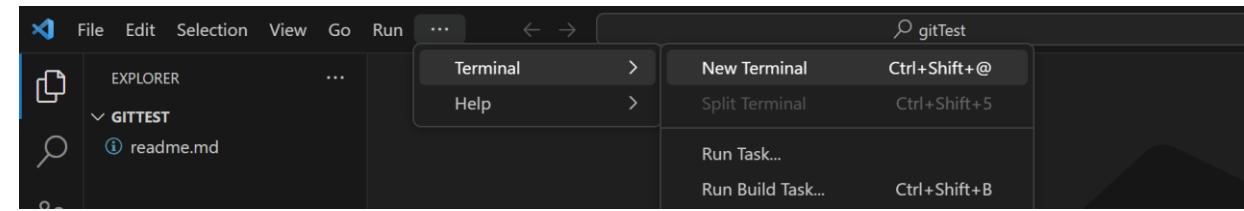
Configを設定 (Set Config)

1. Open VsCode

2. Open Terminal (Ctrl + @)

3. ターミナルに以下のコマンドを入力

Enter the following command in the terminal



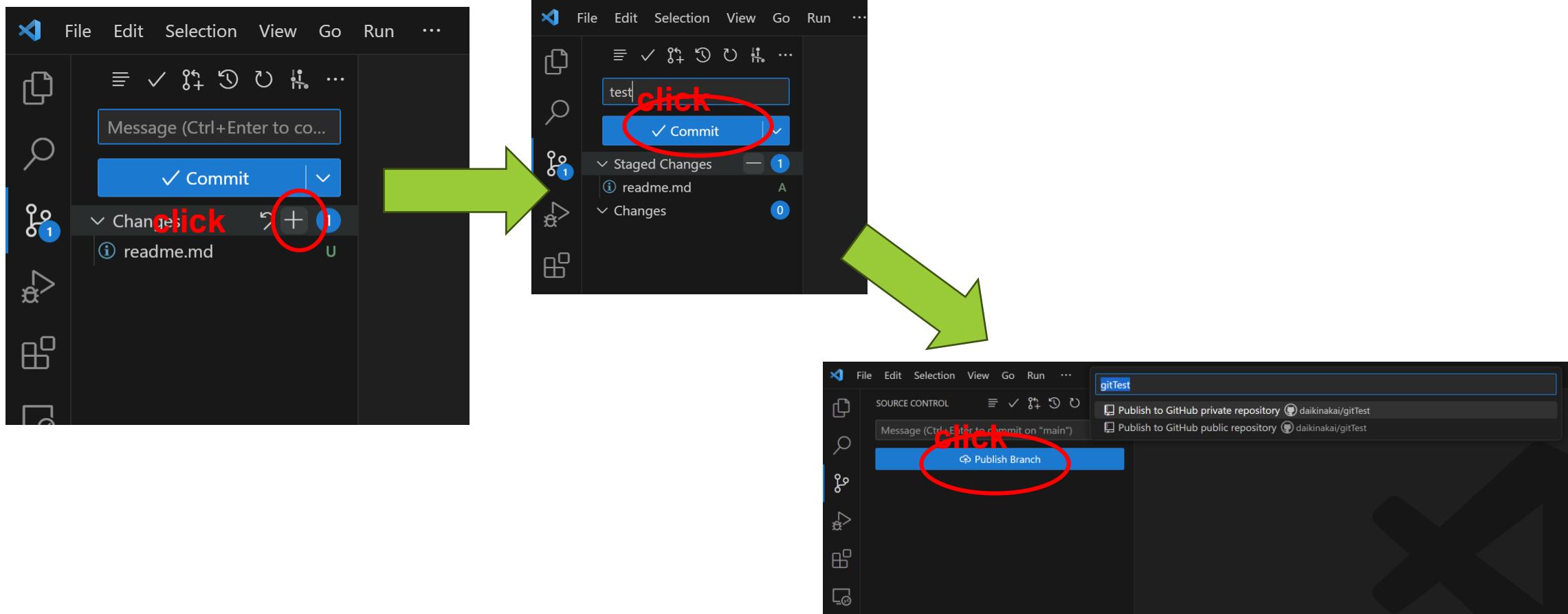
```
daiki@LAPTOP-J2DU4152 MINGW64 /c/gitTutorial (main)
$ git config --global user.name [gitUserName]
```

```
daiki@LAPTOP-J2DU4152 MINGW64 /c/gitTutorial (main)
$ git config --global user.email [gitUserEmail]
```

Prepare

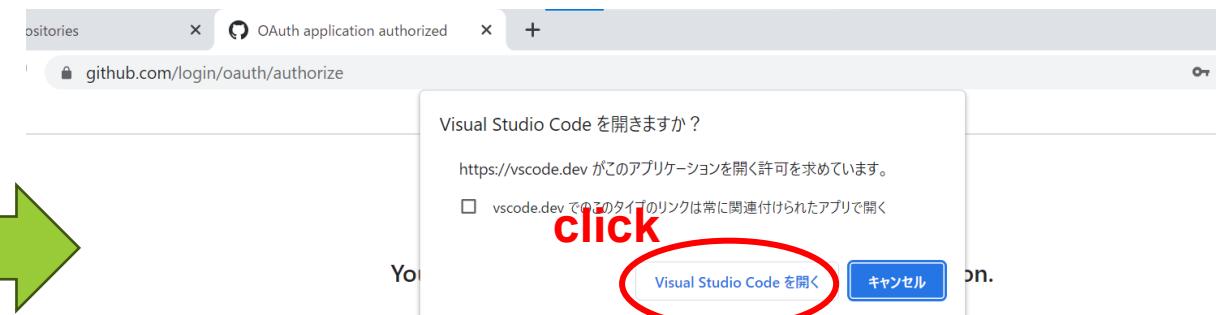
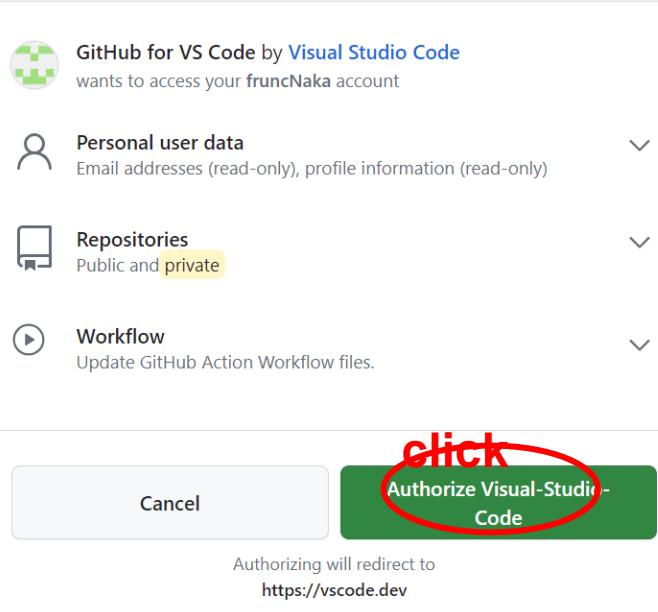
- Create Folder 「gitTest」
- Open 「gitTest」 folder in vscode
- Create File 「readme.md」

動作確認 (Confirmation of operation)

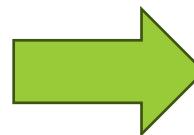
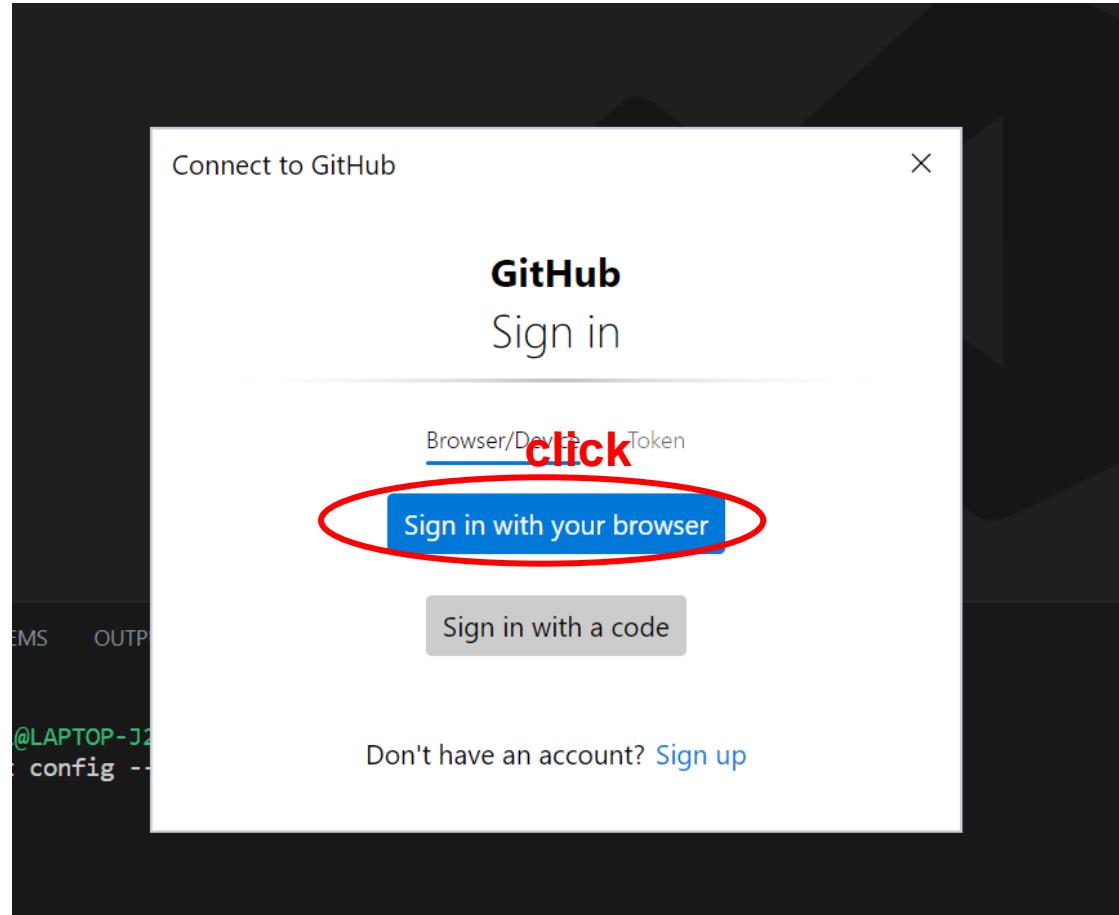


動作確認 (Confirmation of operation)

Authorize GitHub for VS Code



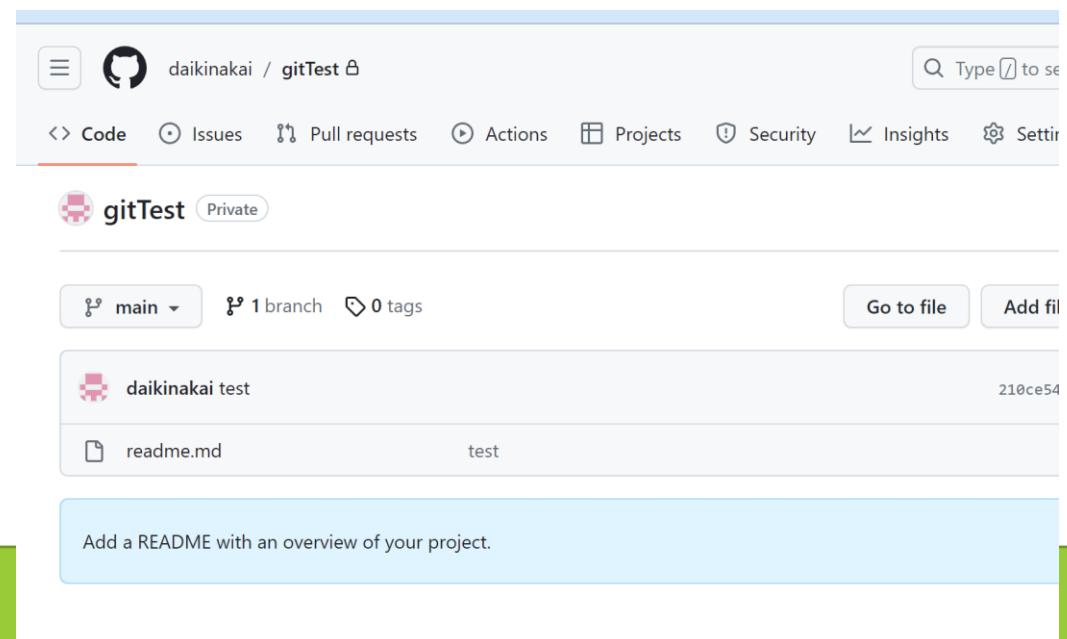
動作確認 (Confirmation of operation)



ユーザ名とパスワードの
入力
**Enter username and
password**

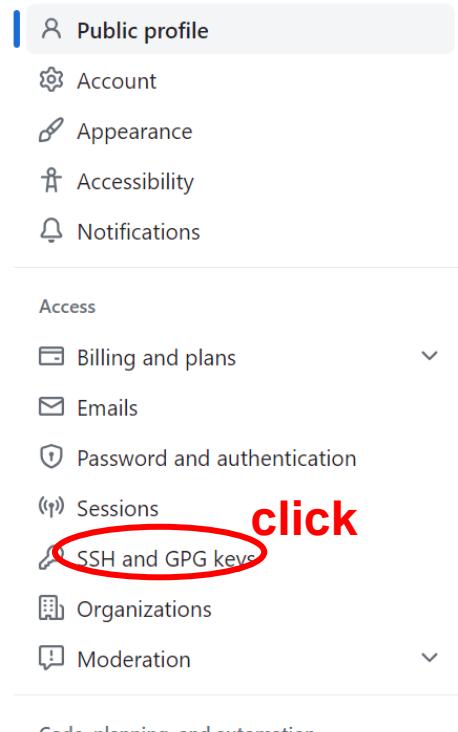
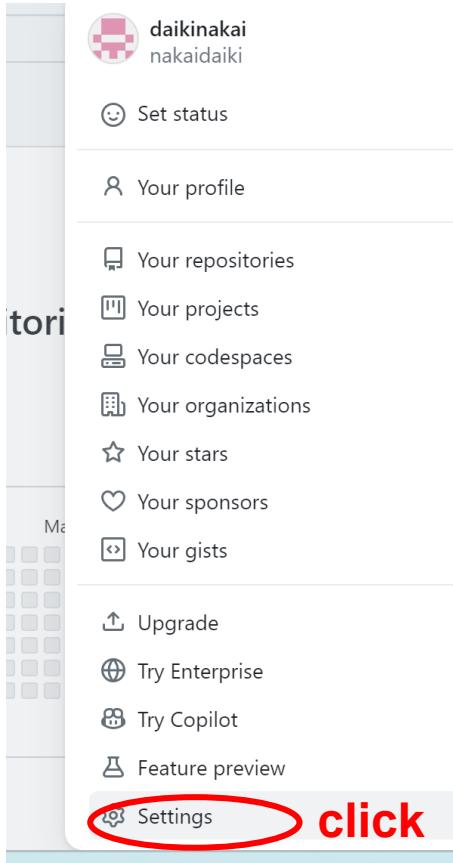
動作確認 (Confirmation of operation)

- 自身のgithubアカウントを開く
- Repository部分から「gitTest」を開き、確認
- Open your own github account
- Open "gitTutorial" from the Repository section and check it.



鍵登録 (key registration)

1. Open your own github account



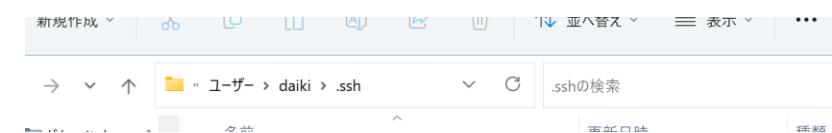
SSH keys

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

4. copy id_ed25519.pub & paste
5. add SSH key



"id_ed25519.pub"はおそらくここにある
The "id_ed25519.pub" is probably here



click

New SSH key

動作確認 (Confirmation of operation)

```
daiki@LAPTOP-J2DU4152 MINGW64 /c/gitTutorial (main)
$ ssh -T git@github.com
The authenticity of host 'github.com (20.27.177.113)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? 
Enter passphrase for key '/c/Users/daiki/.ssh/id_ed25519':
Hi fruncNaka! You've successfully authenticated, but GitHub does not provide shell access.
```

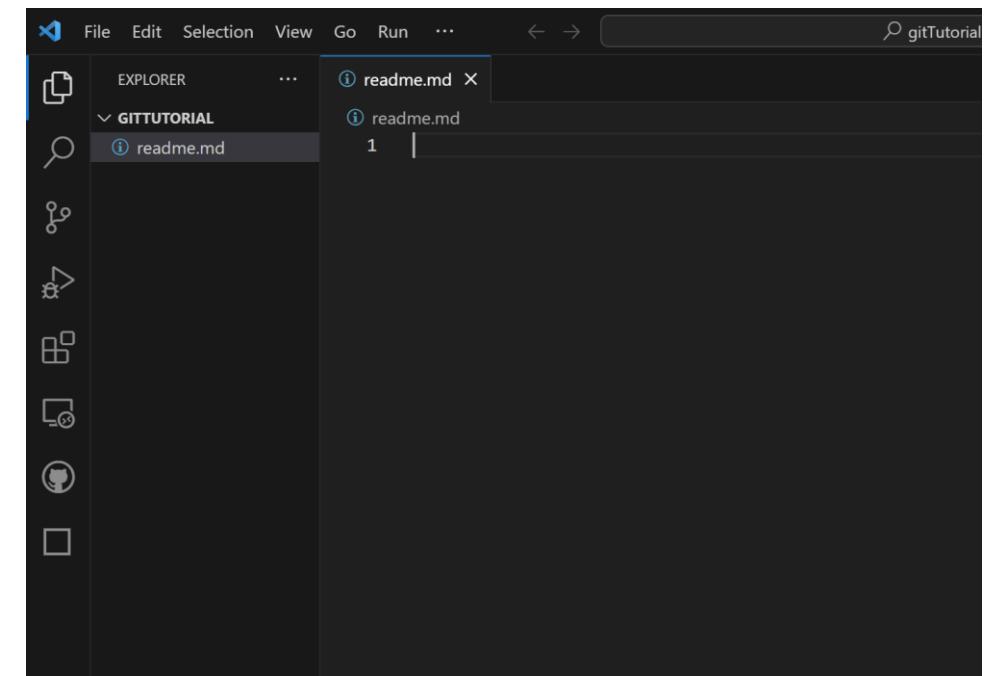
Tutorial git (by yourself)

ひとりでgit勉強編の目的 main purpose of this section

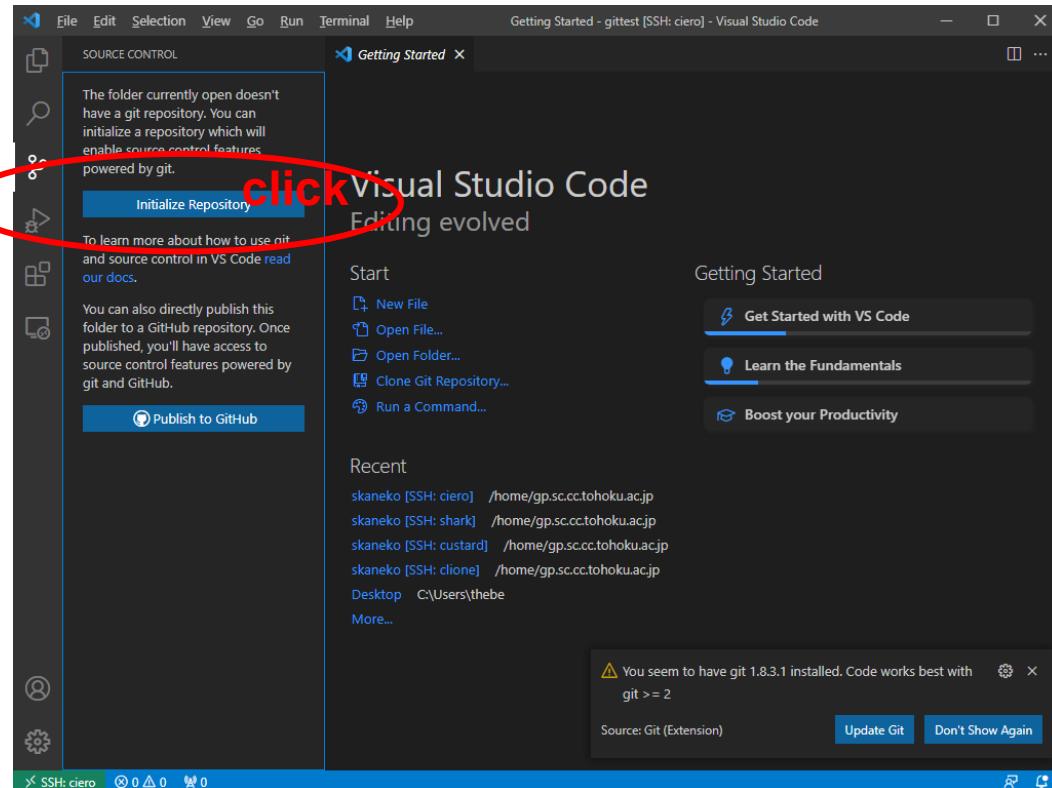
- gitの主要な概念と単語について理解する
- Understand the concepts and vocabulary of git
 - branch, repository
 - clone, pull, commit, push
- gitを使って簡単なファイル管理を試す
- Try simple file management with git

Prepare

- Create Folder 「gitTutorial」
- Open 「gitTutorial」 folder in vscode
- Create File 「readme.md」



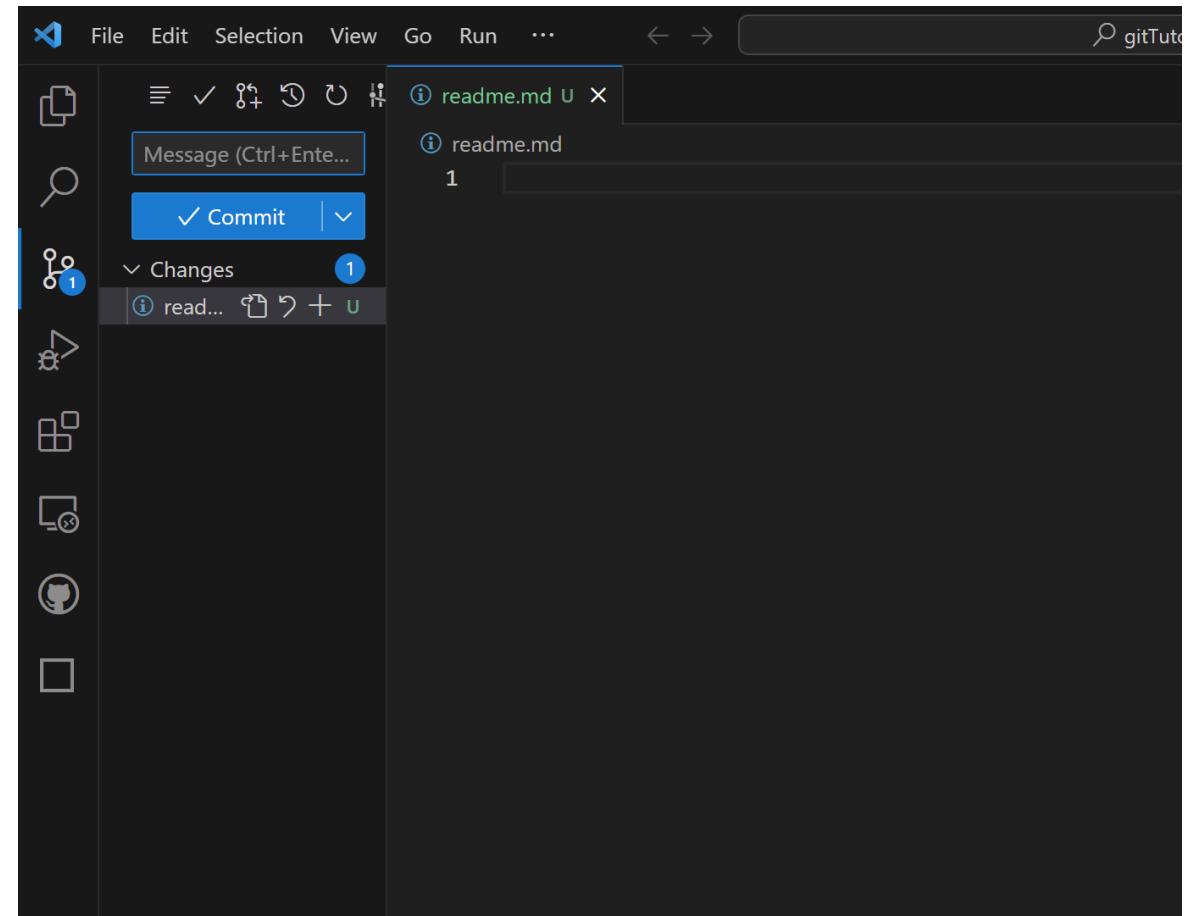
Initialize Repository



Initialize Repository

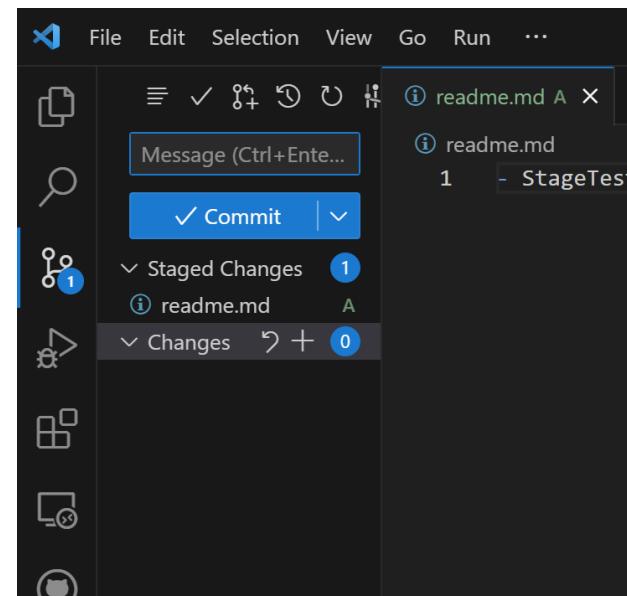
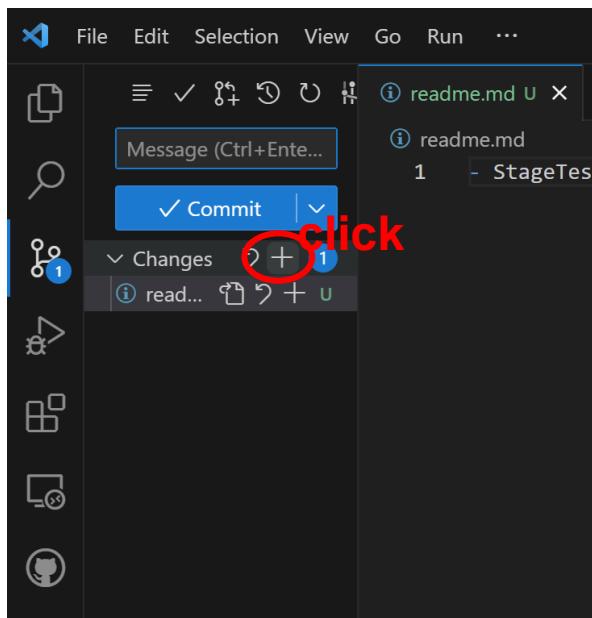
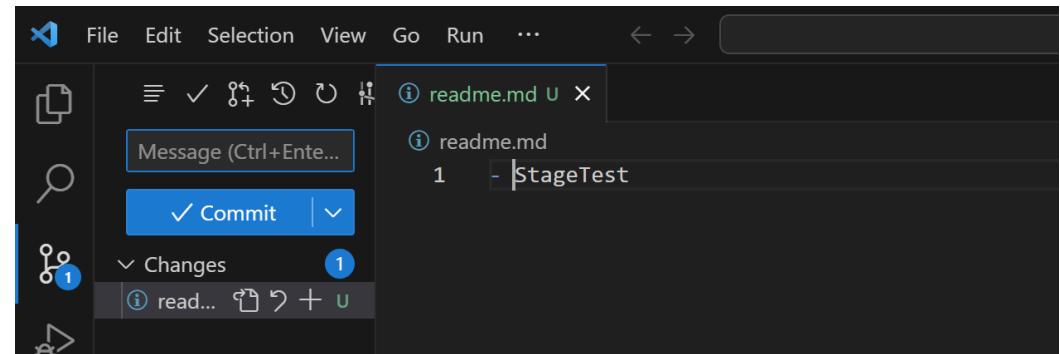
- 成功するとこんな感じ

Here's what it looks like when you succeed



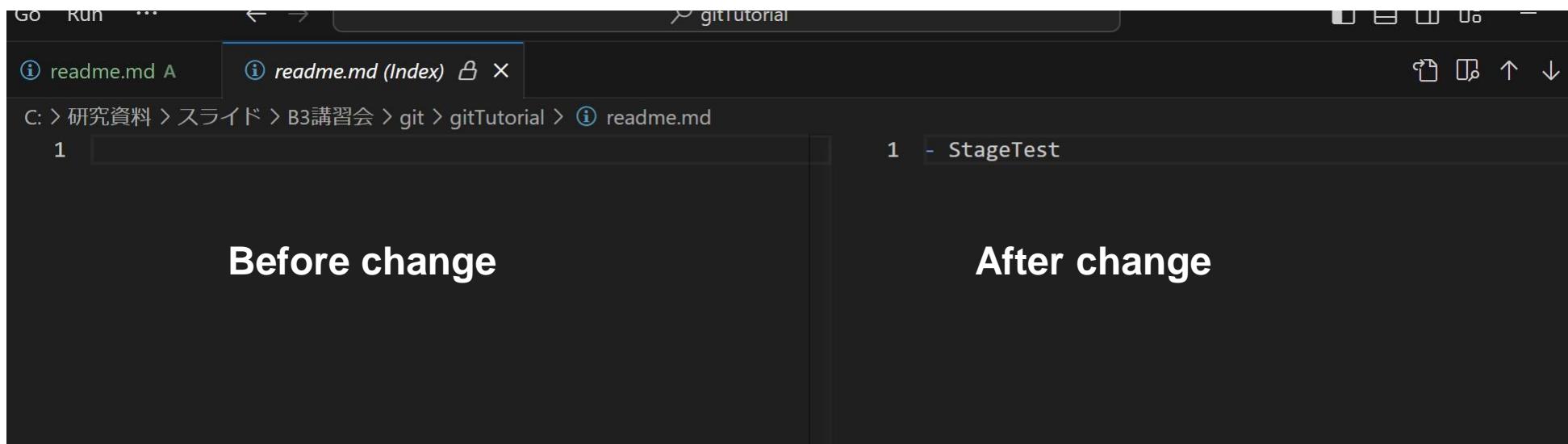
add to the stage

- Fill in text in “readme.md”
- Stage “readme.md”



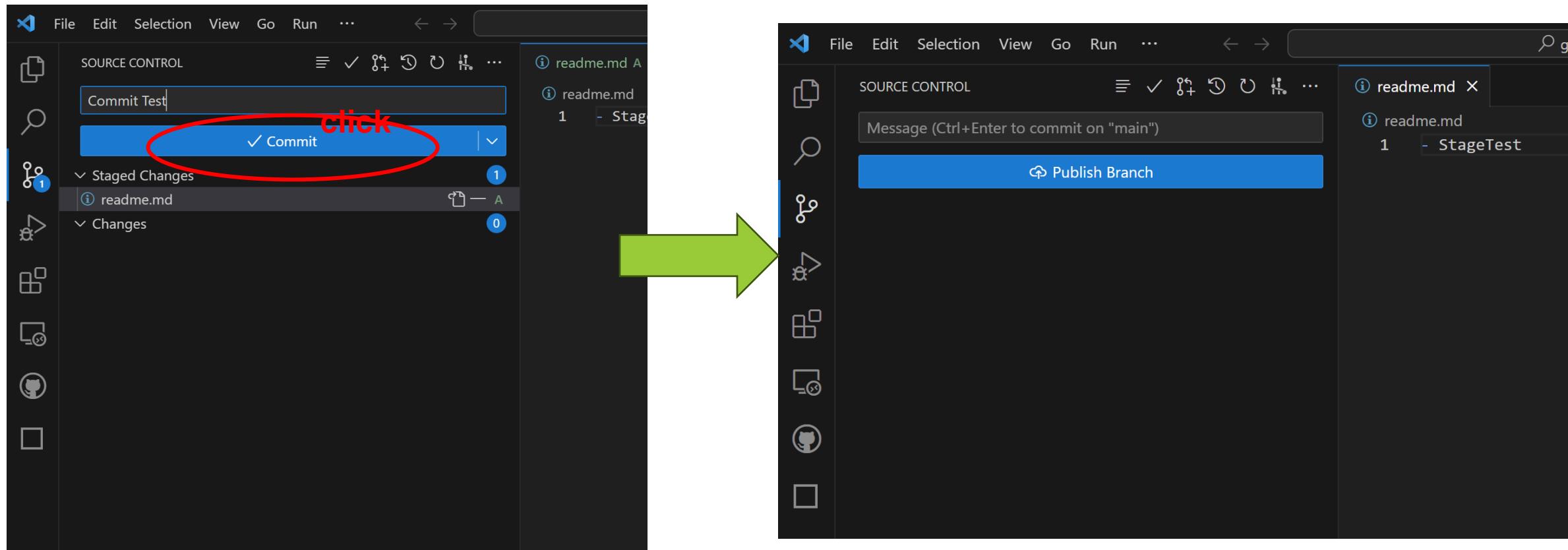
差分の確認 (Confirmation of difference)

- Click on “readme.md” in the stage area



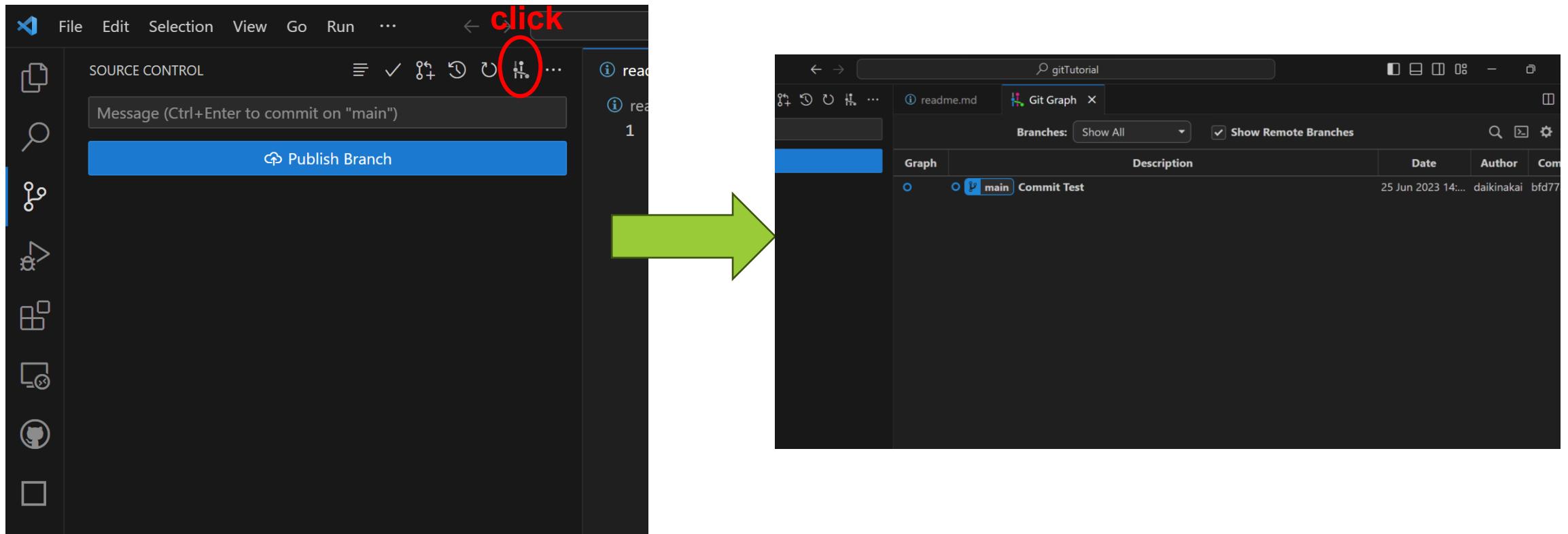
Commit

- コミットメッセージを記入 (Fill in the commit message)
- Click Commit

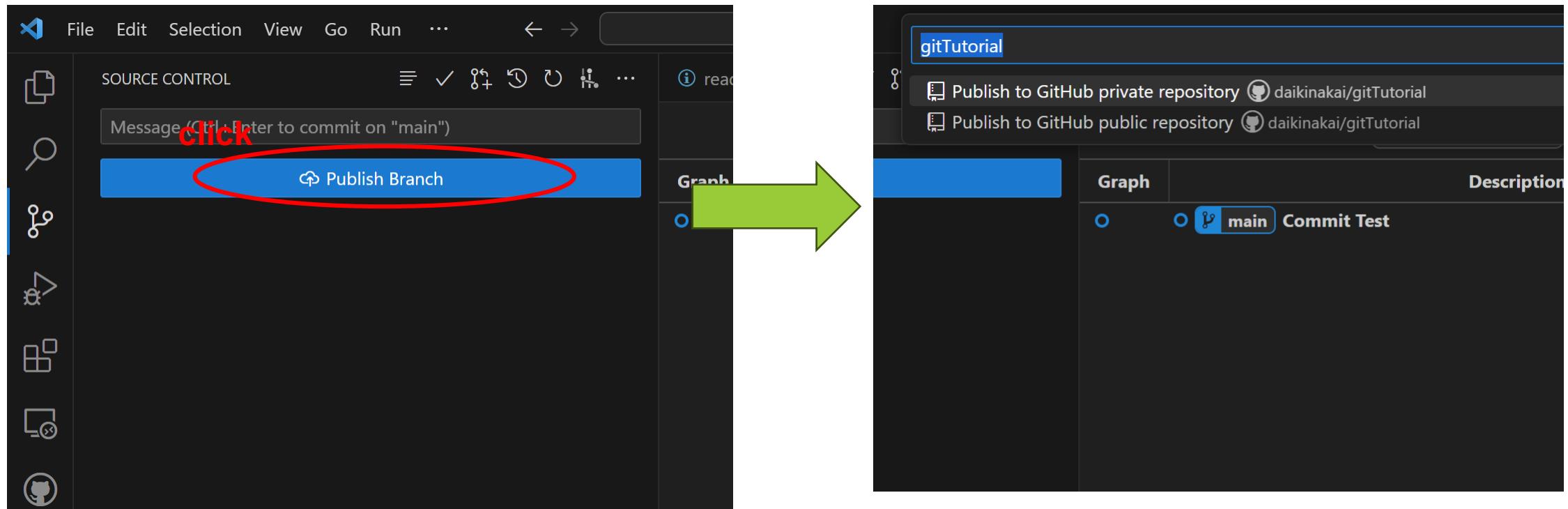


コミット歴の確認(Confirmation of Commit History)

- Open Git Graph

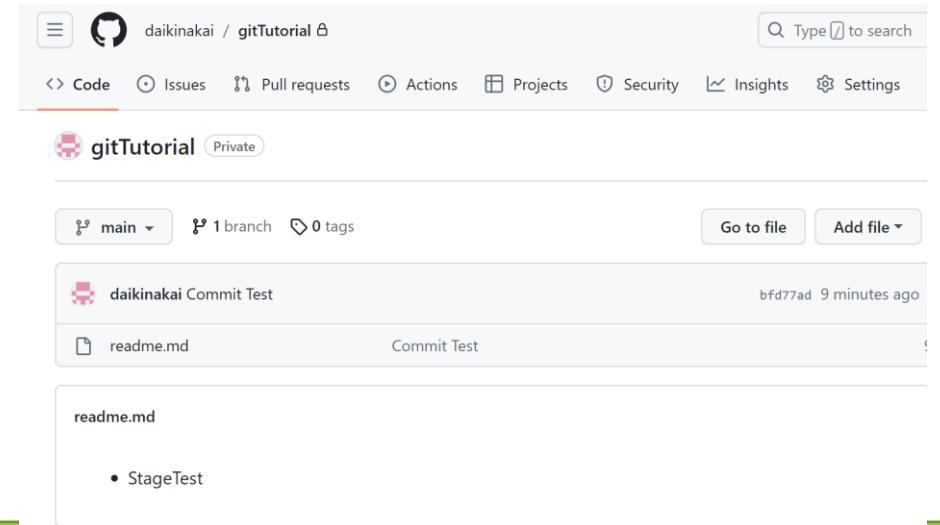


Publish Branch



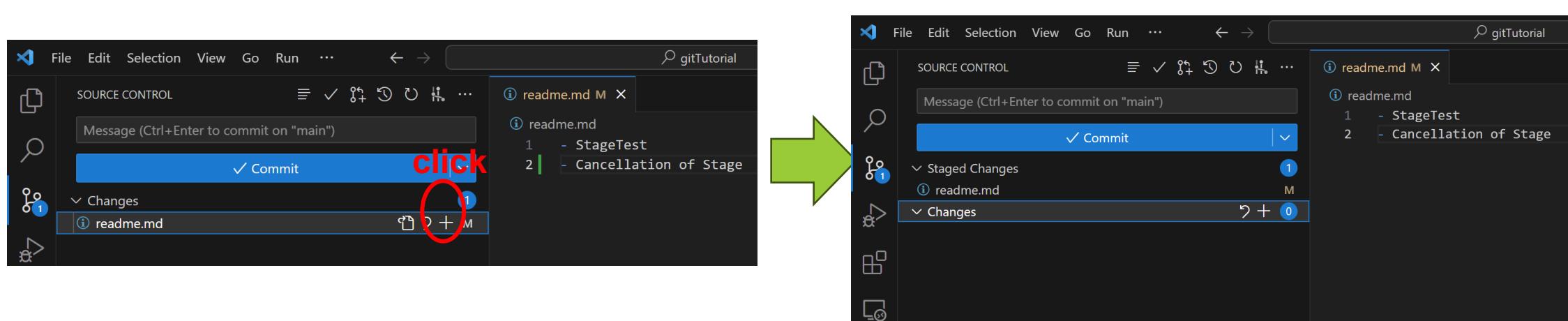
Confirmation

- 自身のgithubアカウントを開く
- Repository部分から「gitTutorial」を開き、確認
- Open your own github account
- Open "gitTutorial" from the Repository section and check it.



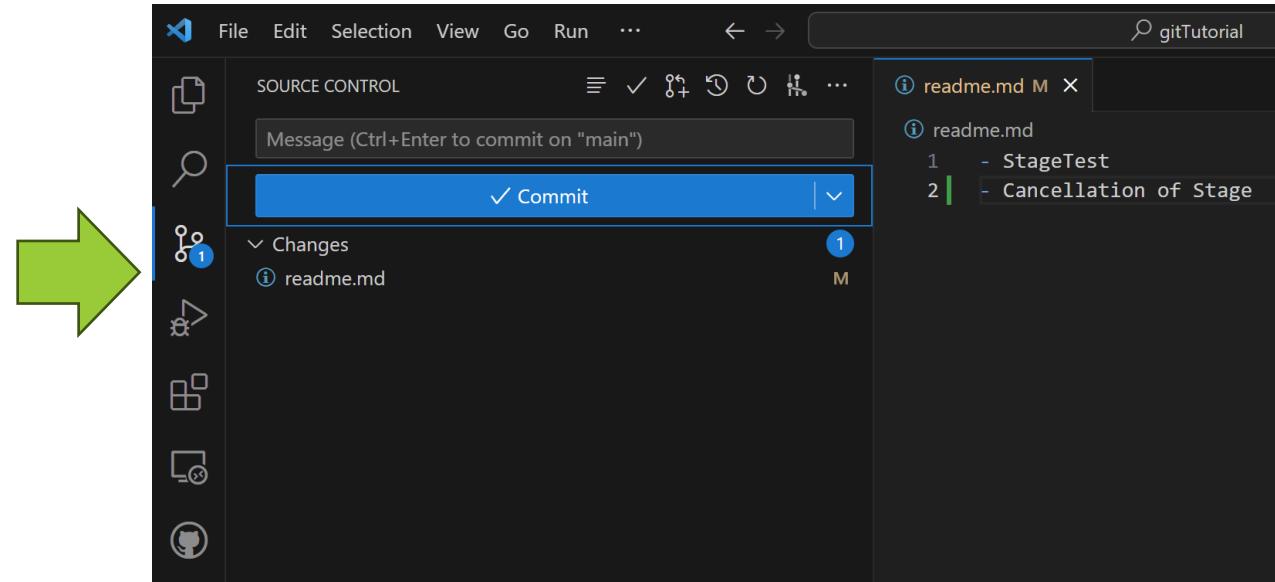
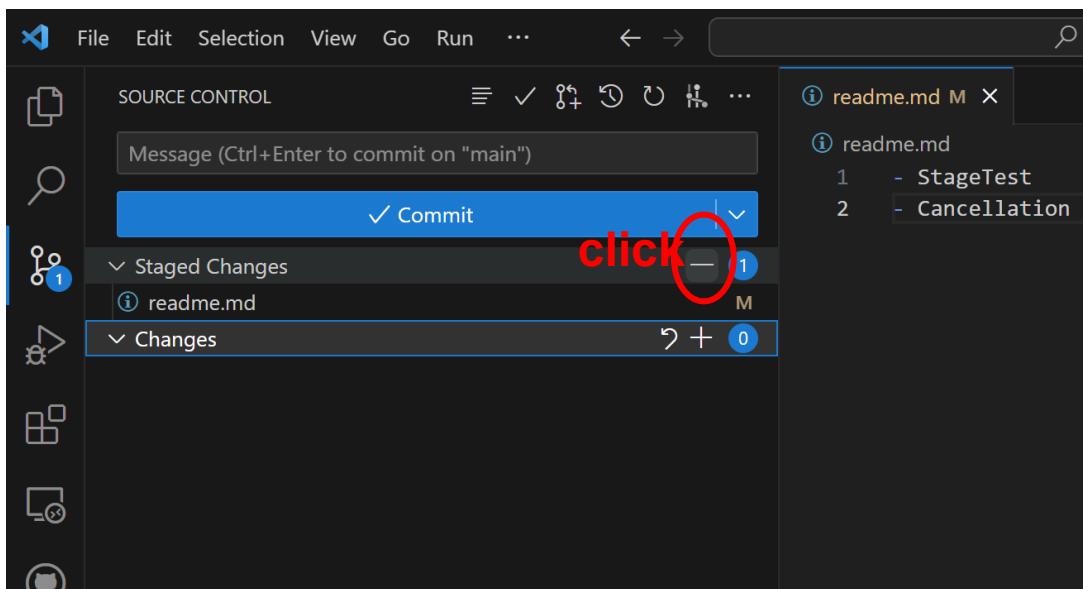
ステージの取り消し (Cancellation of Stage)

1. Fill in text in “readme.md”
2. 変更後、“readme.md”をステージエリアに移動
After the change, move "readme.md" to the stage area



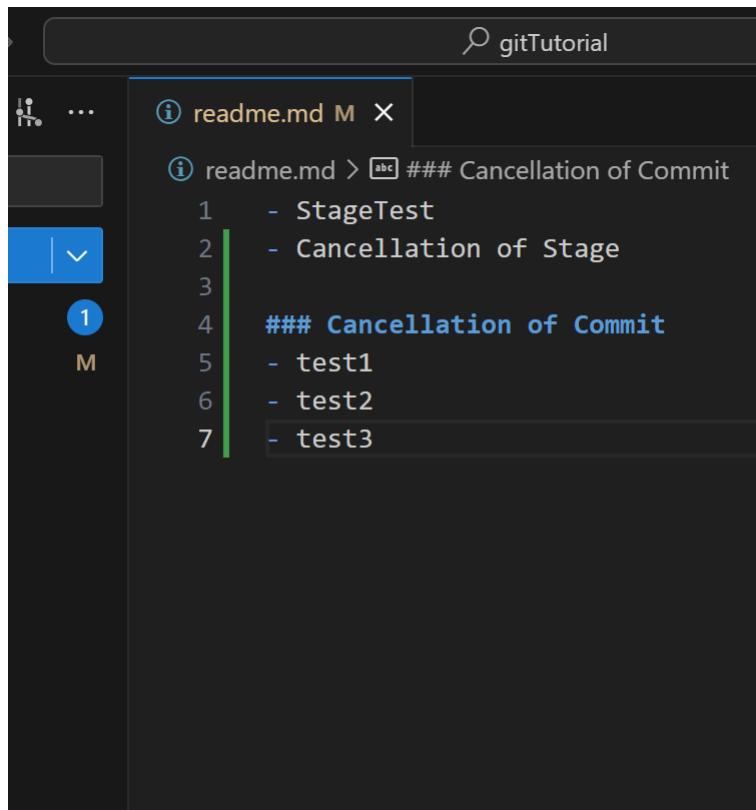
ステージの取り消し (Cancellation of Stage)

1. “readme.md”をステージエリアから削除
Remove "readme.md" from the stage area



コミットの取り消し (Cancellation of Commit)

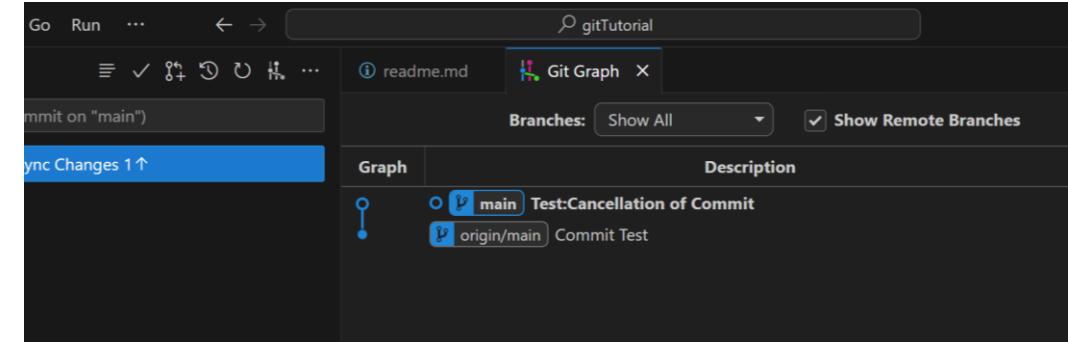
- Fill in text in “readme.md”



```
gitTutorial
readme.md M X
readme.md > abc #### Cancellation of Commit
1 - StageTest
2 - Cancellation of Stage
3
4 #### Cancellation of Commit
5 - test1
6 - test2
7 - test3
```

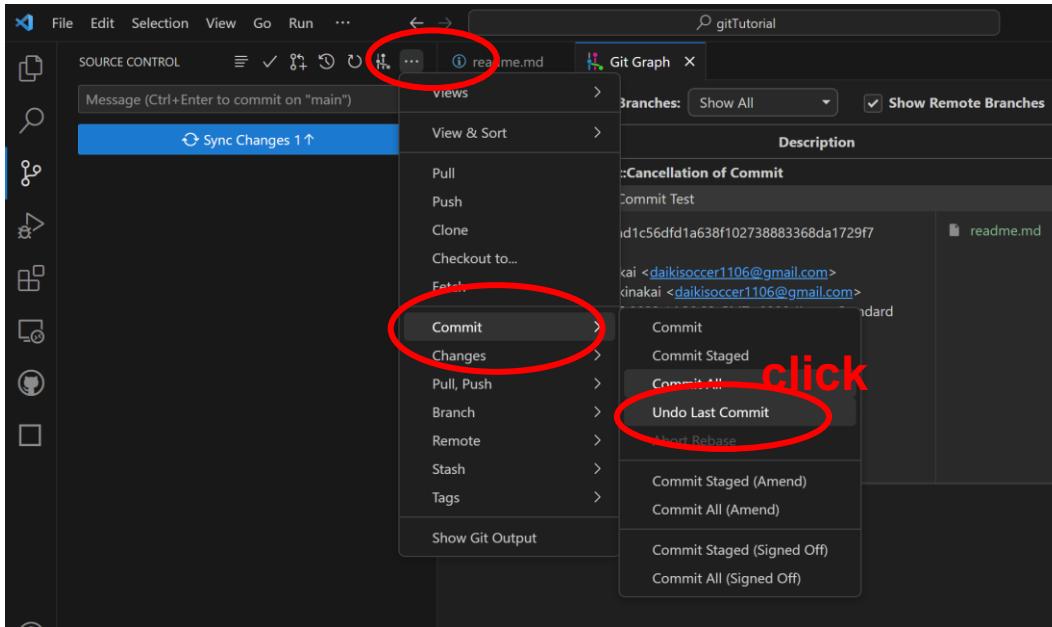


- Commit "readme.md"
 - “gitGraph”でcommitを確認
- Confirmation of commit contents with "gitGraph"**

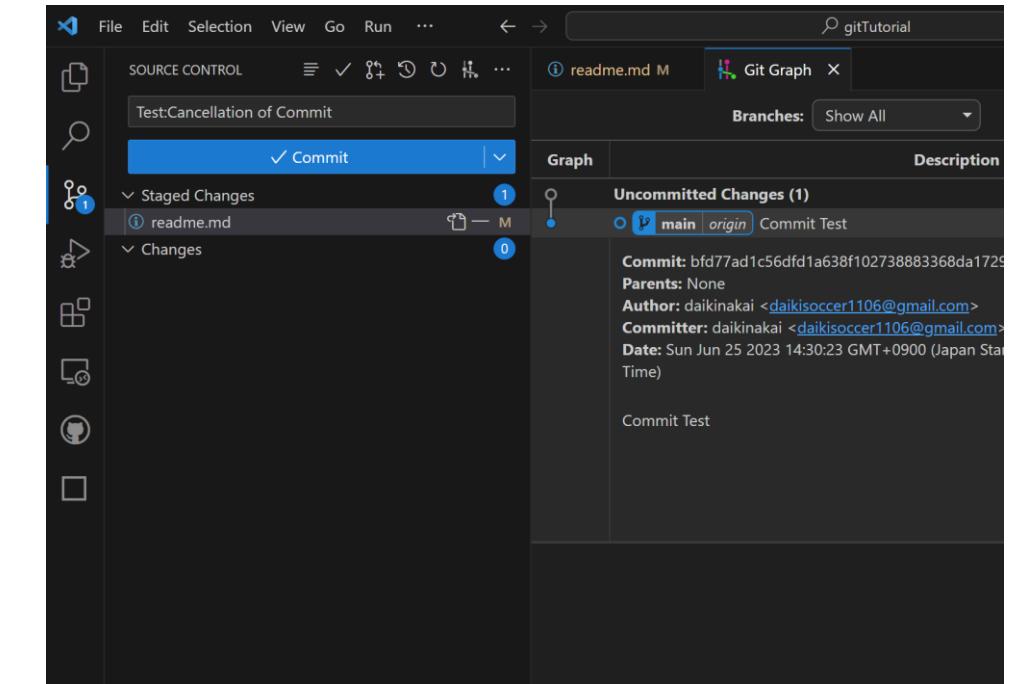


コミットの取り消し (Cancellation of Commit)

Click 「... → Commit → Undo Last Commit」



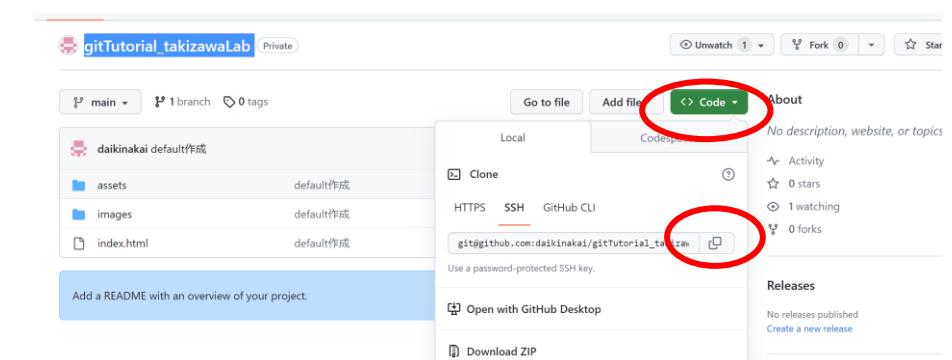
“gitGraph”でcommitを確認
Confirmation of commit contents with "gitGraph"



Tutorial git (cooperation)

Clone

1. vscodeで作業用のディレクトリに移動
2. 自分のgithubを開き、2nd git GitHub レポジトリを開く
3. CodeからSSH用のurlをコピーする。

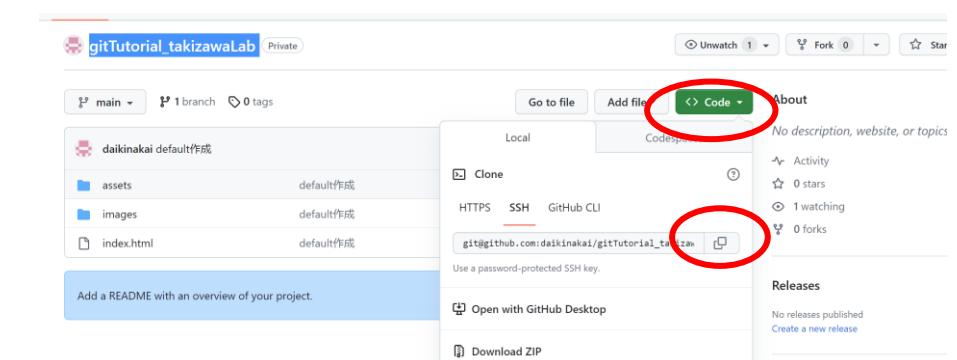


4. Vscodeのターミナルでgit clone “url”を入力

```
takizawaLab> git clone git@github.com:daikinakai/gitTutorial_takizawaLab.git
```

Clone

1. Move to a working directory with `vscodeOpen`
2. your github and open the `2nd_git_GitHub` repository
3. Copy the SSH url from Code.

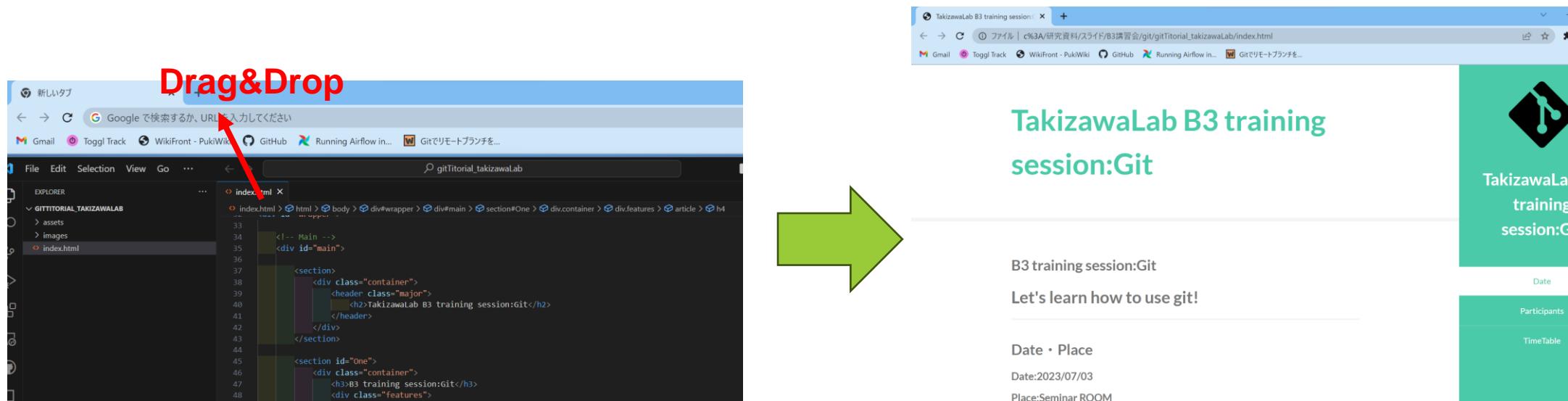


4. Enter `git clone "url"` in Vscode terminal

```
l_takizawaLab> git clone git@github.com:daikinakai/gitTutorial_takizawaLab.git
```

動作確認 (Confirmation of operation)

1. Vscodeでcloneした"gitTrutorial_takizawaLab"フォルダを開く
 2. "index.html"をブラウザにドラッグ & ドロップ
-
1. Open the "gitTrutorial_takizawaLab" folder clonedVscode
 2. Drag and drop "index.html" into the browser.



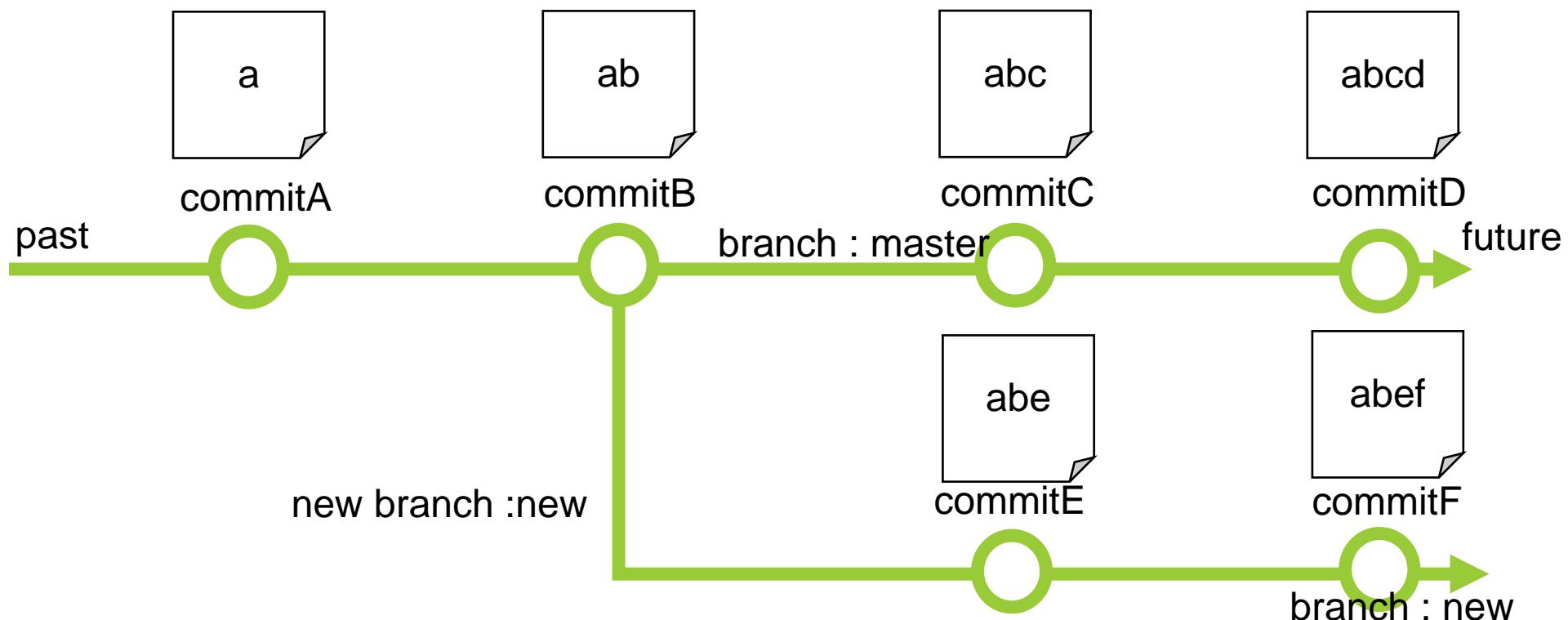
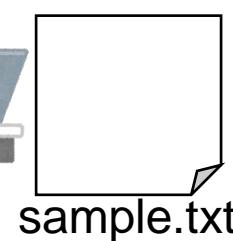
概念:branch

- 平行して最新バージョンを作る

開発中の最新バージョンとデバック済みリリース版の最新バージョンを明確に分けられたりする

You can clearly separate the latest version in development from the latest version in a debugged release.

You can make the latest version in parallel

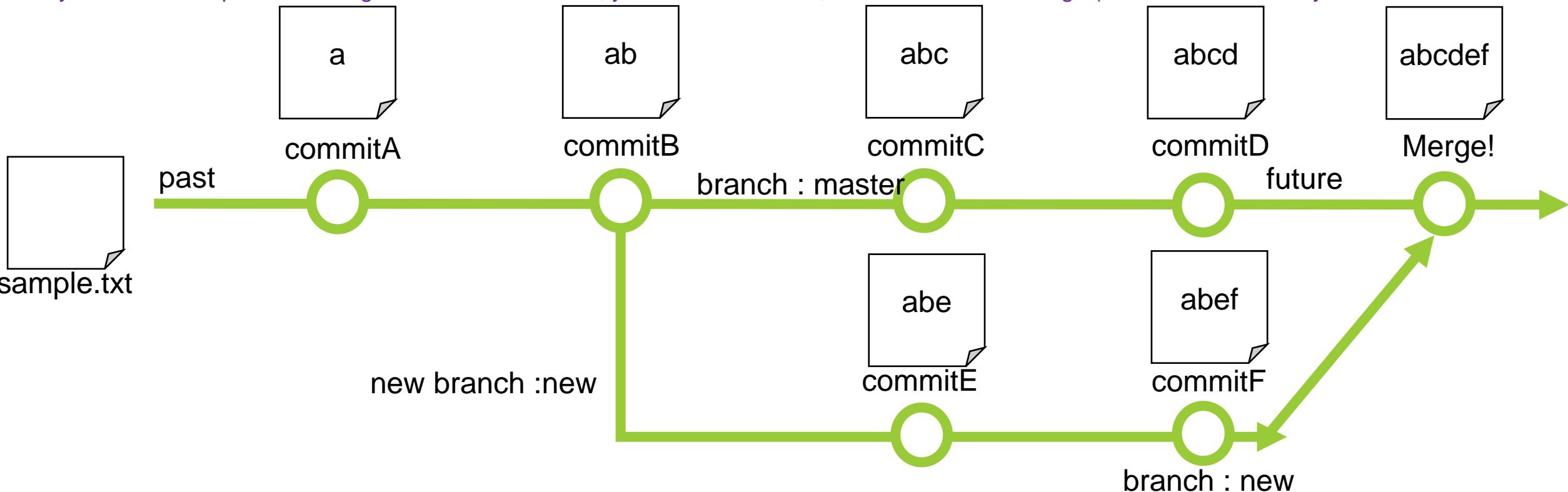


概念:merge

- 枝分かれしたブランチの変更をmasterブランチに反映 = merge
- 差分を確認しながら変更を反映
- newブランチのcommit Logはmasterブランチにはのこらない
- masterブランチにログを残したければmergeと似た操作にrebaseがある(慣れないうちは非推奨)

- Apply changes from new branch to master branch.
- You can reflect changes while viewing the differences.
- The commit log of the new branch is not saved in the master branch.

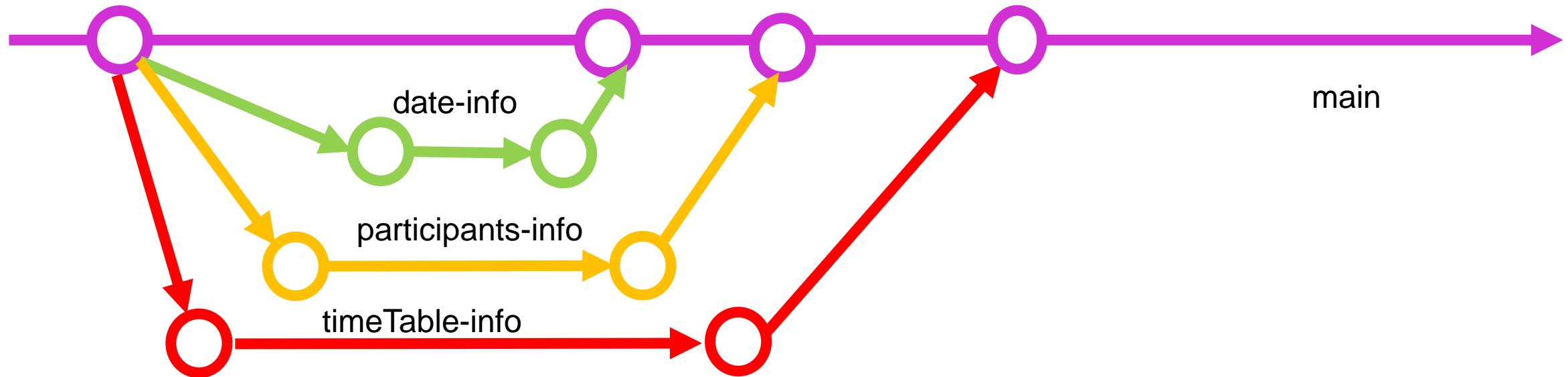
If you want to keep a commit log on the master branch, you can use **rebase**, which is similar to merge (not recommended if you are not familiar with it).



Goal

- 3人で共同で開発を行い、webページを完成させよう
- branchを使った操作になれよう
- Three people work together to develop and complete a web page!
- Let's get used to using the branch!

Branch Image

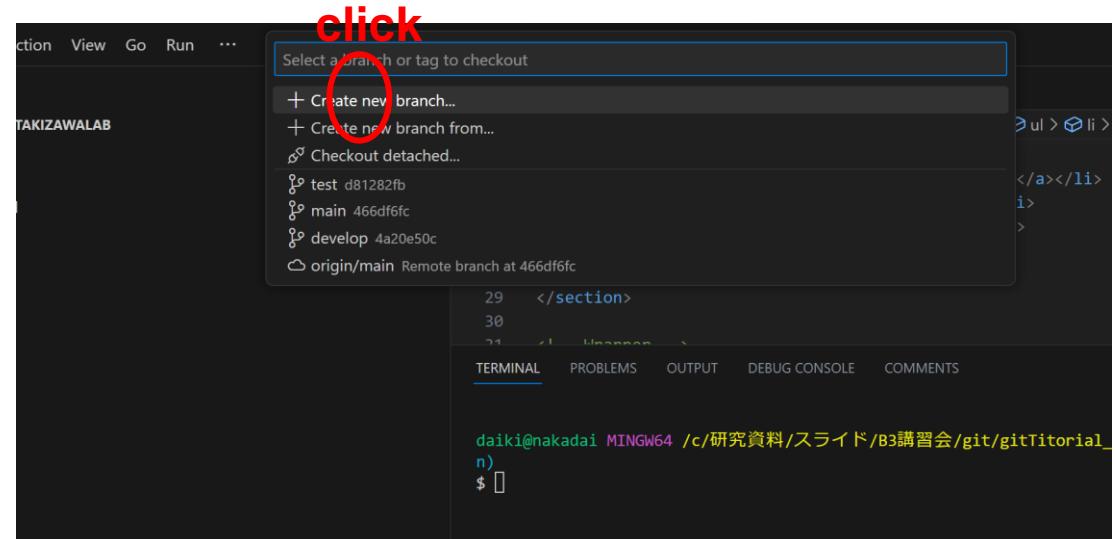
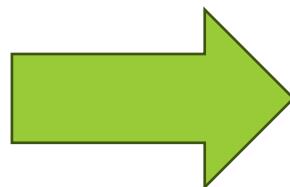
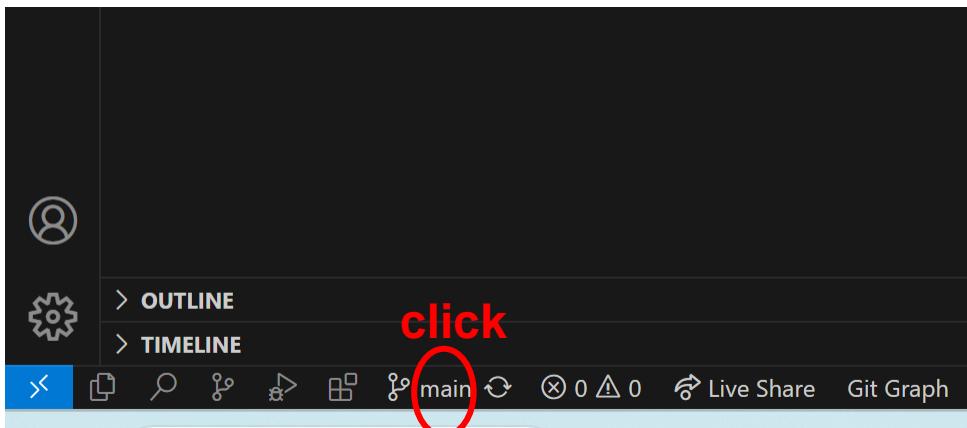


役割 (Role)

- **Date:Watanabe** → p46
- **Participants:ishihara** → p56
- **TimeTable:Pakawat** → p66

date-info

ブランチの作成



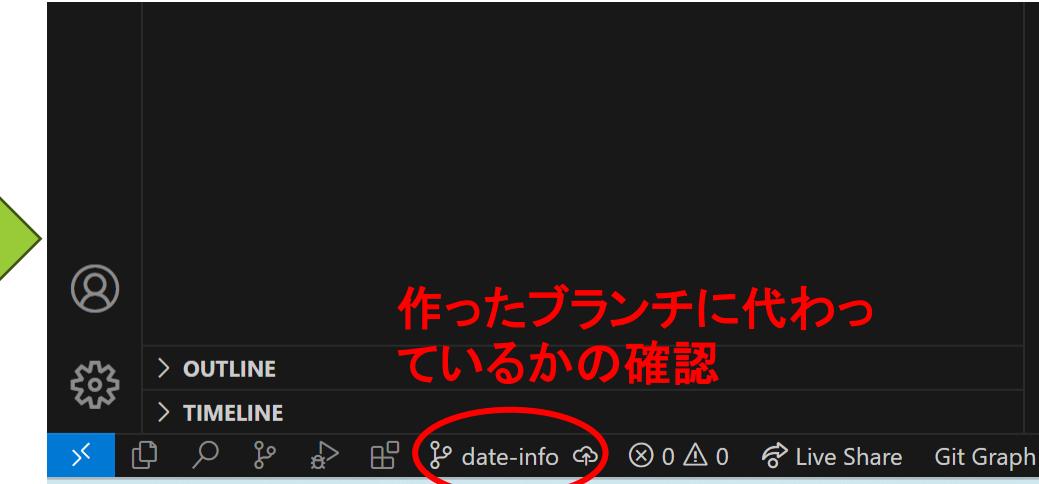
ブランチの作成

date-info ブランチの作成

The screenshot shows the VS Code interface. In the top right, a modal window prompts the user to "Please provide a new branch name (Press 'Enter' to confirm or 'Escape' to cancel)". Below the modal, the code editor displays the file index.html with the following code:

```
<ul>
  <li><a href="#one" class="active">Date</a></li>
  <li><a href="#two">Participants</a></li>
  <li><a href="#three">TimeTable</a></li>
</ul>
```

The modal has the title "date-info". The status bar at the bottom shows the path "daiki@nakada1 MINGW64 /c/研究資料/スライド/B3講習会/git/gitTutorial_takizawaLab (n)" and a terminal prompt "\$".



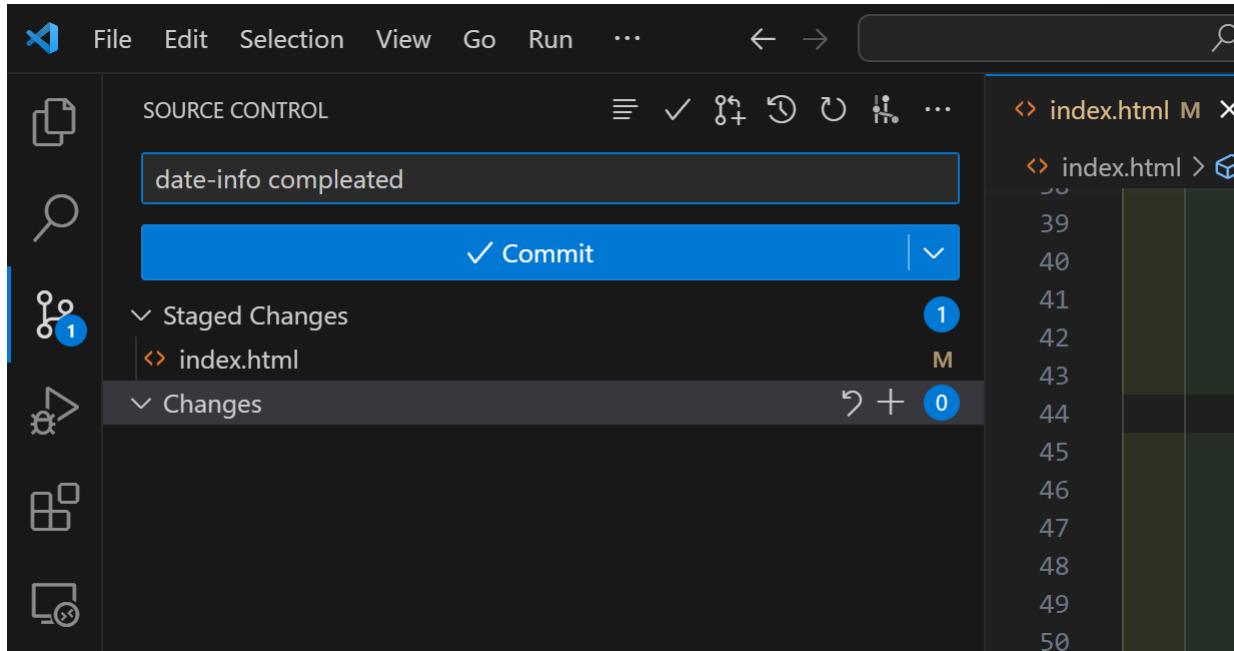
ファイル内容を変更

- “index.html”を開き、section id=“One” のファイルを以下のように変更

```
1 42      </div>
2 43    </section>
3 44
4 45  <section id="One">
5 46    <div class="container">
6 47      <h3>B3 training session:Git</h3>
7 48      <div class="features">
8 49        <h2>Let's learn how to use git!</h2>
9 50        <article>
10 51          <h3>Date・Place</h3>
11 52          <h4>Date:2023/07/03</h4>
12 53          <h4>Place:Seminar ROOM</h4>
13 54        </article>
14 55      </div>
15 56    </div>
16 57  </section>
17 58  <section id="Two">
```

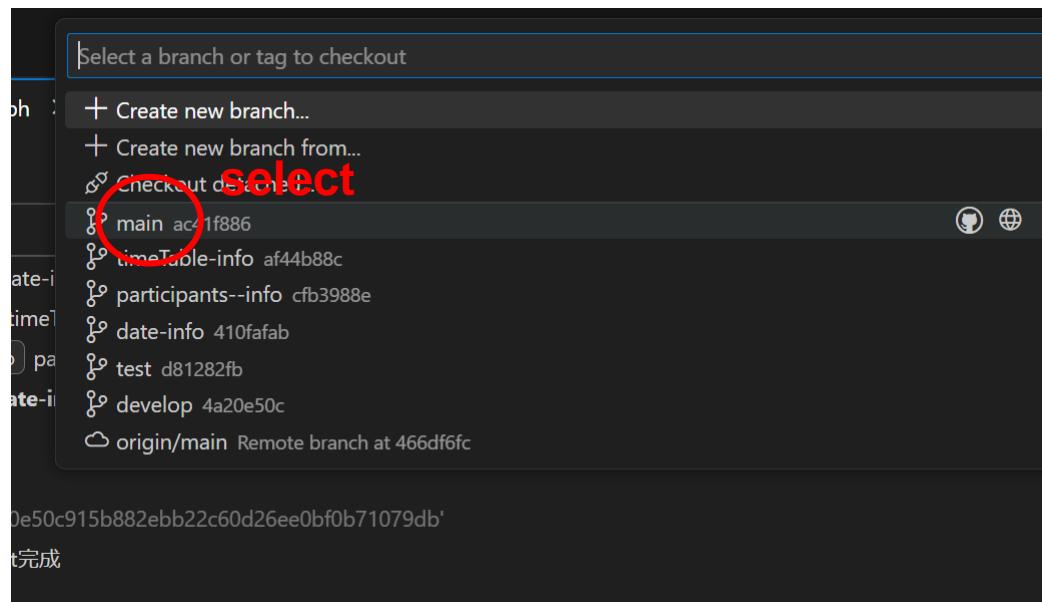
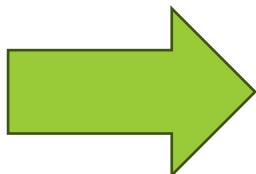
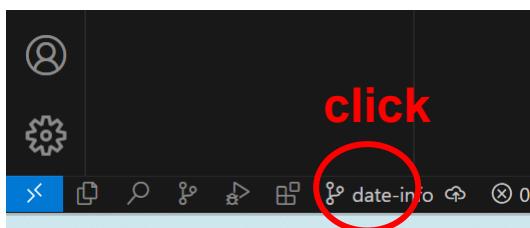
コミットする

- 変更をコミットする

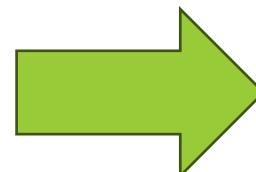
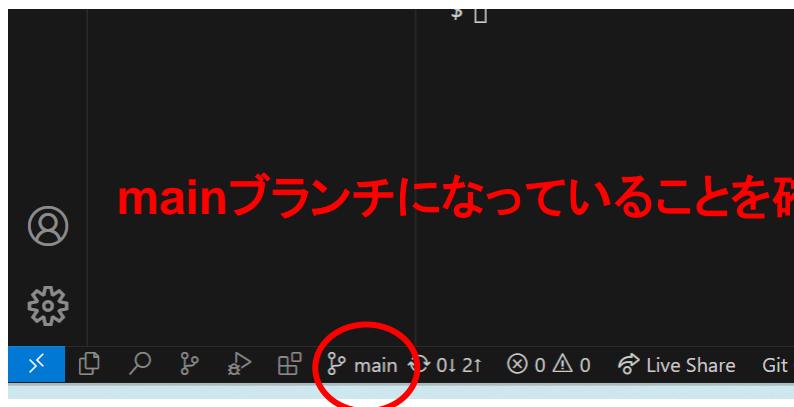


mainブランチにmerge

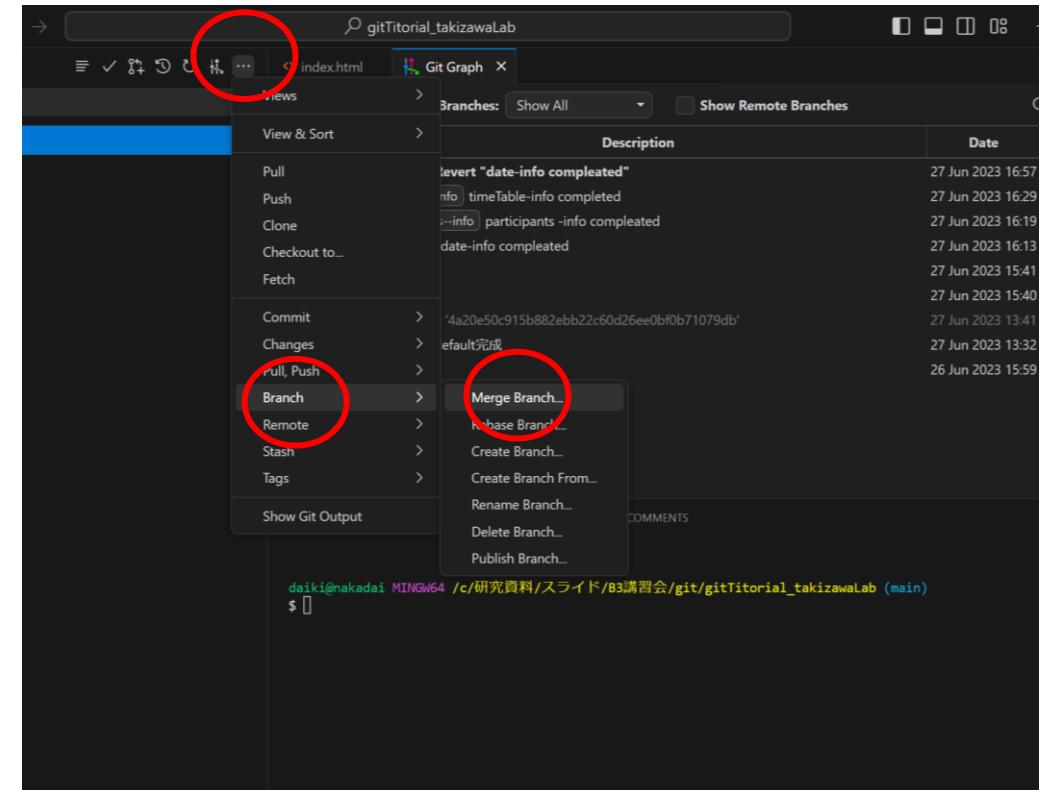
- mainブランチに移動



mainブランチにmerge

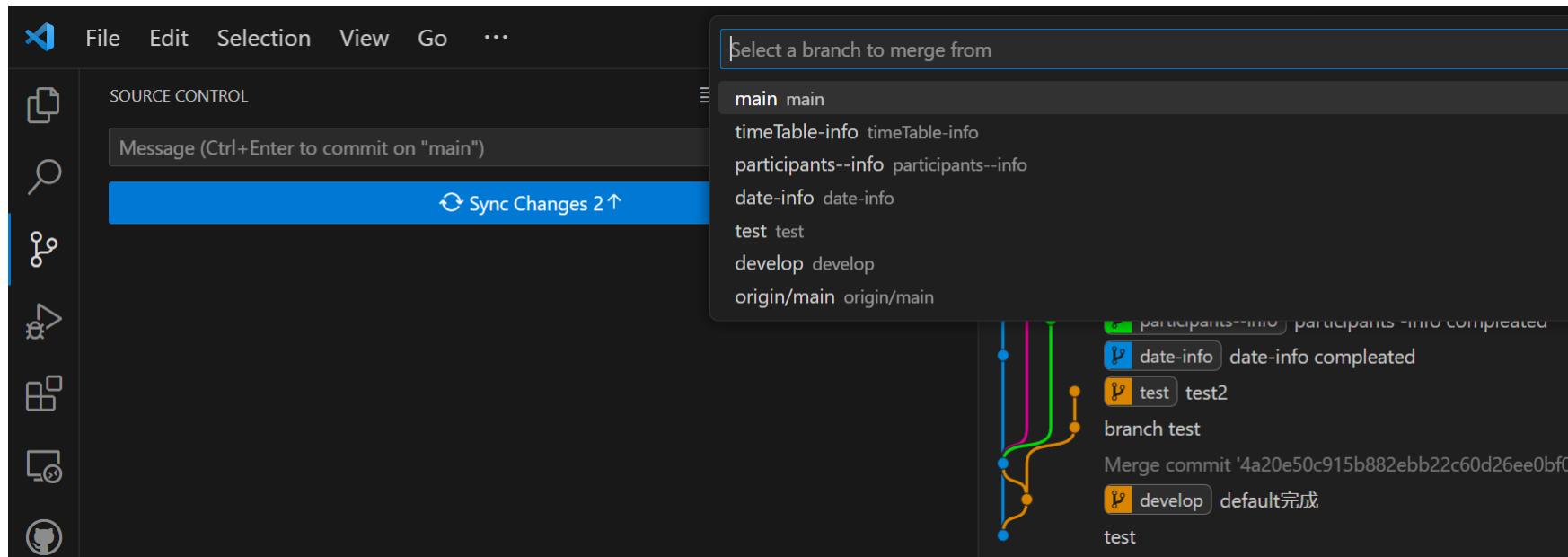


Merge Branch を選択



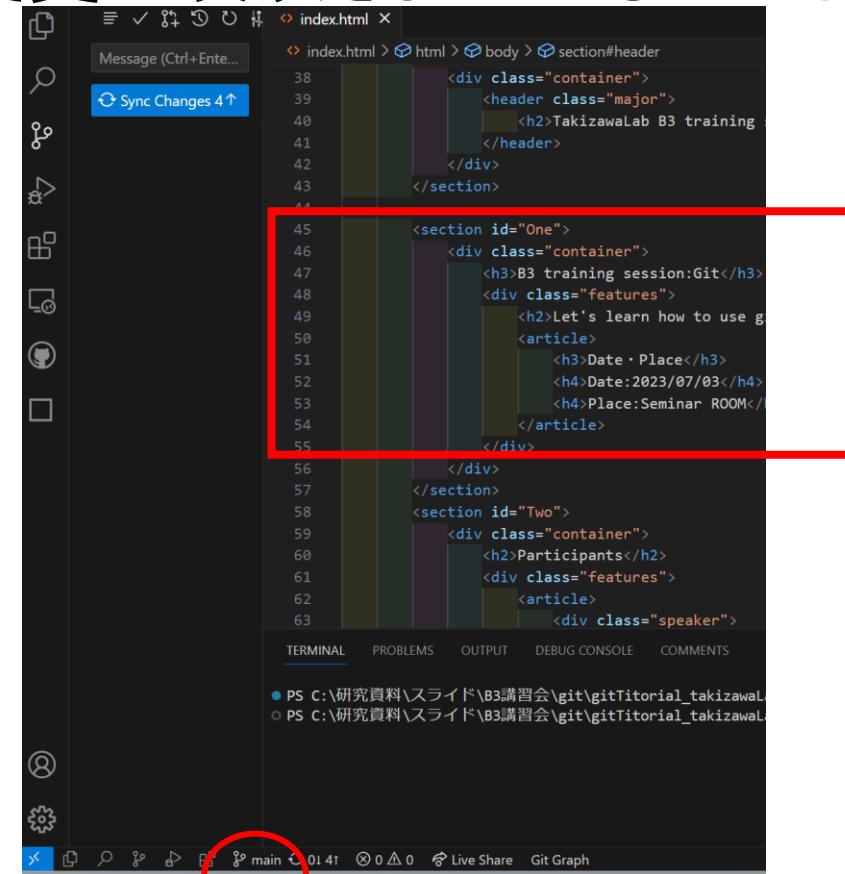
mainブランチにmerge

- Mergeさせたいブランチを選択
- “date-info”を選択



確認

- mainブランチの”index.html”に変更が反映されているかの確認



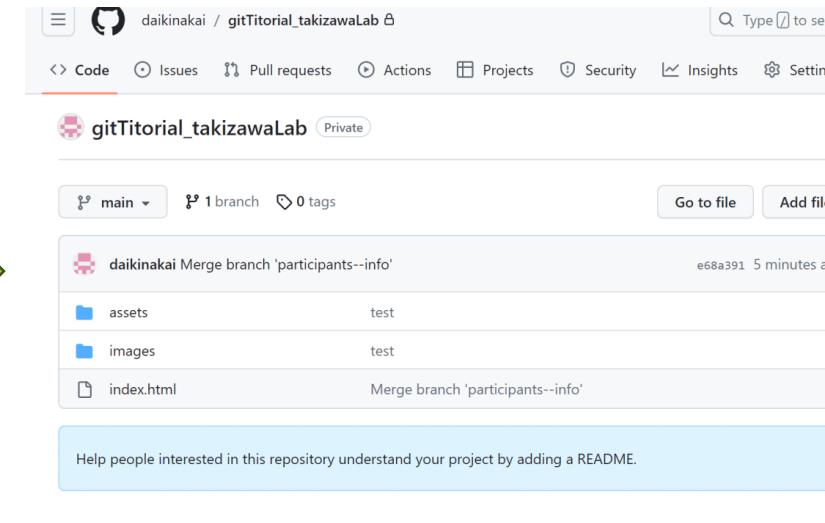
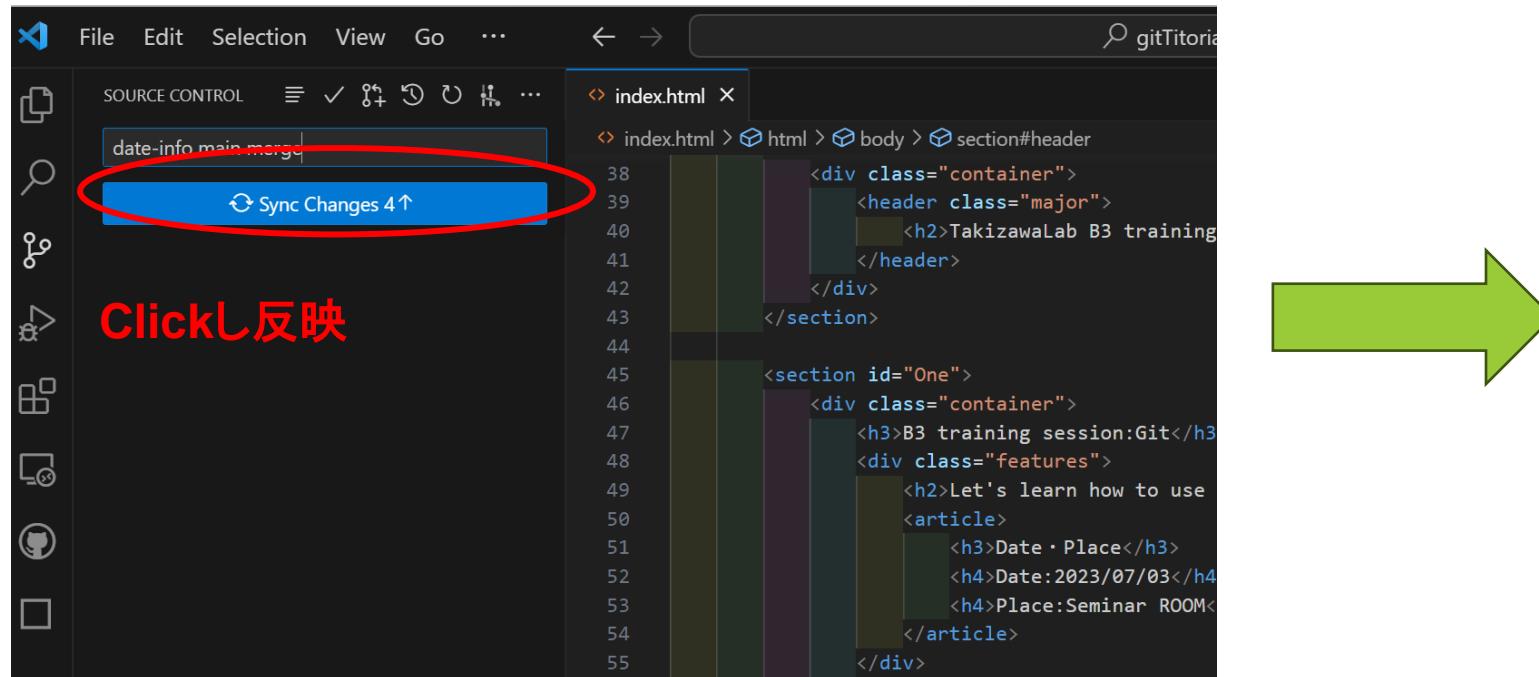
The screenshot shows a code editor window for the file "index.html". The code is displayed in a dark-themed editor. A red box highlights a section of the code starting from line 45, which contains the content for the "One" section. Below this, another red circle highlights the status bar at the bottom of the editor, which shows the branch name "main".

```
<div class="container">
  <header class="major">
    <h2>TakizawaLab B3 training</h2>
  </header>
</div>
</section>

<section id="One">
  <div class="container">
    <h3>B3 training session:Git</h3>
    <div class="features">
      <h2>Let's learn how to use git</h2>
      <article>
        <h3>Date・Place</h3>
        <h4>Date:2023/07/03</h4>
        <h4>Place:Seminar ROOM</h4>
      </article>
    </div>
  </div>
</section>
<section id="Two">
  <div class="container">
    <h2>Participants</h2>
    <div class="features">
      <article>
        <div class="speaker"></div>
      </article>
    </div>
  </div>
</section>
```

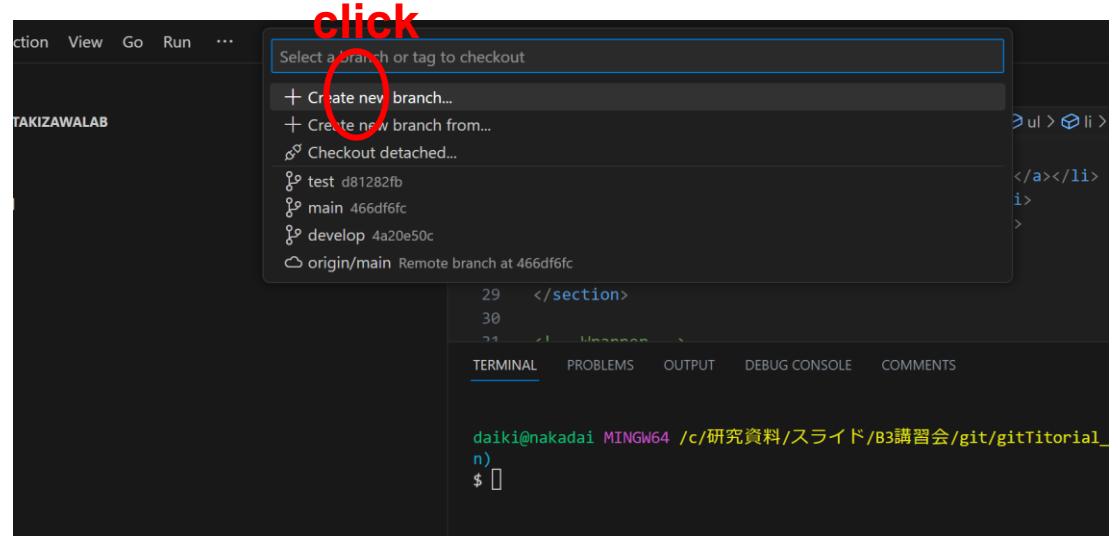
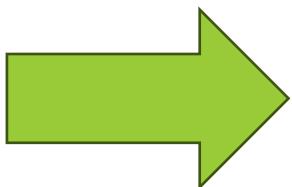
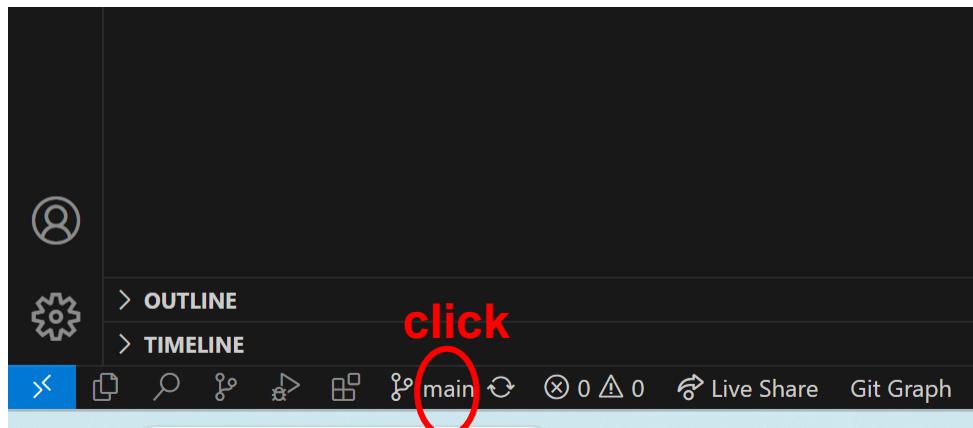
Push:Remoteに反映させる

Githubを開き確認



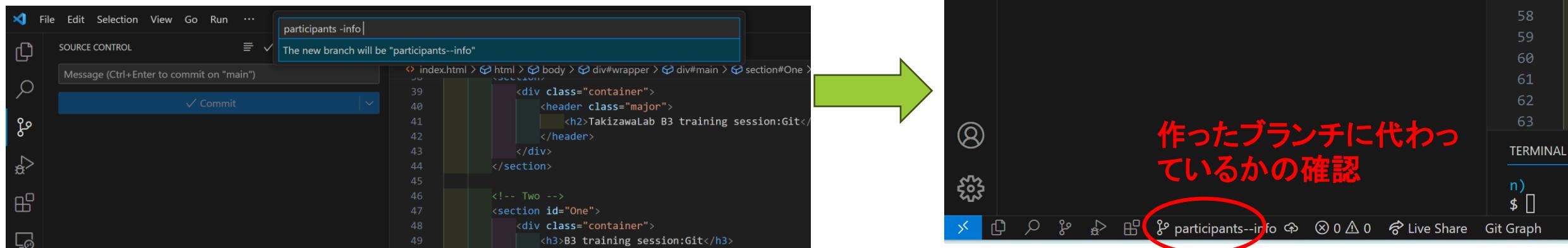
Participants -info

ブランチの作成



ブランチの作成

participants -info ブランチの作成



The new branch will be "participants--info"

Message (Ctrl+Enter to commit on "main")

✓ Commit

```
index.html
39
40
41
42
43
44
45
46
47
48
49
```

作ったブランチに代わっているかの確認

participants--info

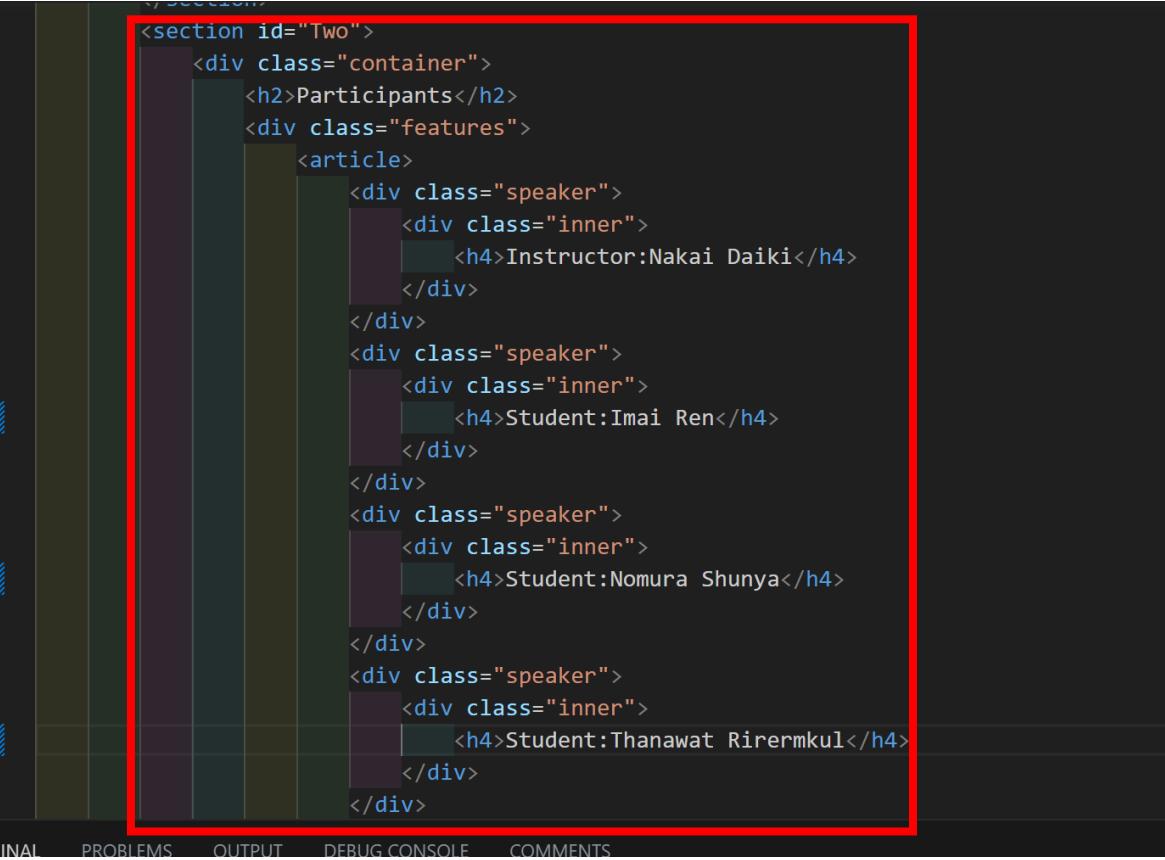
TERMINAL

58
59
60
61
62
63

58

ファイル内容を変更

- “index.html”を開き、section id=“Two” のファイルを以下のように変更

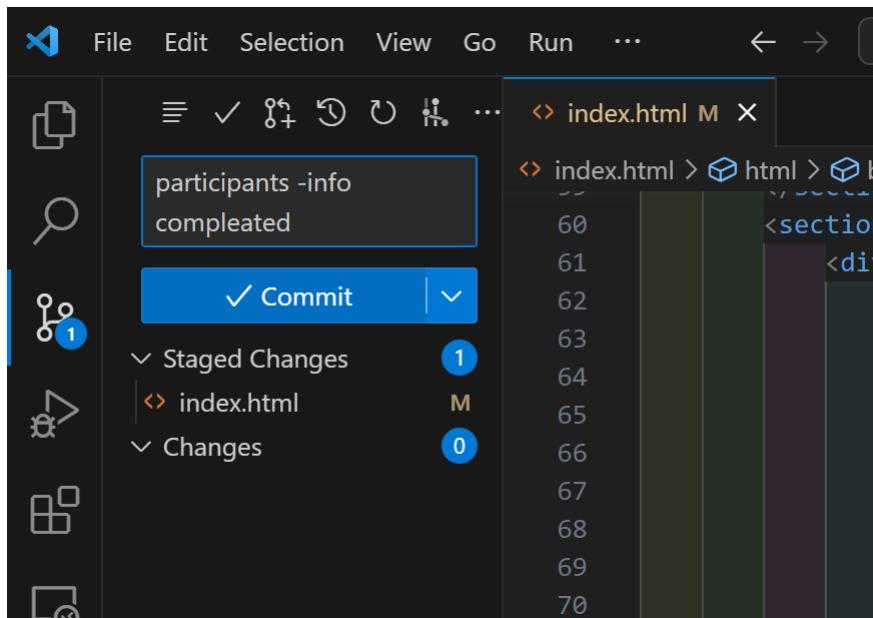


```
<section id="Two">
  <div class="container">
    <h2>Participants</h2>
    <div class="features">
      <article>
        <div class="speaker">
          <div class="inner">
            <h4>Instructor:Nakai Daiki</h4>
          </div>
        </div>
        <div class="speaker">
          <div class="inner">
            <h4>Student:Imai Ren</h4>
          </div>
        </div>
        <div class="speaker">
          <div class="inner">
            <h4>Student:Nomura Shunya</h4>
          </div>
        </div>
        <div class="speaker">
          <div class="inner">
            <h4>Student:Thanawat Rirermkul</h4>
          </div>
        </div>
      </article>
    </div>
  </div>
</section>
```

FINAL PROBLEMS OUTPUT DEBUG CONSOLE COMMENTS

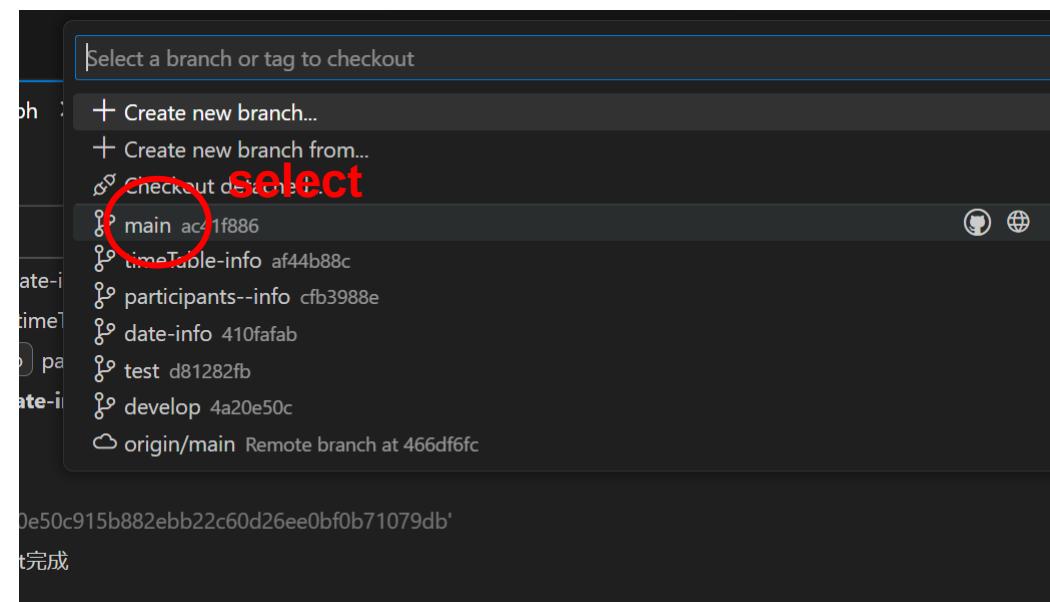
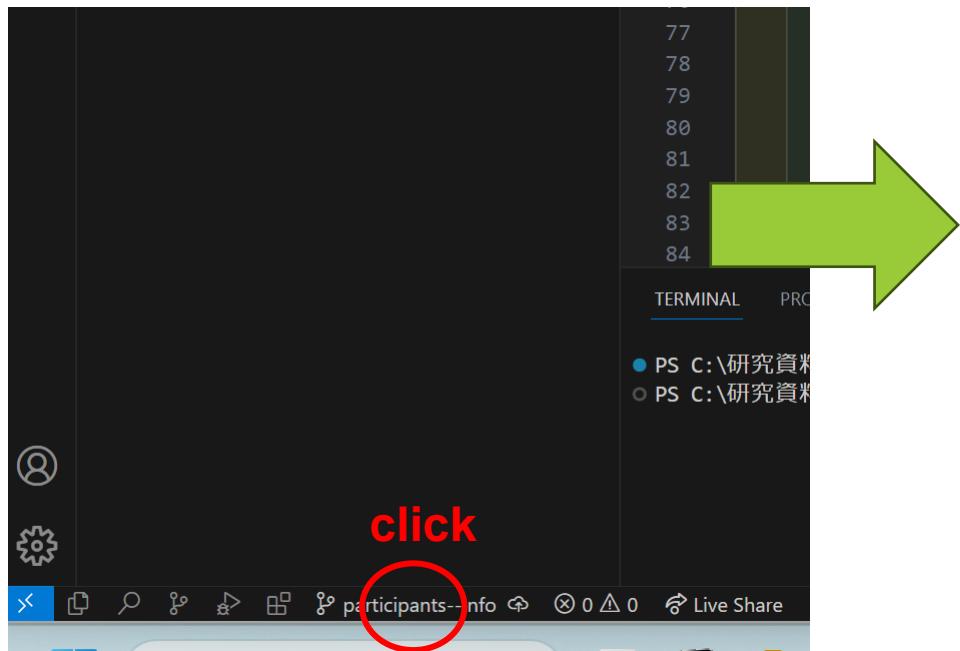
コミットする

- 変更をコミットする

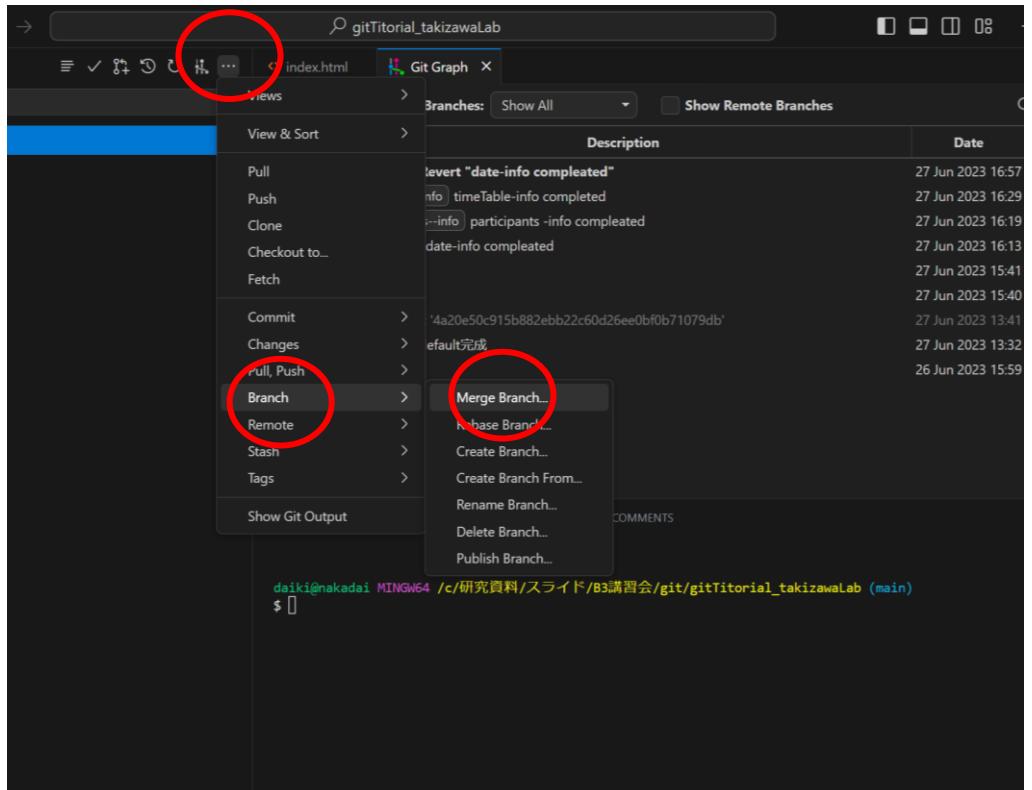
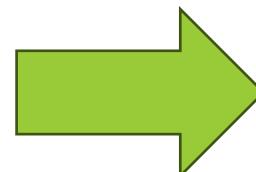
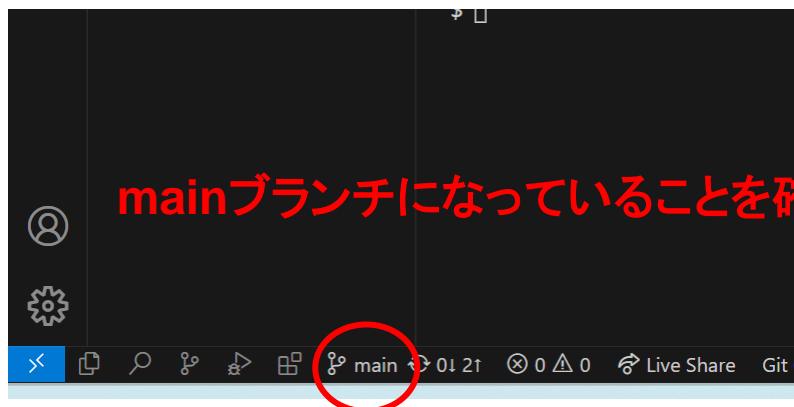


mainブランチにmerge

- mainブランチに移動

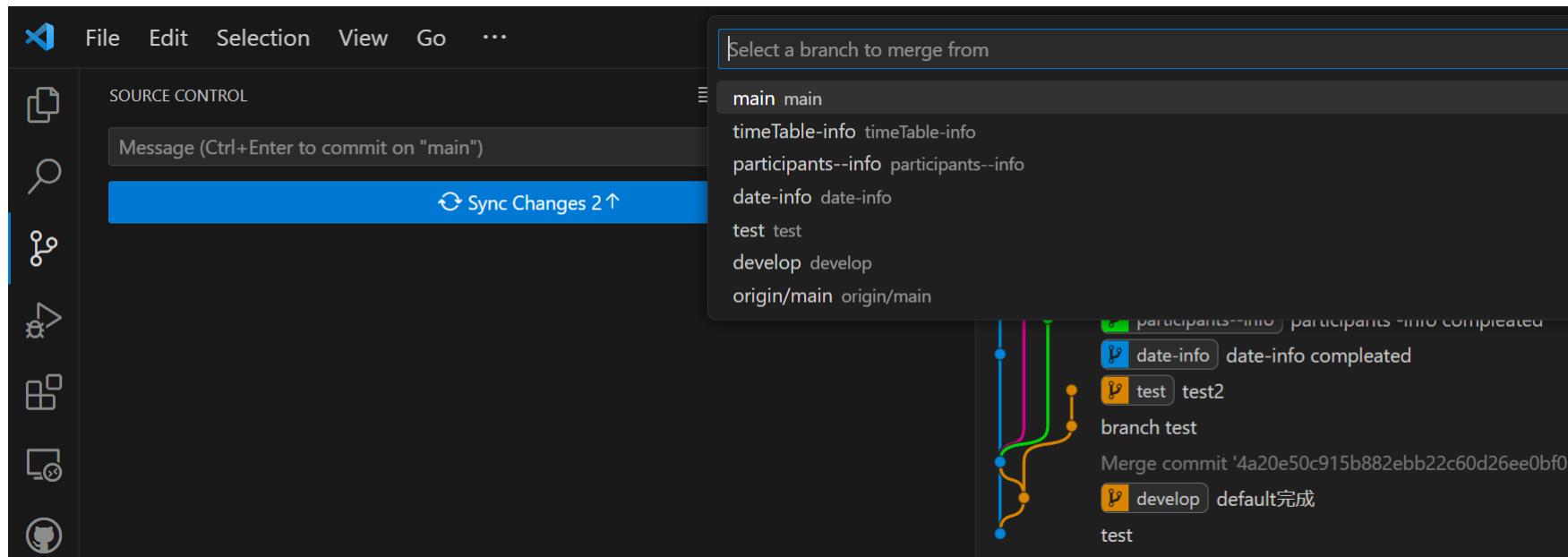


mainブランチにmerge



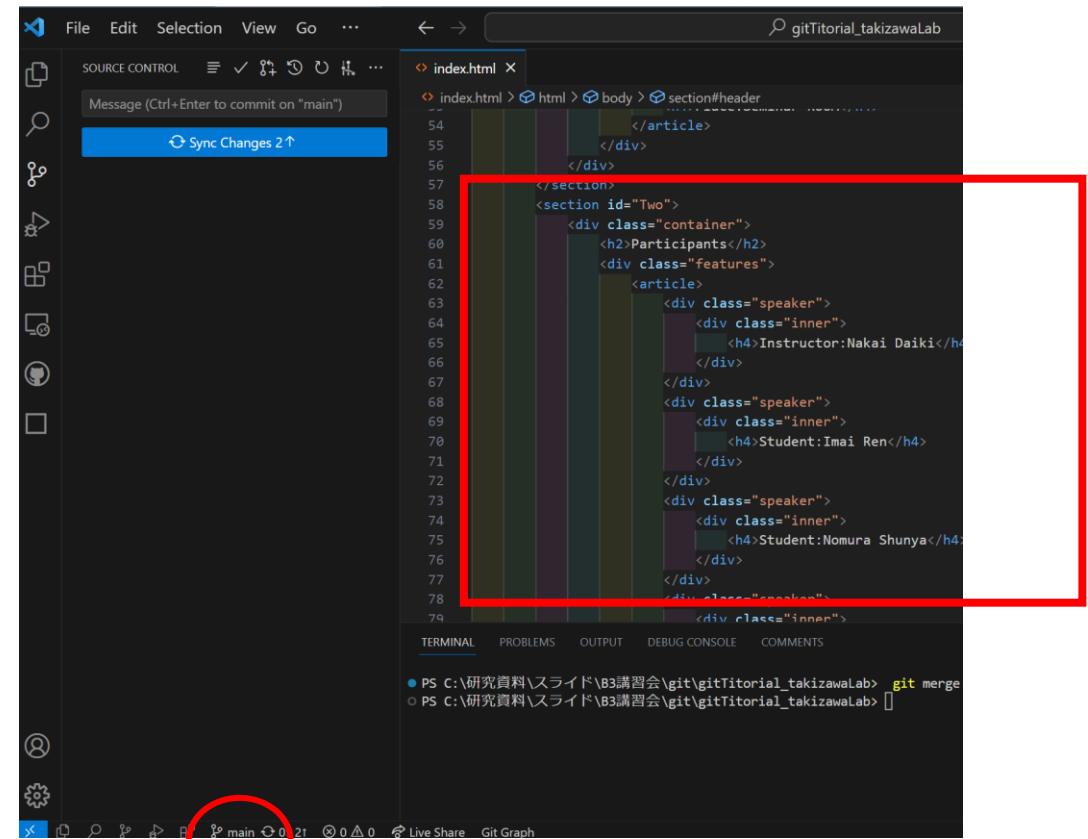
mainブランチにmerge

- Mergeさせたいブランチを選択
- “participants-info”を選択



確認

- mainブランチの”index.html”に変更が反映されているかの確認

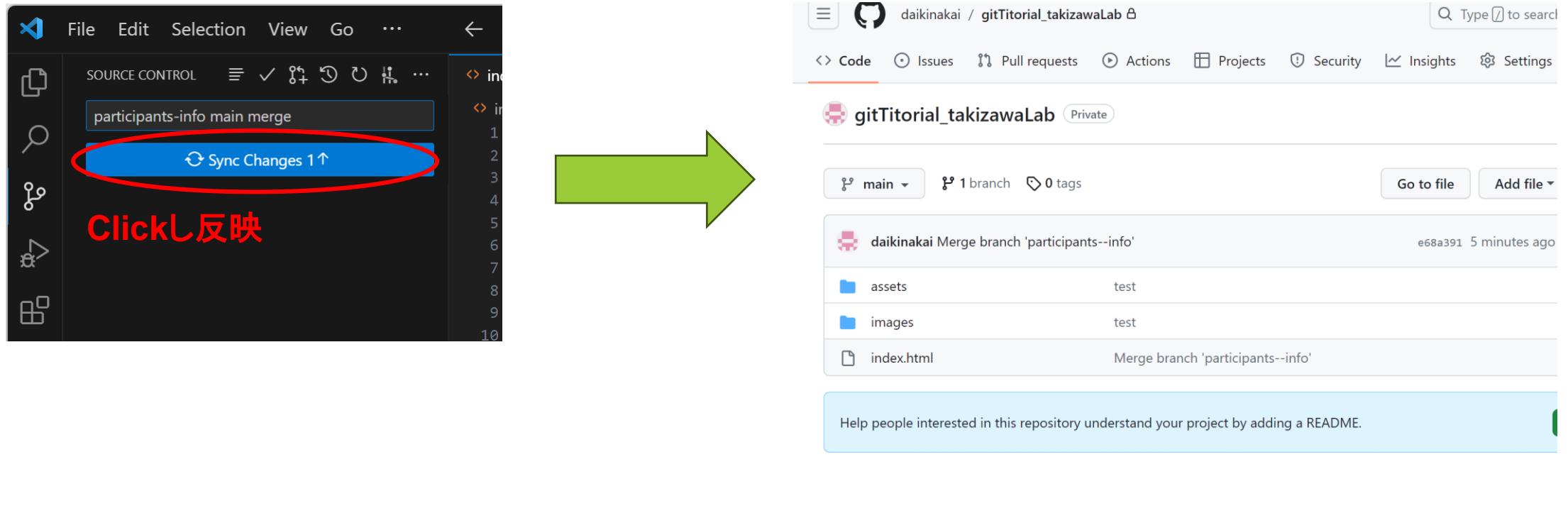


The screenshot shows the VS Code interface with the file "index.html" open. A red box highlights a section of the code where three speaker profiles are listed under the "features" section:

```
<div class="features">
  <article>
    <div class="speaker">
      <div class="inner">
        <h4>Instructor:Nakai Daiki</h4>
      </div>
    </div>
    <div class="speaker">
      <div class="inner">
        <h4>Student:Imai Ren</h4>
      </div>
    </div>
    <div class="speaker">
      <div class="inner">
        <h4>Student:Nomura Shunya</h4>
      </div>
    </div>
  </article>
</div>
```

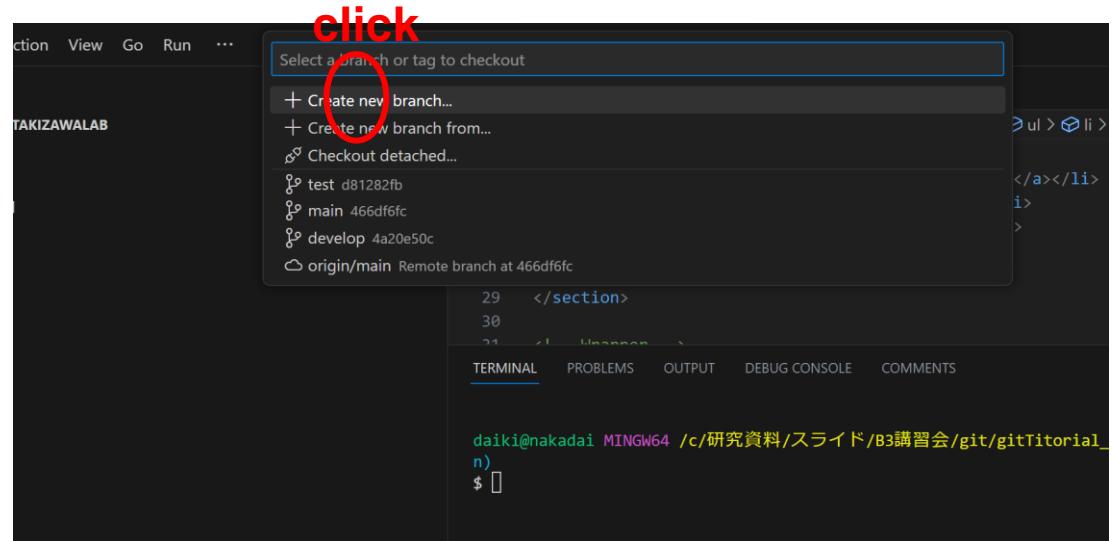
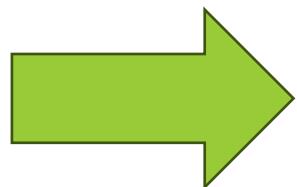
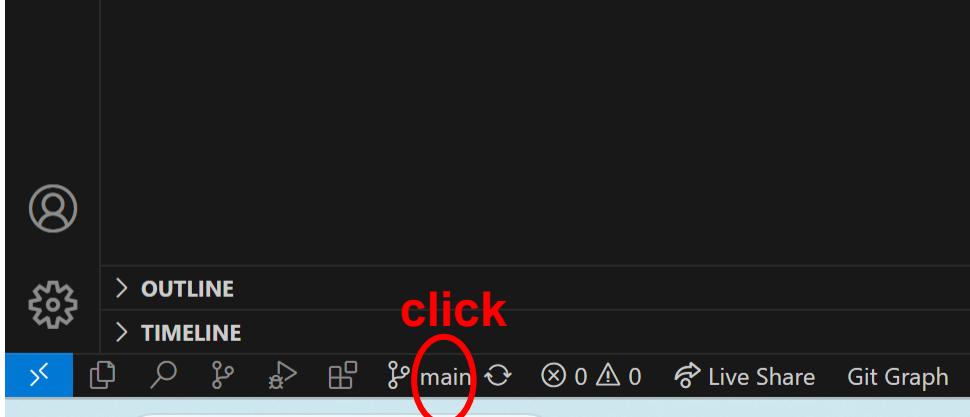
A red circle is drawn around the status bar at the bottom, which displays the branch name "main".

Push: Remoteに反映させる Githubを開き確認



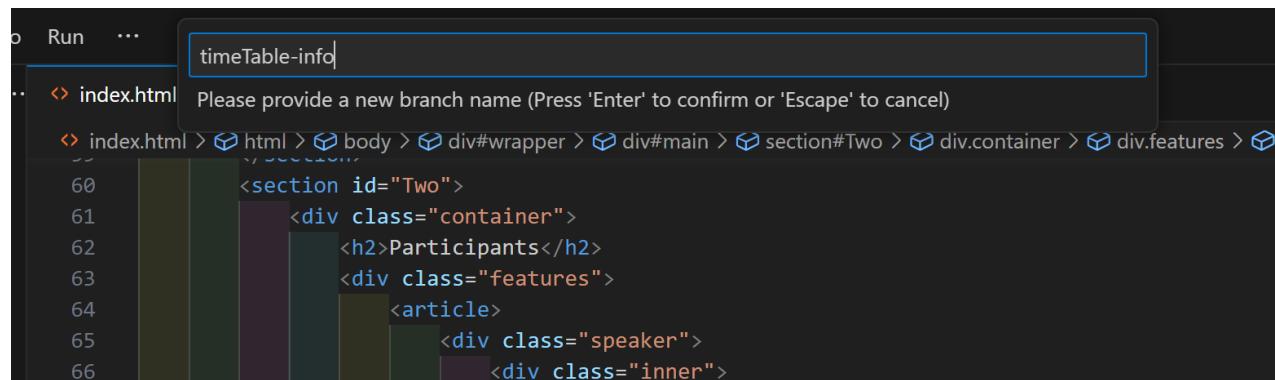
TimeTable-info

Make Branch



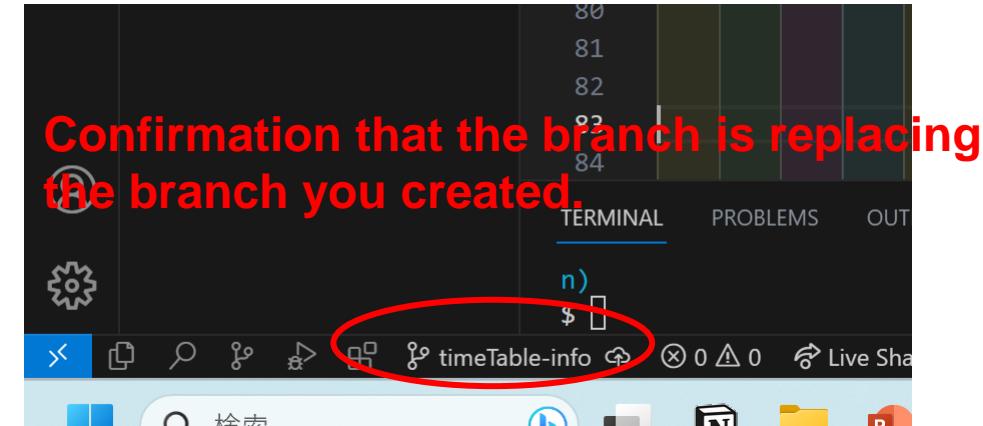
Make Branch

make “timeTable-info” branch



A screenshot of a code editor interface. In the top right corner, there is a search bar containing the text "timeTable-info". Below the search bar, a message says "Please provide a new branch name (Press 'Enter' to confirm or 'Escape' to cancel)". The main area shows a portion of an HTML file with the following code:

```
index.html
<html>
  <body>
    <div id="wrapper">
      <div id="main">
        <section id="Two">
          <div class="container">
            <h2>Participants</h2>
            <div class="features">
              <article>
                <div class="speaker">
                  <div class="inner">
```



A screenshot of a terminal window. The title bar of the window is circled in red and contains the text "timeTable-info". The main area of the terminal shows several lines of text, with the last few lines being:

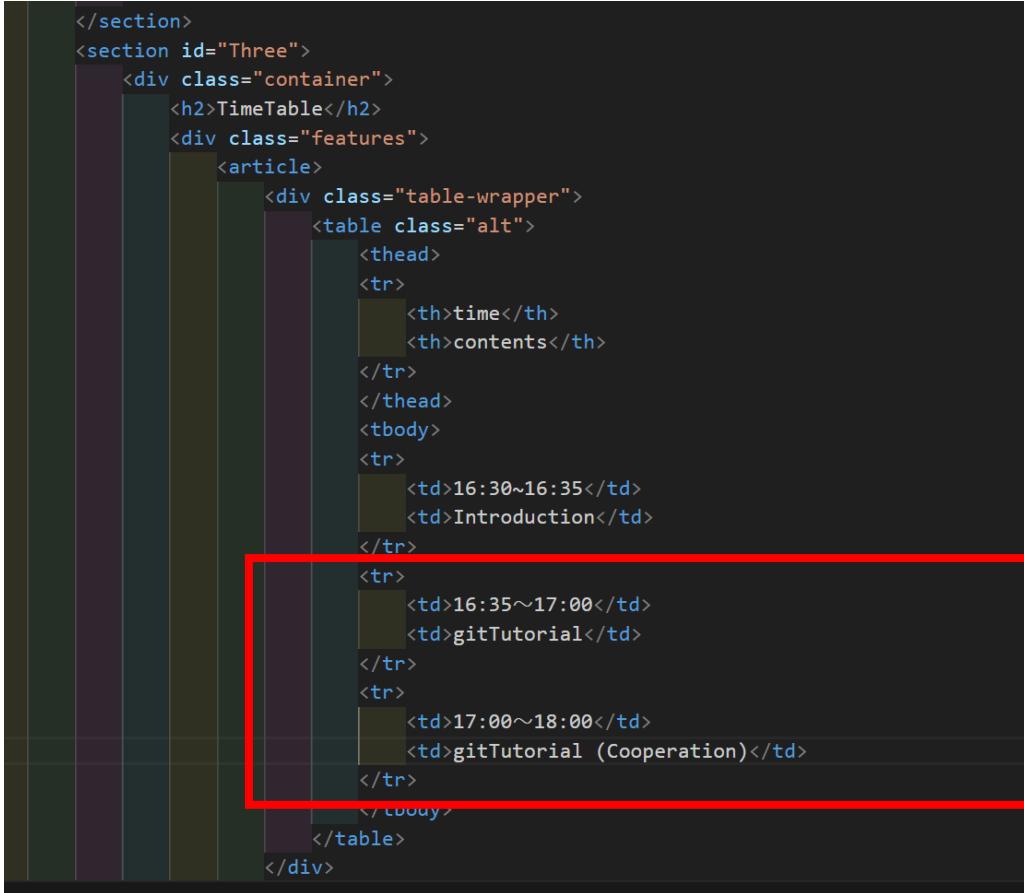
```
80
81
82
83
84
```

Below the terminal window, the status bar shows the path "Live Share" followed by a redacted URL.

Confirmation that the branch is replacing the branch you created.

Change file contents

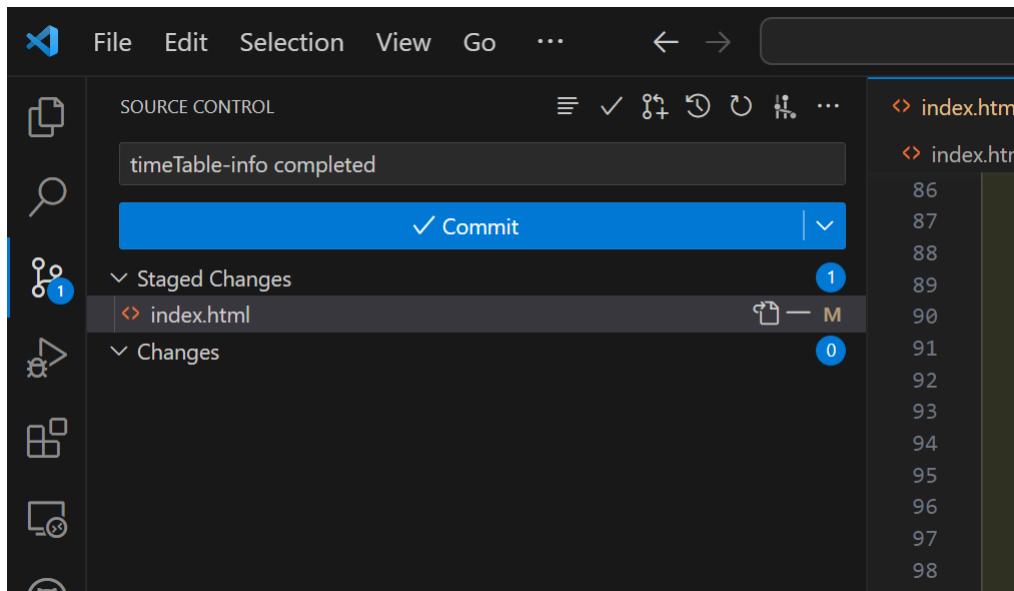
- Open "index.html" and change the section id="Three" file as follows



```
</section>
<section id="Three">
  <div class="container">
    <h2>TimeTable</h2>
    <div class="features">
      <article>
        <div class="table-wrapper">
          <table class="alt">
            <thead>
              <tr>
                <th>time</th>
                <th>contents</th>
              </tr>
            </thead>
            <tbody>
              <tr>
                <td>16:30~16:35</td>
                <td>Introduction</td>
              </tr>
              <tr>
                <td>16:35~17:00</td>
                <td>gitTutorial</td>
              </tr>
              <tr>
                <td>17:00~18:00</td>
                <td>gitTutorial (Cooperation)</td>
              </tr>
            </tbody>
          </table>
        </div>
      </article>
    </div>
  </div>
</section>
```

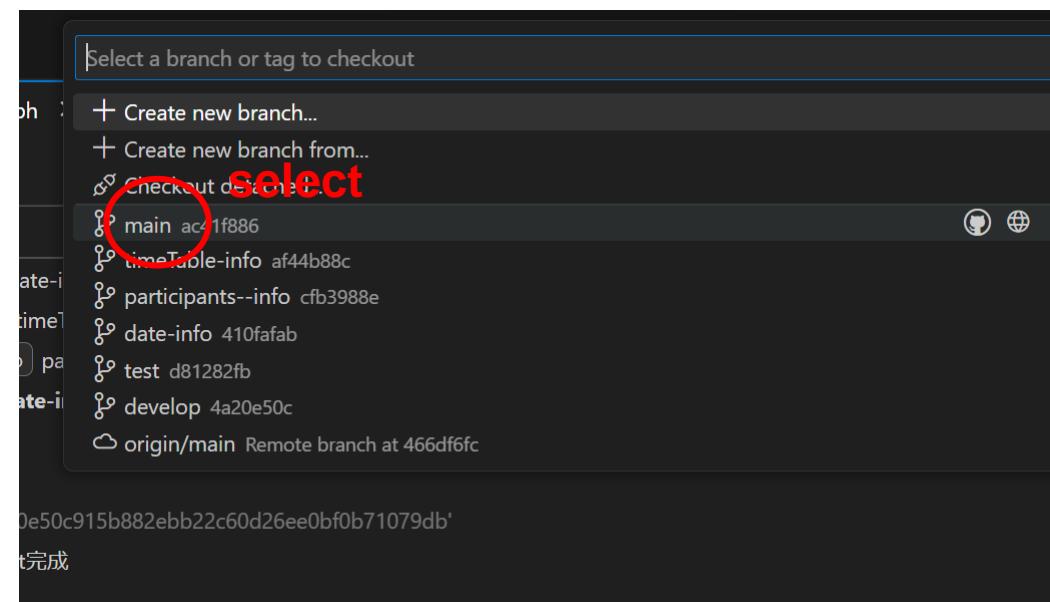
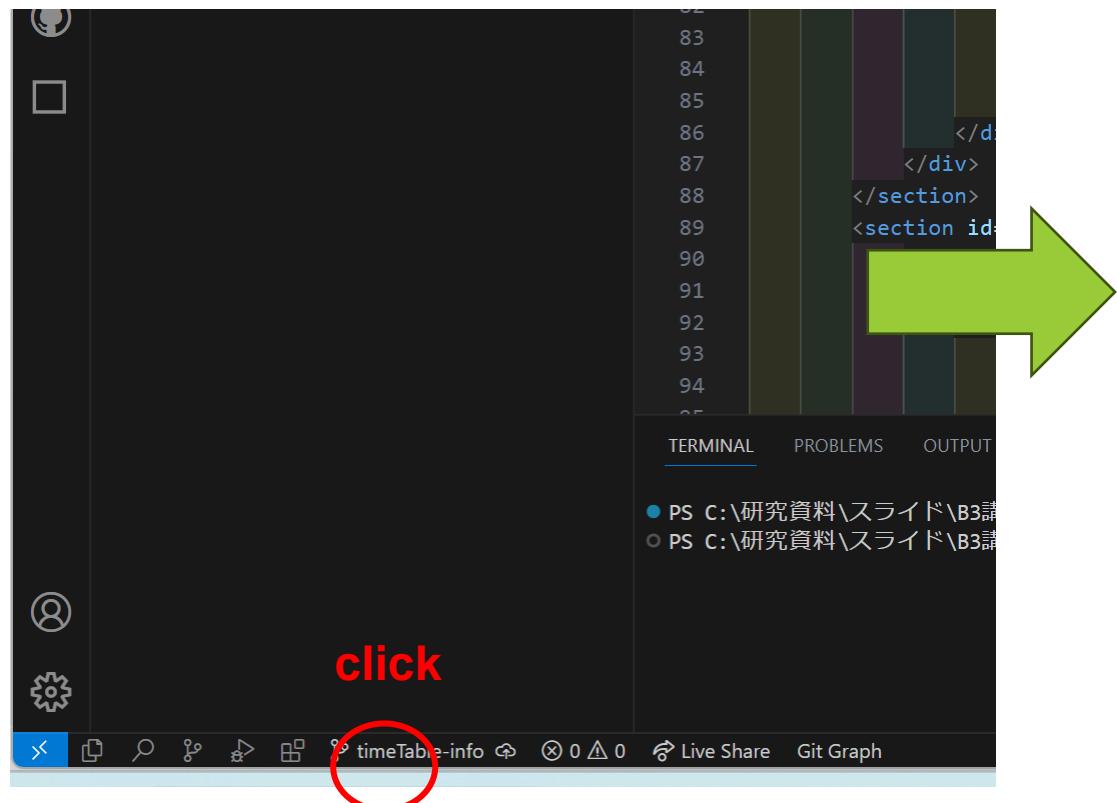
Commit

- Committing Changes

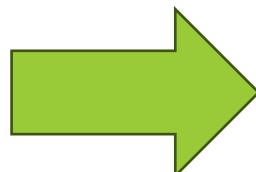
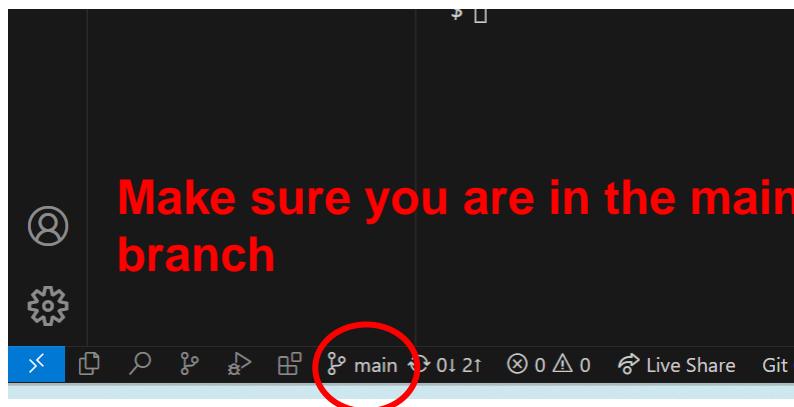


merge into main branch

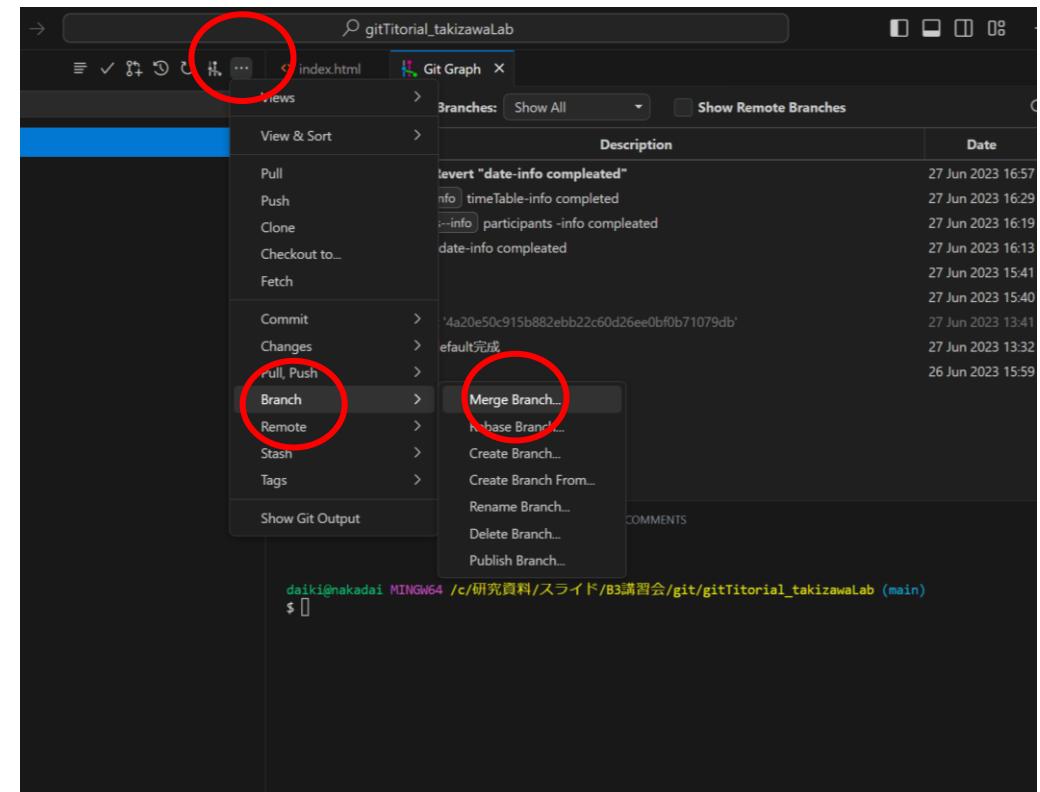
- Move to main branch



merge into main branch

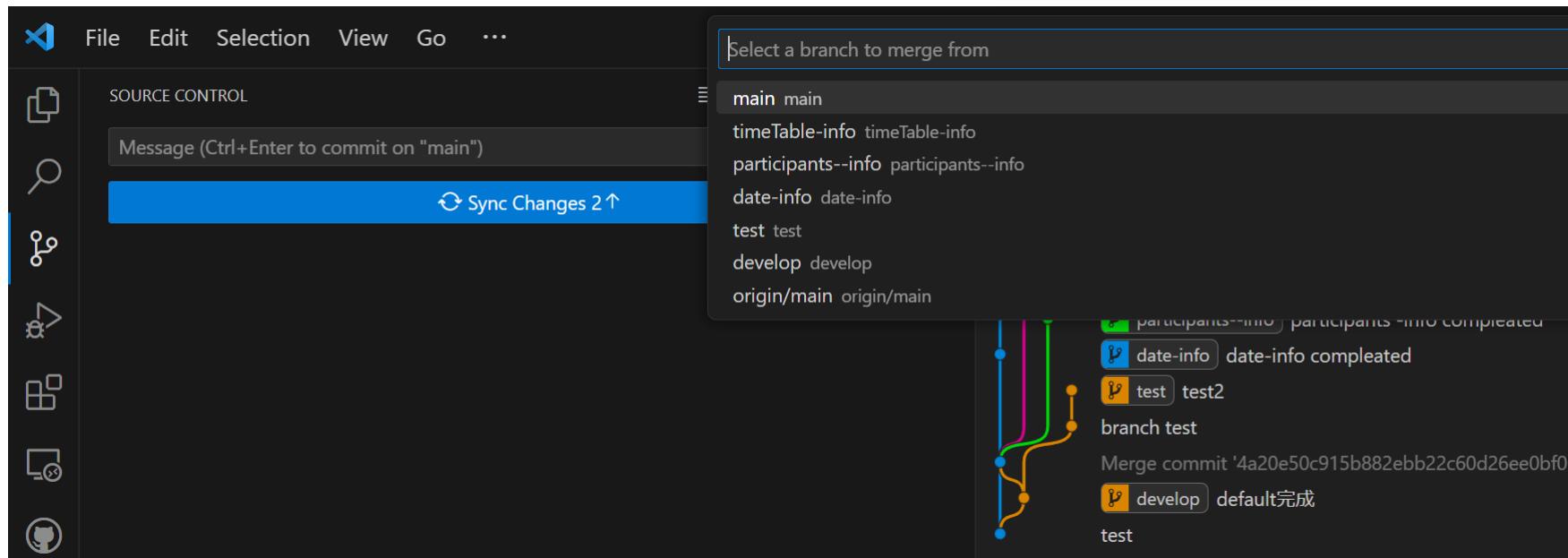


“Merge Branch” select



merge into main branch

- Select the branch you want to merge
- “timetable-info” select



confirmation

- Check if the changes are reflected in "index.html" in the main branch

The screenshot shows a code editor interface with the file 'index.html' open. A red box highlights the content of the table, specifically the rows for 'TimeTable' and 'gitTutorial'. A red circle highlights the 'main' tab in the bottom navigation bar.

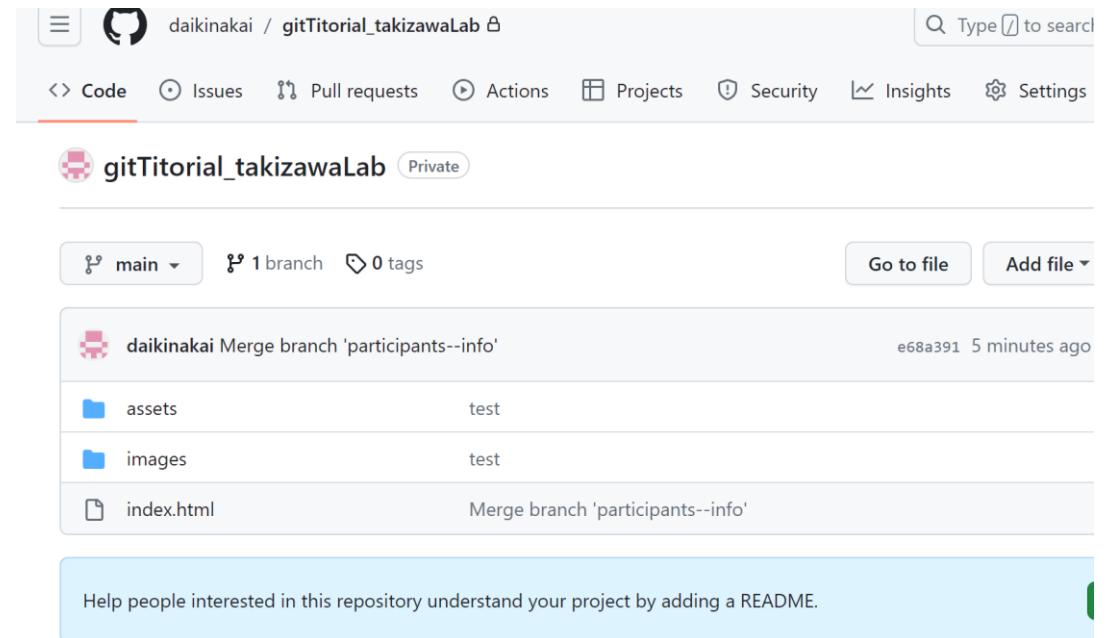
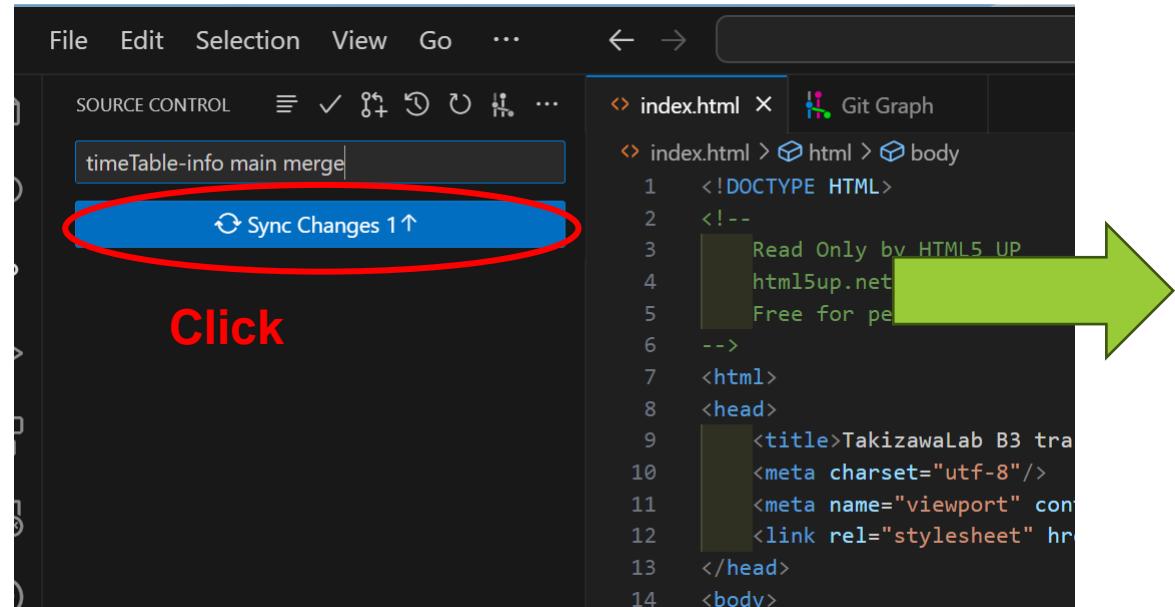
```
<div class="container">
  <h2>TimeTable</h2>
  <div class="features">
    <article>
      <div class="table-wrapper">
        <table class="alt">
          <thead>
            <tr>
              <th>time</th>
              <th>contents</th>
            </tr>
          </thead>
          <tbody>
            <tr>
              <td>16:30~16:35</td>
              <td>Introduction</td>
            </tr>
            <tr>
              <td>16:35~17:00</td>
              <td>gitTutorial</td>
            </tr>
            <tr>
              <td>17:00~18:00</td>
              <td>gitTutorial (Cooperation)</td>
            </tr>
          </tbody>
        </table>
      </div>
    </article>
  </div>
</div>
```

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE COMMENTS

```
PS C:\研究資料\スライド\B3講習会\git\gitTutorial_takizawaLab> git merge --abort
PS C:\研究資料\スライド\B3講習会\git\gitTutorial_takizawaLab> []
```

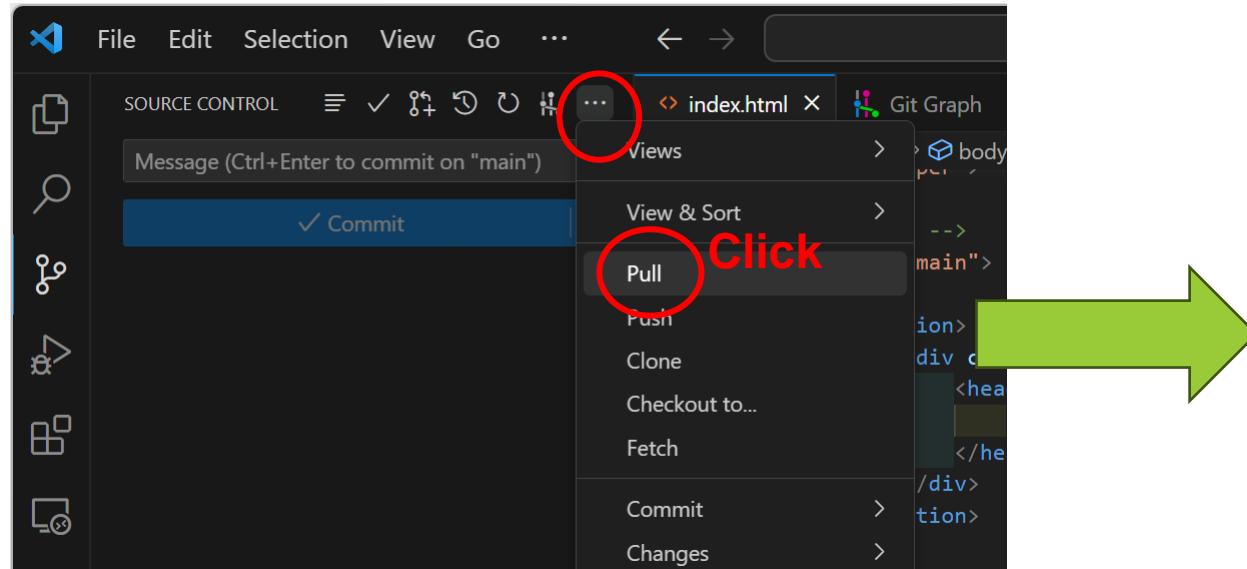
Push: Reflect in Remote

Open Github and check



これ以降はみんなで作業します
We will work together from
this point forward.

Pull (Remoteの状況をlocalに反映) (Reflects Remote status local)



Open Github and check

A screenshot of a GitHub repository page for 'gitTitorial_takizawaLab'. The repository is private. It shows one branch ('main') and zero tags. A recent merge commit by 'daikinakai' has been pushed, merging the 'participants--info' branch. The commit message is 'Merge branch 'participants--info''. The commit hash is 'e68a391' and it was pushed 5 minutes ago. The commit details show files 'assets', 'images', and 'index.html' were updated. A message at the bottom encourages adding a README.

File	Test
assets	test
images	test
index.html	Merge branch 'participants--info'

まとめ みんなでGitHub編

- GitHubもつかってチームでの共同開発もできる！
 - Clone : 共有リポジトリを手元にコピー
 - Push : ローカルの変更を共有リポジトリに反映
 - Pull : 共有リポジトリの変更をローカルに反映
- Git Graph きれいですね
- You can also use GitHub to collaborate with your team!
 - Clone : Copy shared repository to local
 - Push : Reflect local changes to the shared repository
 - Pull : Reflect changes in the shared repository to the local repository
- Git Graph is beautiful!

おつかれさんでした！！！！！

補助スライド&時間が余った 時の補足説明

GitHubの権限について

- ・全員がPushできるわけではない
- ・b3はstudy_sessionのWrite権限をもってた
- ・Pushできないな？と思ったら自分の権限に注意

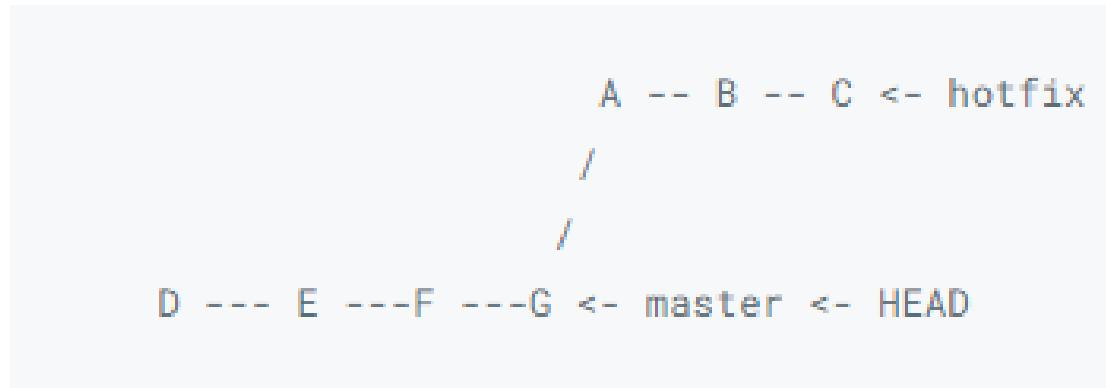
Role: Write ▾

Choose role	Manage roles
Read Can read and clone this repository. Can also open and comment on issues and pull requests.	
Triage Can read and clone this repository. Can also manage issues and pull requests.	
✓ Write Can read, clone, and push to this repository. Can also manage issues and pull requests.	
Maintain Can read, clone, and push to this repository. They can also manage issues, pull requests, and some repository settings.	
Admin Can read, clone, and push to this repository. Can also manage issues, pull requests, and repository settings, including adding collaborators.	

Fast-Forwardとは？

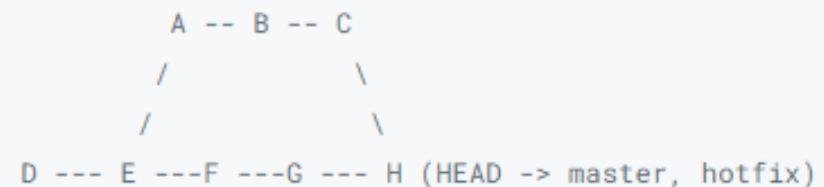
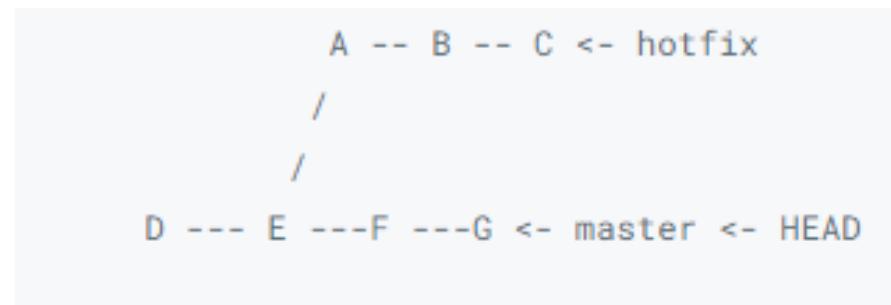
マージするときに元のブランチに連続させてそのままくっつける

- **Fast-Forwardする**



D --- E --- F --- G --- A --- B --- C (HEAD -> master, hotfix)

- **Fast-Forwardしない**



トラブルシューティング

- VSCodeでのpushができない
 - ・ターミナル上ではできる
- `git config --global core.sshCommand "C:/Windows/System32/OpenSSH/ssh.exe"`
- 公開鍵認証ができているか確認