

Github ワークショップに向けての 事前準備マニュアル

Hack U Project

本イベントへの参加ありがとうございます!!

イベントに参加していただくにあたって

以下の6つの事前準備をしていただく必要があります

1. ワークショップでのZoomのセットアップ
2. ワークショップでのSlackのセットアップ
3. GitHubのアカウント作成
4. Sourcetreeのインストールとセットアップ
5. 演習準備
6. GitHubへの疎通確認



事前準備 1

ワークショップで利用する
Zoom のセットアップ

お願い（アプリ上の名前について）

イベントではZoomやSlackなど、複数のアプリケーションを使います。

それぞれのアプリの準備手順をこれから案内しますが、

そこで使うお名前（表示名）は、Connpassの「表示名」と必ず揃えてください。（参考）Connpassプロフィールページの「表示名」



学校など、本イベント以外で利用しているZoomやSlackには影響がありませんので、ご安心ください。

(事前準備1) ZOOM

ZoomはZoom Video Communications社が開発したWeb会議サービスです。

PC、Android、iOS、ブラウザから利用可能です。

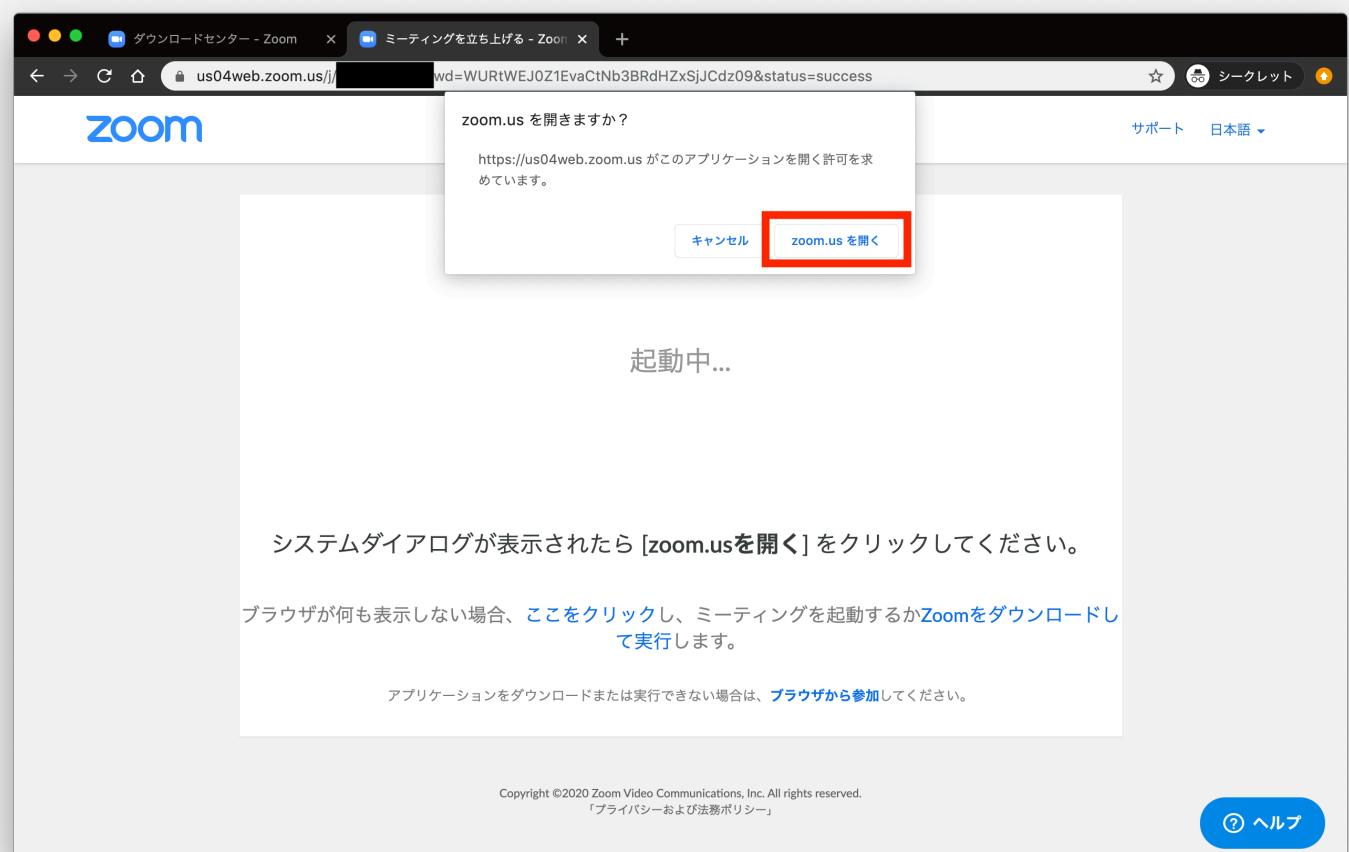
今回のオンラインイベントはZoom（とSlack）を利用します。



(事前準備 1) 利用方法

別途配布したURLをPCのブラウザから開くと、右図の画面になります。

すでにZoomがインストールされていれば、[zoom.usを開く]でアプリを起動してください。



(事前準備1) アプリインストール



※インストールが済んでいない方のみ

[Zoomをダウンロードして実行]のからインストーラーをダウンロードできます。

インストーラーの指示に従い、インストールを完了させてください。

完了後、もう一度リンクを開いて
[zoom.usを開く]からアプリを起動してください。

(事前準備 1) 手動入力で参加する場合 (1/3)



もしURLを開いてもポップアップ
ウィンドウが出てこない場合は、
アプリを起動して、[ミーティングに参加]から部屋番号を
手入力することで参加できます。

アカウントは必須ではないです
が、アカウントをお持ちの方は
[サイン イン]してからミーティ
ングに参加できます。

(事前準備 1) 手動入力で参加する場合 (2/3)



別途配布したURLに入っている
9~10桁の数字を[会議IDまたは
会議室名]に入力してください。

次に[名前]の欄に、お名前
(Connpassの表示名) を入力
した上で[参加]ボタンを押しま
す。

(事前準備 1) 手動入力で参加する場合 (3/3)



もしパスワードの入力が求められたら、メールにてご連絡しているパスワードを貼り付けてください。

以上で入室の手続きが完了します。

(事前準備1) マイク・カメラチェック

入室したら右図のような画面になります。

画面の左下にマイクとカメラの
オン/オフを切り替えるボタン
があります。

イベントが開始する前に簡単に
マイクとカメラの接続をご確認
してください。



(事前準備 1) connpass の表示名について

connpass のプロフィールページにおける、
この部分の表示名を Slack, zoom に
必ず 登録するようにしてください。

Hack U (Yahoo! JAPAN)

(HackU)



プログラミングやデザインを通して、あなたのアイデアを形にしてみませんか？

Hack U（ハック・ユー）は、限られた期間の中で、学生がプロダクトを自ら企画・開発・発表するハッカソンや各種ワークショップなどを開催しています。

ものづくりの楽しさを体現できるよい機会となるよう、ヤフーの現役社員が学生のみなさんを全力でサポートいたします！



プロフィールを変更する

利用開始日: 2018/05/22



事前準備 2

本ワークショップで利用する
Slack のセットアップ

(事前準備2) 本マニュアルの流れ

1. Slackのワークスペースに参加しよう
2. 本イベントでのSlackチャンネルの使い方

(事前準備2) Slackのワークスペースに参加しよう

- Slackの招待URLからアクセス
- 「HackUイベント企画(または hacku-event)に参加する」と表示されていることを確認して、自分のメールアドレスを記入
- 「メールをチェックしてください」という表示が出たら、登録したメールアドレスに書かれた「ここをクリックして続行」を押す
- 氏名 (**Connpassの表示名**) とパスワードを入力してアカウントを作成



The image displays two screenshots of the Slack sign-up process. The left screenshot shows the initial step where the user is invited to join the 'HackUイベント企画' workspace. It asks for the user's email address ('あなたのメールアドレスは?') and provides a field with the placeholder '自分の名前@example.com'. A green button labeled 'メールアドレスを確認する' (Check email address) is at the bottom. The right screenshot shows the final step of creating the account. It asks for the user's name ('氏名') and password ('パスワード (必須)'). It includes a note that the name will be displayed next to messages. A checkbox for receiving service-related emails is checked. A large grey button labeled 'アカウントを作成する' (Create account) is at the bottom. Both screenshots feature the Slack logo and a blue sidebar with icons representing users and files.

(事前準備2) connpass の表示名について

connpass のプロフィールページにおける、
この部分の表示名を Slack, zoom に
必ず 登録するようにしてください。

Hack U (Yahoo! JAPAN)

(HackU)



プログラミングやデザインを通して、あなたのアイデアを形にしてみませんか？

Hack U（ハック・ユー）は、限られた期間の中で、学生がプロダクトを自ら企画・開発・発表するハッカソンや各種ワークショップなどを開催しています。

ものづくりの楽しさを体現できるよい機会となるよう、ヤフーの現役社員が学生のみなさんを全力でサポートいたします！



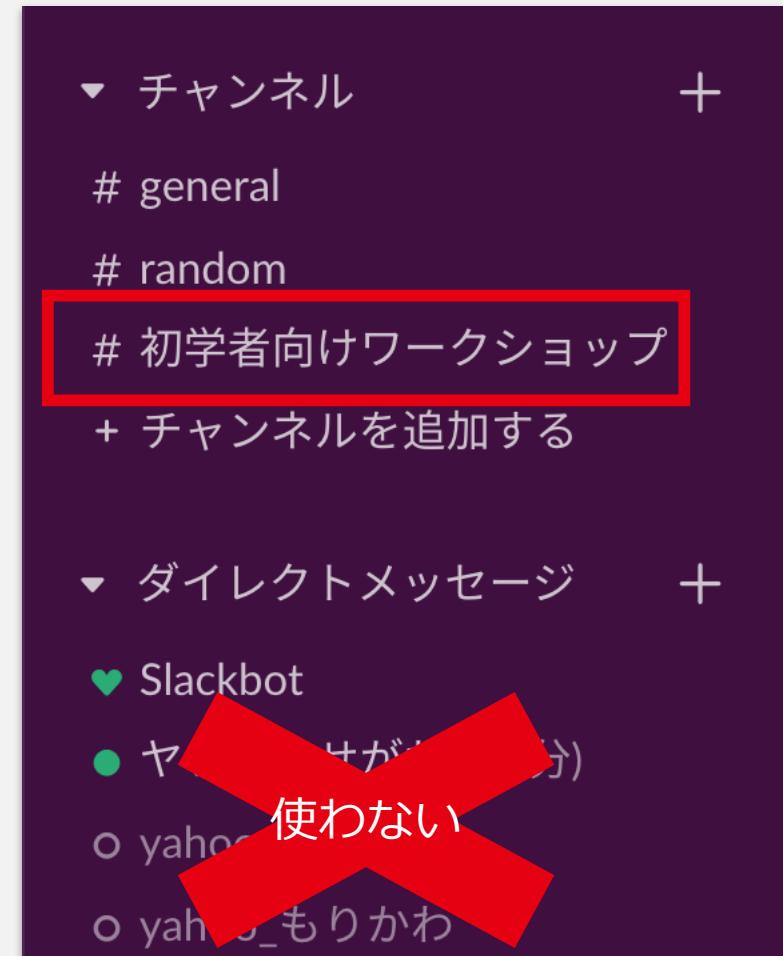
プロフィールを変更する

利用開始日: 2018/05/22



(事前準備2) 本イベントでのSlackチャンネルの使い方

- 運営からの全体連絡
「# 初学者向けワークショップ」チャンネル
- 演習中
学生さん数人の「#group_XX」チャンネル
Zoomもそれぞれのグループに招待します



事前準備 3

GitHub のアカウント作成



Git と Github

Git	バージョン管理の方式 (バージョン管理とは、いつ、誰が、どのファイルを、どう変更したかを記録すること)
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かんたんにできる

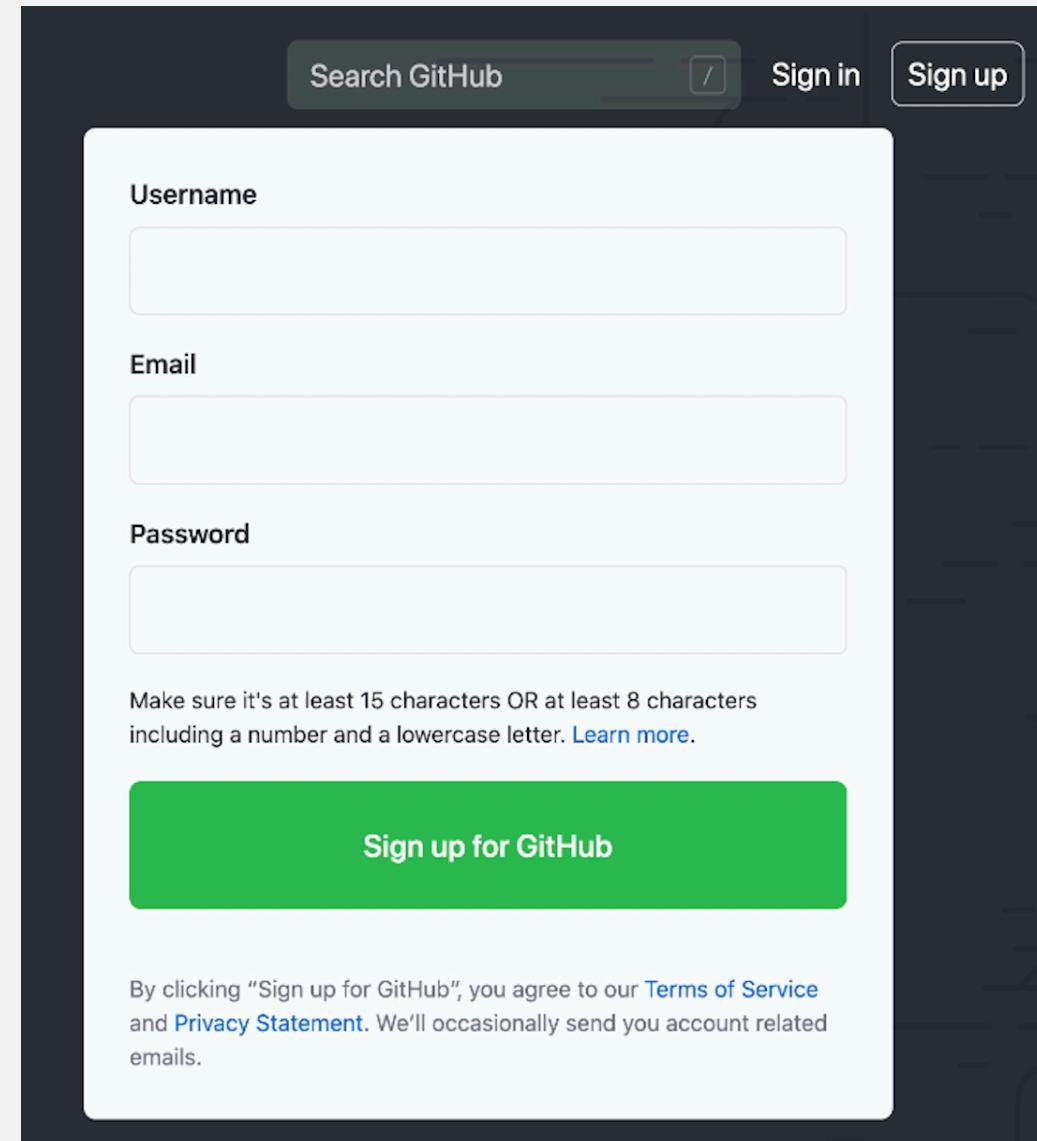
Git を使うには

- コマンドラインでGitコマンドを使う
- GUIツールを使う

今回は Sourcetree という
GUIツールを使います

Github のアカウントを作る

- 右上のSign upをクリック
 - <https://github.com>
- 必要事項を記入する
 - アカウント名
 - メールアドレス
 - パスワード



アカウント名は公開されます

Github のアカウントを作る

- 認証
- Welcome to GitHub
 - 仕事は何か
 - どのくらいプログラミングができるか
 - 何のためにGitHubを使うか
 - 興味があるもの
- メールを確認 “Verify email address” でアクセス

Github のアカウントを確認する

- ・ プロフィールページにアクセス
 - ・ <https://github.com/アカウント名>

アカウント作成されていない場合、404になります

リポジトリを作る

Create repositoryをクリック

Owner には先程作った
アカウント名が入る

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Owner *  アカウント名 / **Repository name ***

Great repository names are short and memorable. Need inspiration? How about [scaling-eureka](#)?

Description (optional)

 **Public**
Anyone on the internet can see this repository. You choose who can commit.

 **Private**
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

Initialize this repository with a README
This will let you immediately clone the repository to your computer.

Add .gitignore: None ▾ | Add a license: None ▾ | ⓘ

Create repository

リポジトリを作る

Create repositoryをクリック

Owner には先程作った
アカウント名が入る

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner * Repository name *

Great repository names are short and memorable. Need inspiration? How about [scaling-eureka](#)?

Description (optional)

Public
Anyone on the internet can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

Initialize this repository with a README
This will let you immediately clone the repository to your computer.

Add .gitignore: None ▾ | Add a license: None ▾ | ⓘ

Create repository

今回は「**github_ws**」
というリポジトリ名で
作りましょう

リポジトリを作る

Create repositoryをクリック

Owner には先程作ったアカウント名が入る

リポジトリを誰でも参照可能にするかを設定する
(今回は public)

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner * Repository name *

Great repository names are short and memorable. Need inspiration? How about [scaling-eureka](#)?

Description (optional)

Public
Anyone on the internet can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

Initialize this repository with a README
This will let you immediately clone the repository to your computer.

Add .gitignore: None ▾ | Add a license: None ▾ | ⓘ

Create repository

「github_ws」
というリポジトリ名で
作りましょう

事前準備 4

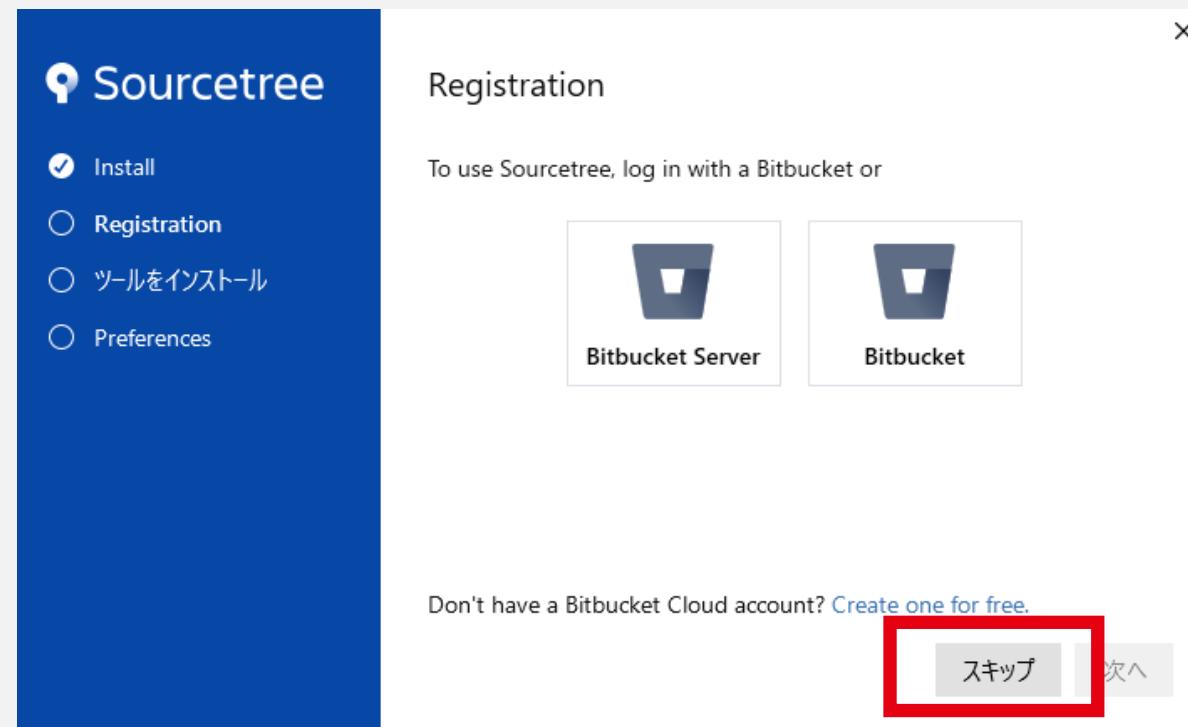
Sourcetree のセットアップ

Sourcetree のインストール (Win)

- Sourcetree のダウンロード
 - <https://www.sourcetreeapp.com/>
- exe ファイルを実行

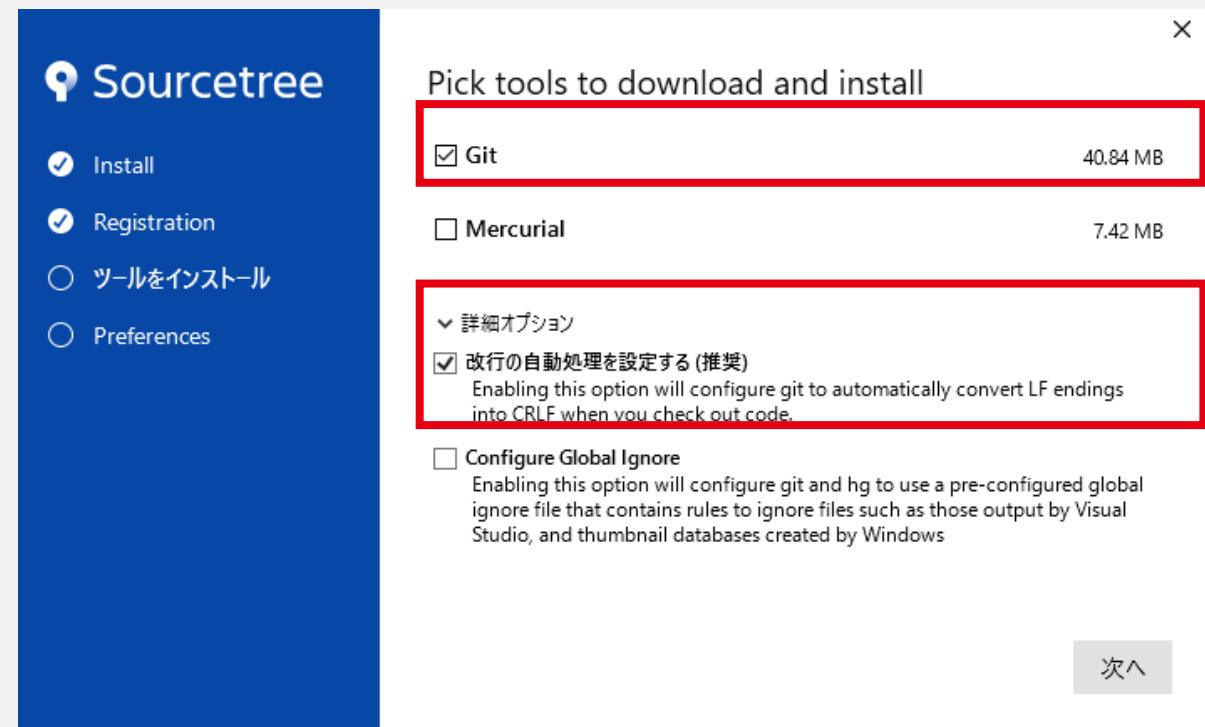
Sourcetree のインストール (Win)

- exe ファイルを起動すると以下のような画面に
- 冒頭の画面はスキップする



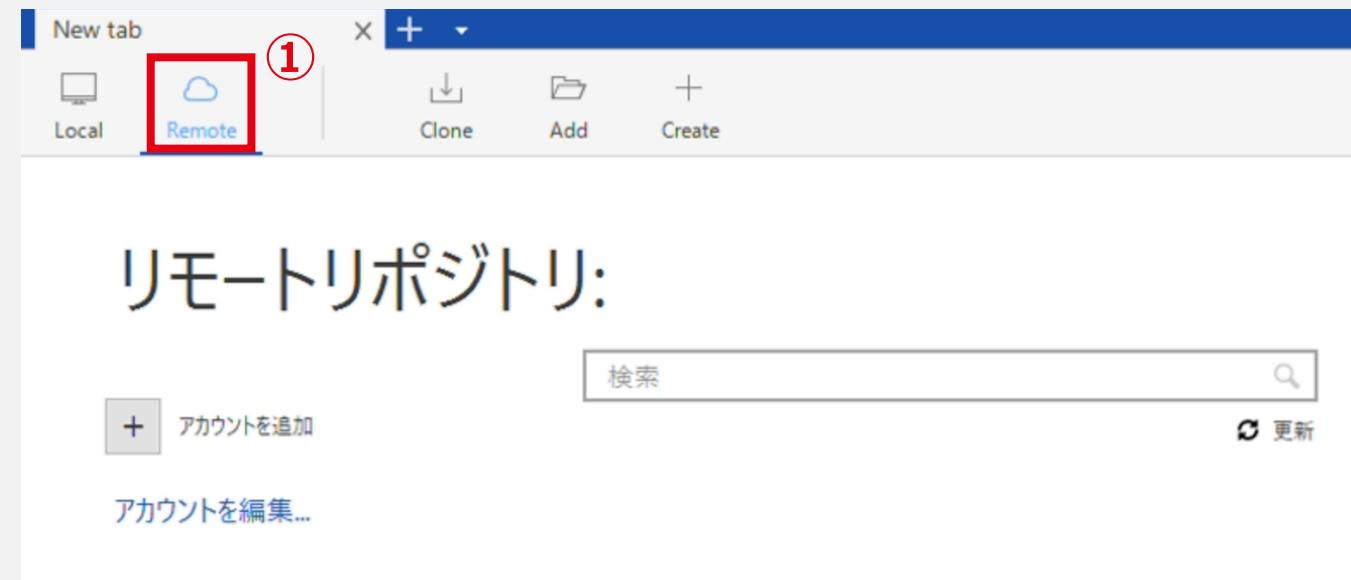
Sourcetree のインストール (Win)

- Registration は **Git** のみチェック
 - ※Gitがインストール済みの場合はチェック不要です
- 詳細オプションを開いて改行の自動処理を設定もチェックしておく
- 次の画面で名前とメールアドレスを聞かれるので設定する



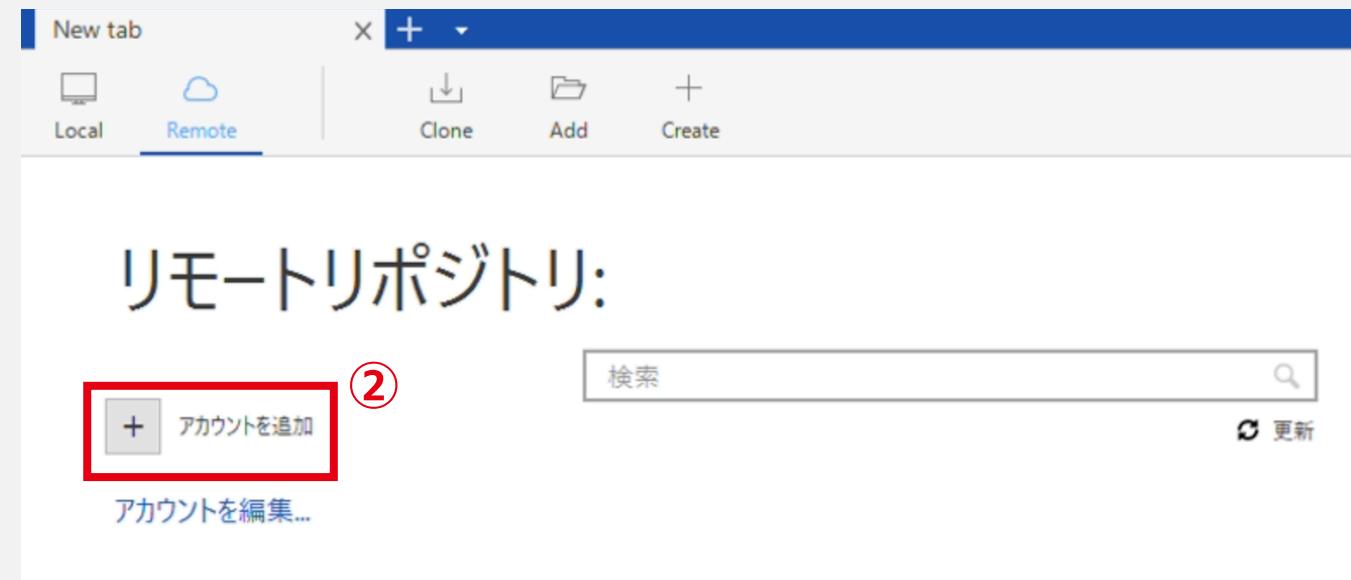
Sourcetree と Github を連携 (Win)

1. リモートタブに移動
2. アカウントを追加
3. ホスティングアカウントの設定を埋める
4. OAuth トークンを再読み込みを押す
5. ブラウザ上で Authorize atlassian を選択
6. OKを押せば完了



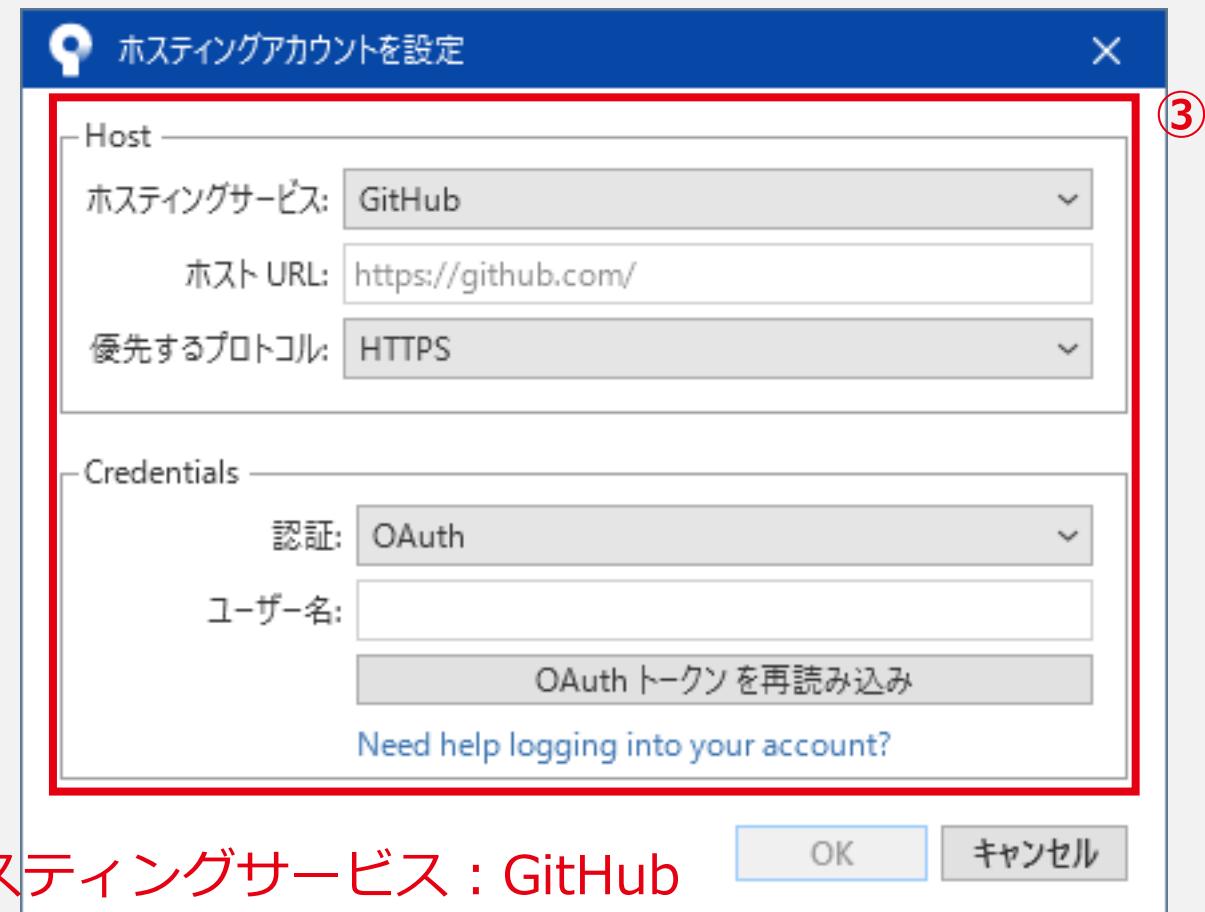
Sourcetree と Github を連携 (Win)

1. リモートタブに移動
2. アカウントを追加
3. ホスティングアカウントの設定を埋める
4. OAuth トークンを再読み込みを押す
5. ブラウザ上で Authorize atlassian を選択
6. OKを押せば完了



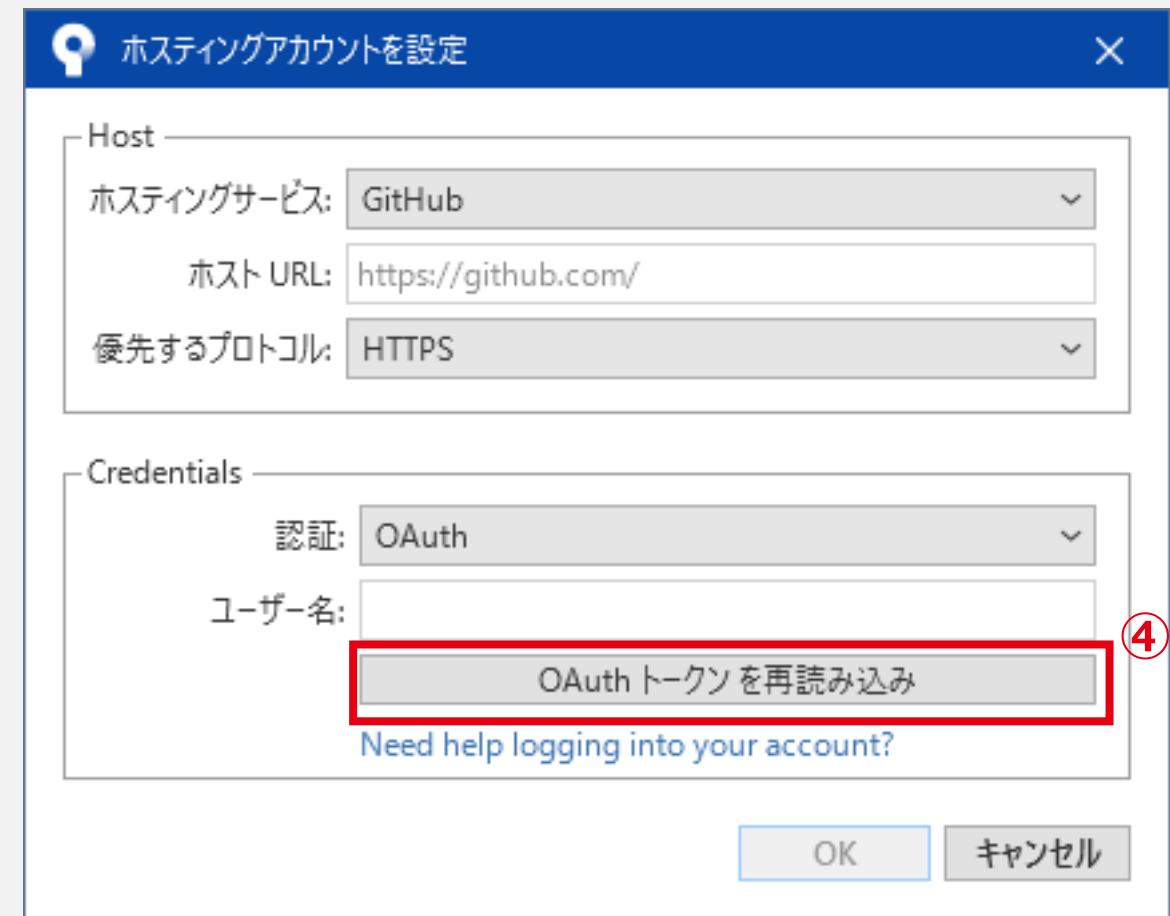
Sourcetree と Github を連携 (Win)

1. リモートタブに移動
2. アカウントを追加
3. ホスティングアカウントの設定を埋める
4. OAuth トークンを再読み込みを押す
5. ブラウザ上で Authorize atlassian を選択
6. OKを押せば完了



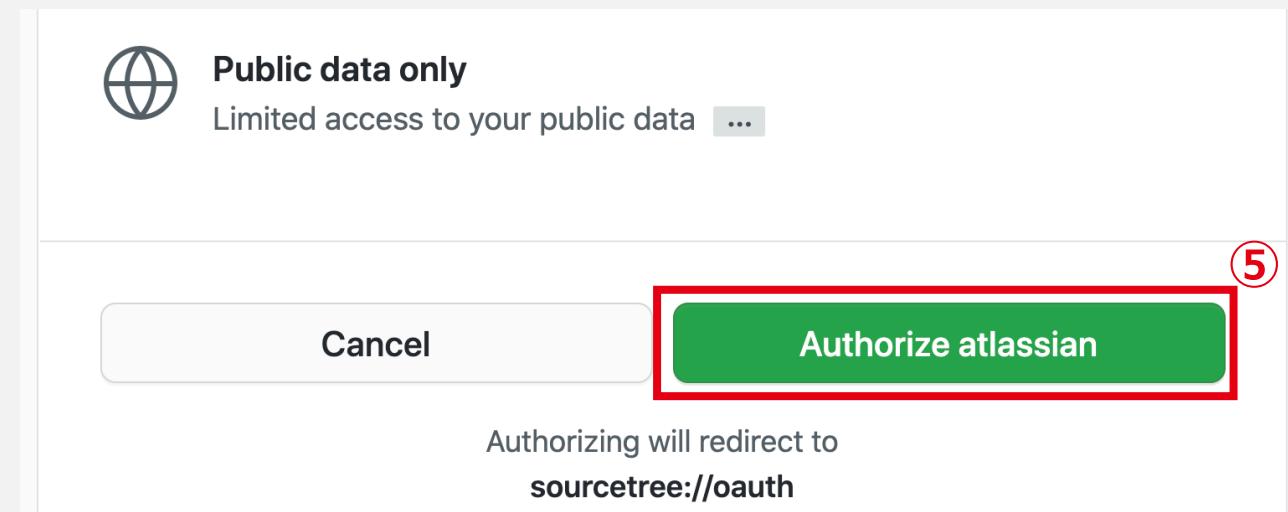
Sourcetree と Github を連携 (Win)

1. リモートタブに移動
2. アカウントを追加
3. ホスティングアカウントの設定を埋める
4. OAuth トークンを再読み込みを押す
5. ブラウザ上で Authorize atlassian を選択
6. OKを押せば完了



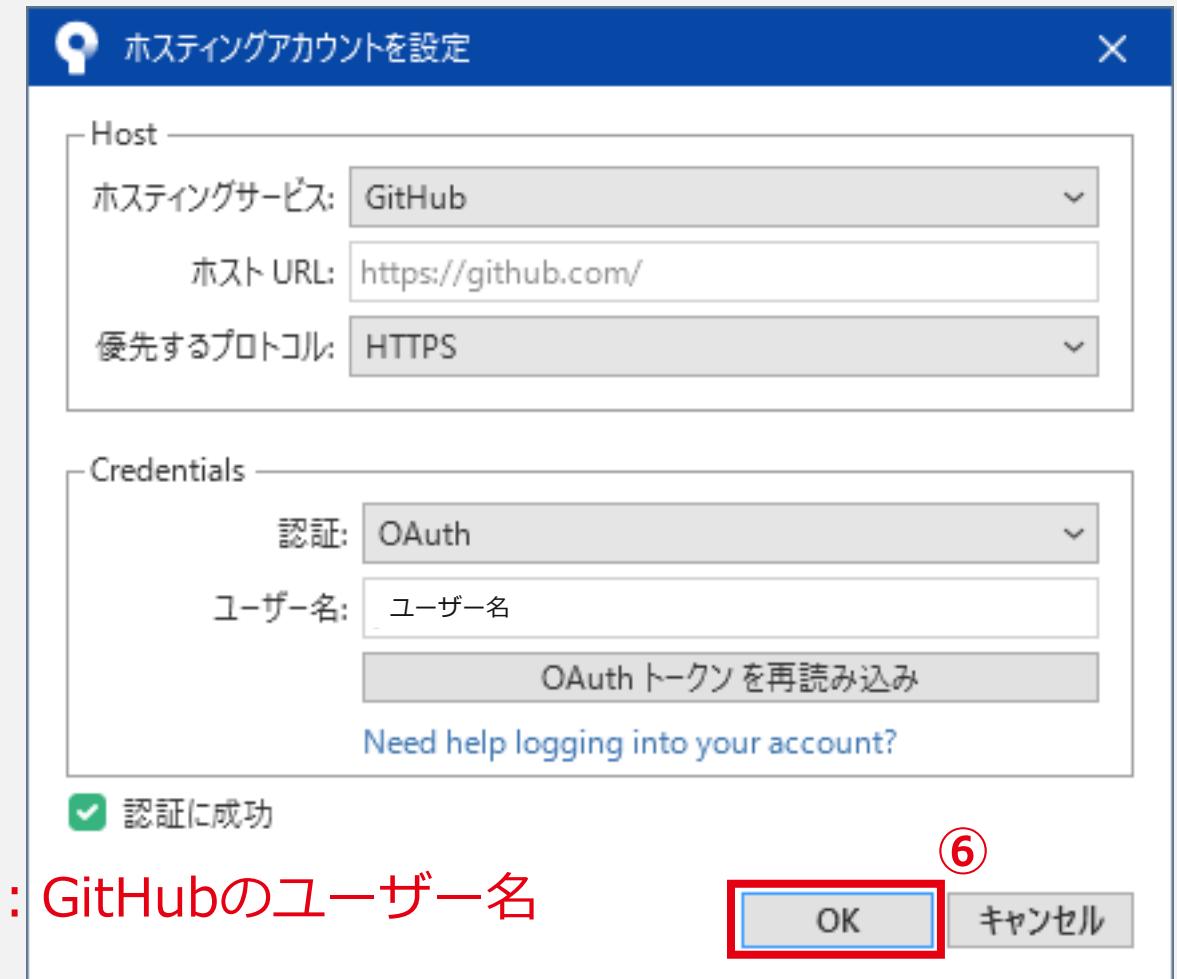
Sourcetree と Github を連携 (Win)

1. リモートタブに移動
2. アカウントを追加
3. ホスティングアカウントの設定を埋める
4. OAuth トークンを再読み込みを押す
5. ブラウザ上で Authorize atlassian を選択
6. OKを押せば完了



Sourcetree と Github を連携 (Win)

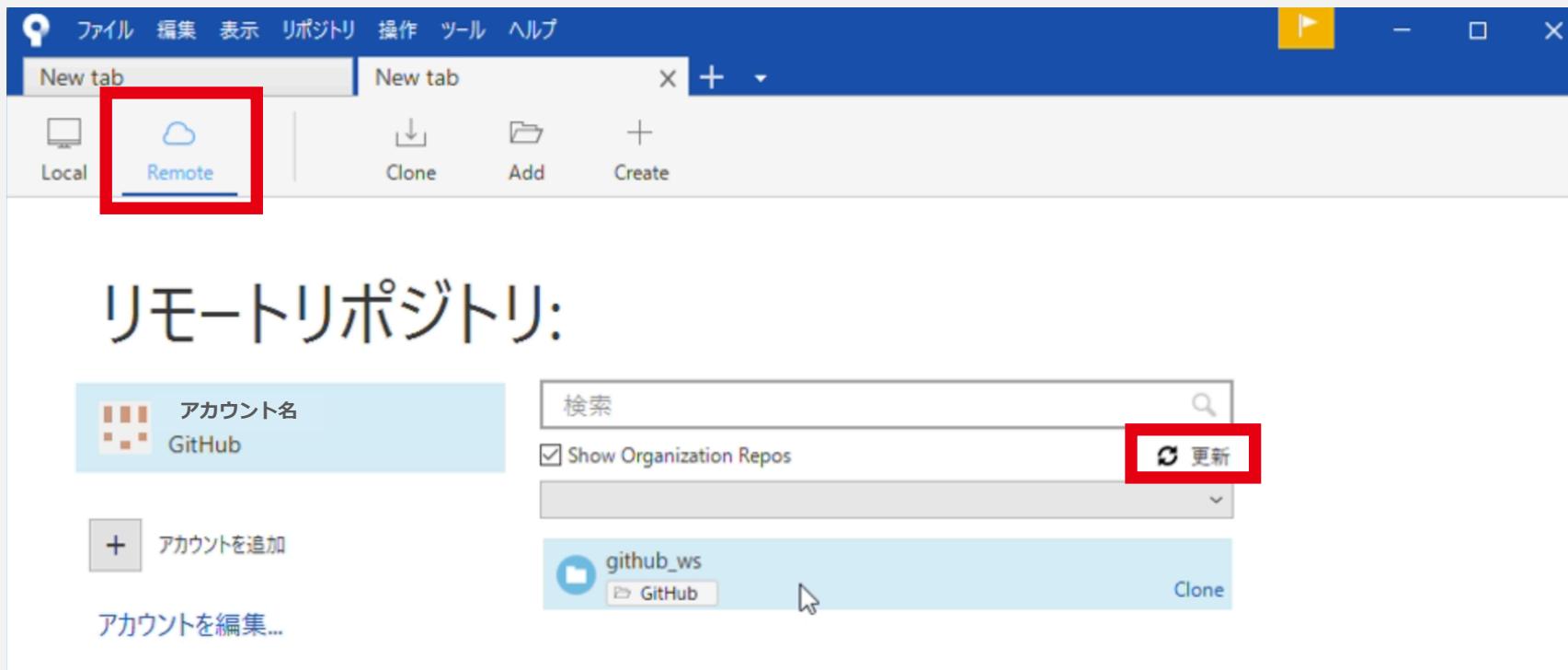
1. リモートタブに移動
2. アカウントを追加
3. ホスティングアカウントの設定を埋める
4. OAuth トークンを再読み込みを押す
5. ブラウザ上で Authorize atlassian を選択
6. OKを押せば完了



認証に成功 となっていることを確認

Sourcetree と Github を連携

アカウント連携後、「リモート」を選択して、先程作ったリポジトリが表示されていれば連携成功🎉
(表示されていない場合、更新をクリックすると出てきます)



事前準備 5

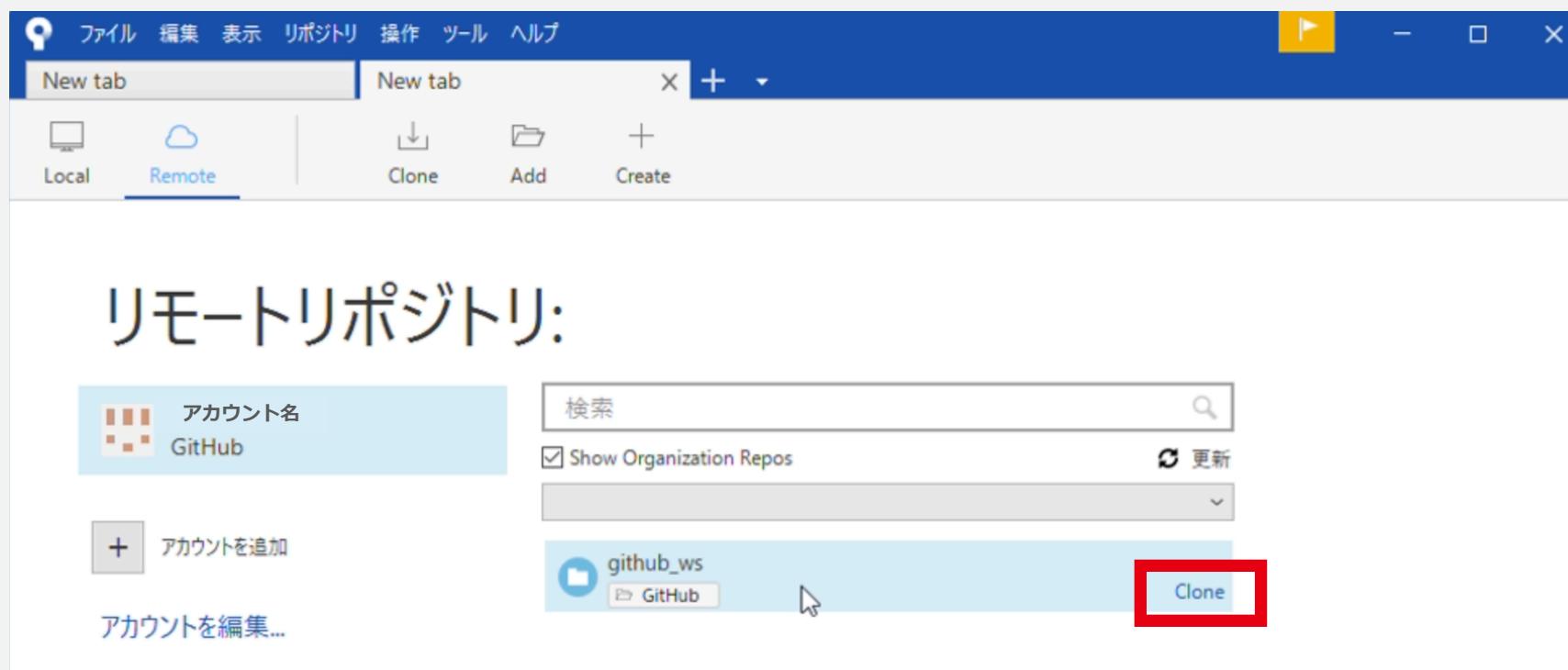
演習準備

ローカルリポジトリの作成

- Git の機能の一つ `clone` を使って、GitHub で作成したリポジトリをローカルにコピーします
- 各Gitの機能やリポジトリについては演習の資料で解説します

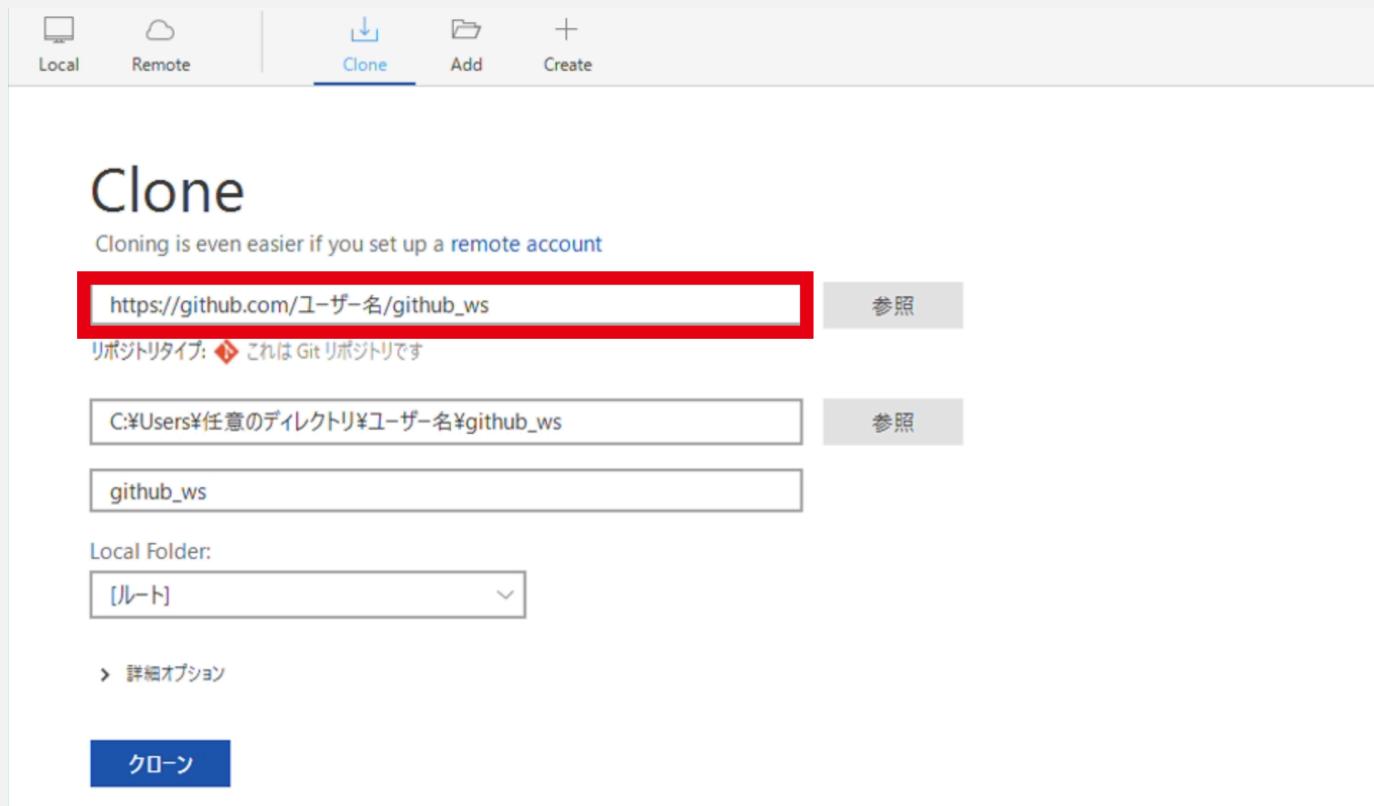
ローカルリポジトリの作成

- 先程の画面からリポジトリの右下にある **Clone** をクリック



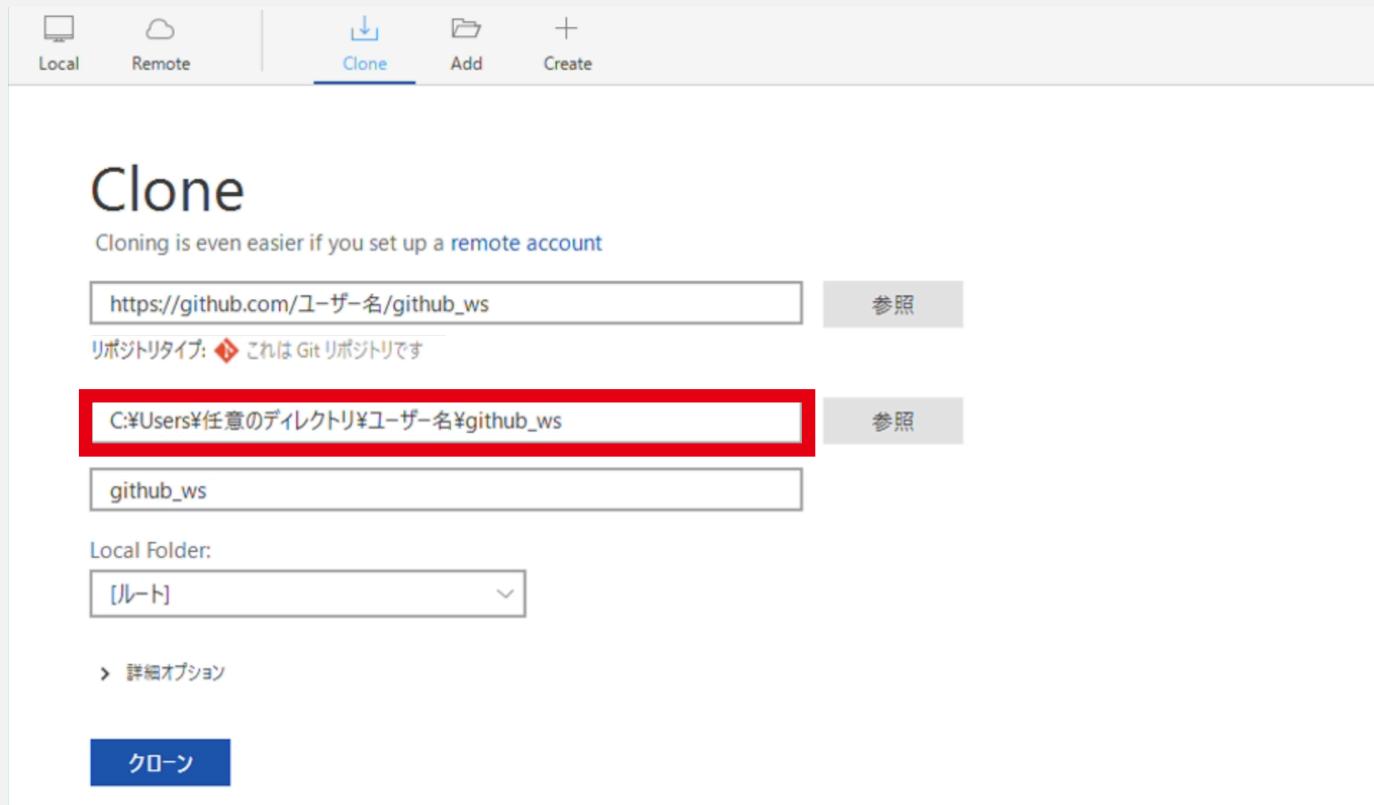
ローカルリポジトリの作成

- 「https://github.com/ユーザー名/github_ws」
となっていることを確認



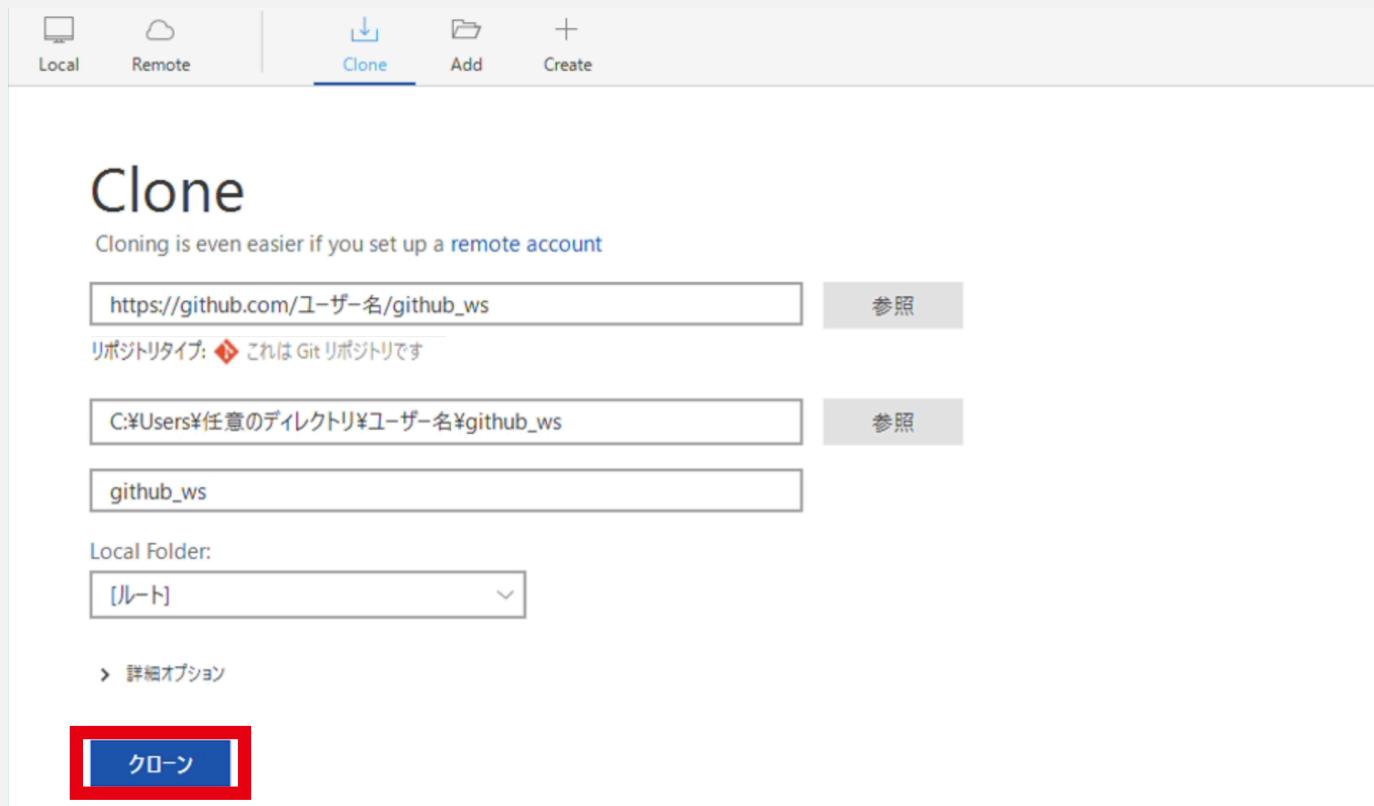
ローカルリポジトリの作成

- ・ ローカルリポジトリをどのパスに作成するか決める



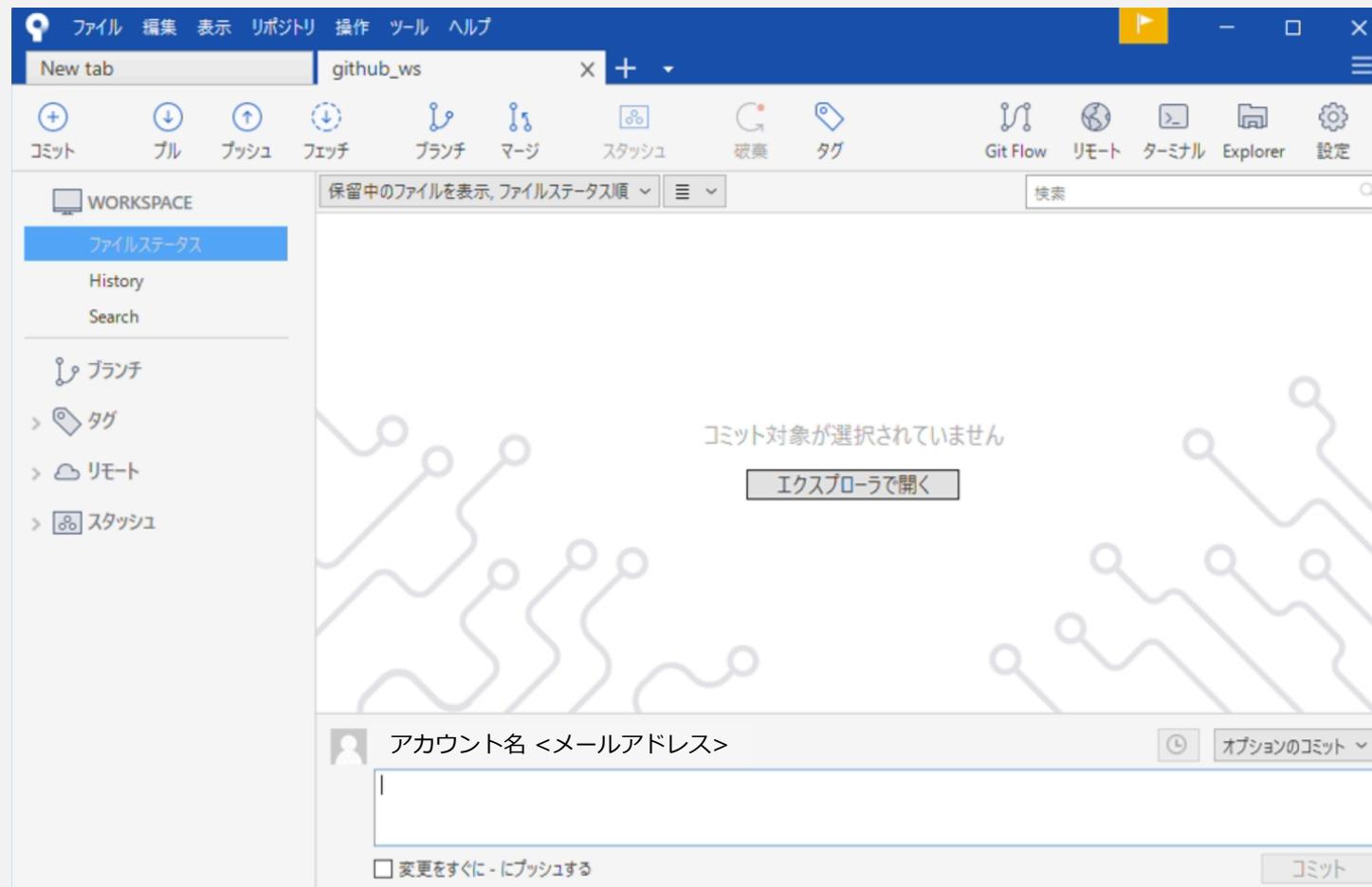
ローカルリポジトリの作成

- ・ クローンを押してローカルリポジトリを作成する

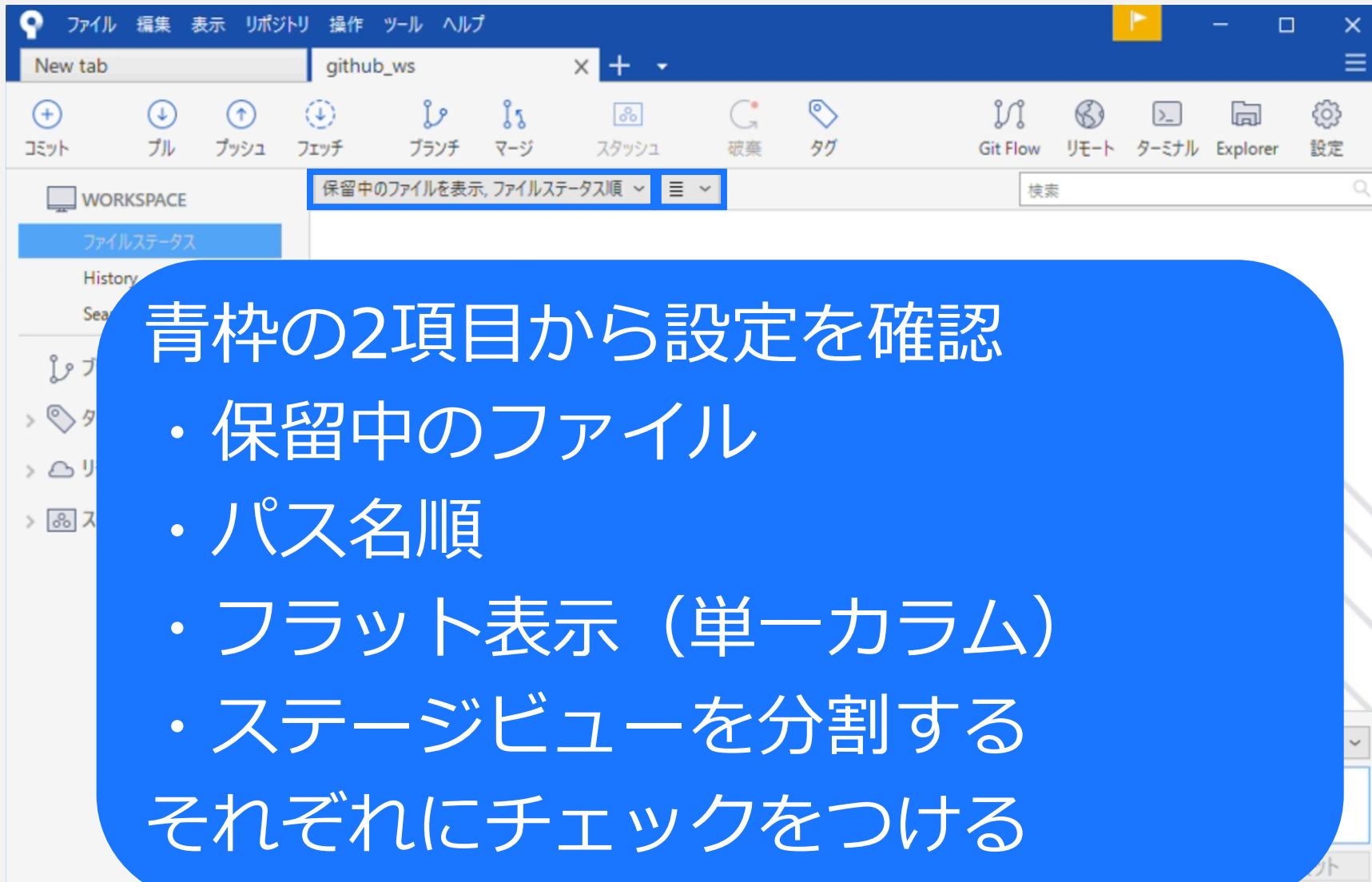


ローカルリポジトリの作成

- このような画面になつたら作成完了です

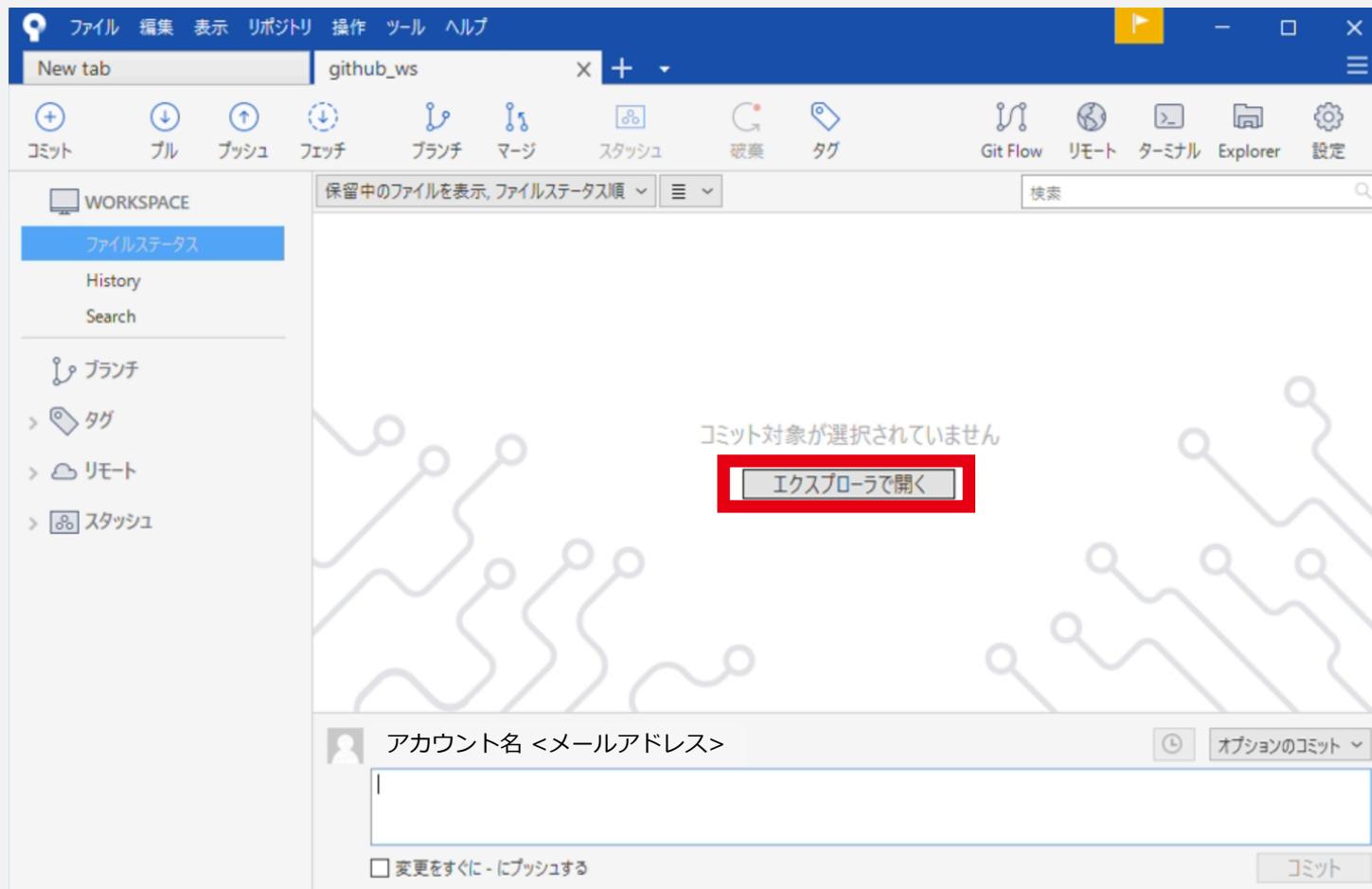


Sourcetreeの設定



演習ファイルの準備

- 先程の画面からローカルリポジトリを開きます



演習ファイルの準備

- GitHub の Search or jump to … から「hackujp/github_tutorial」を検索
- リポジトリ内の ver_GUI/hands-on.zip を開く
 - Download ボタンよりダウンロード
- hands-on.zip を解凍

演習ファイルの準備

- ・ 先程開いたローカルリポジトリに展開したファイルを移動させる
- ・ 移動させる対象は以下の通り
 - ・ index.html
 - ・ src

事前準備 6

GitHubへの疎通確認

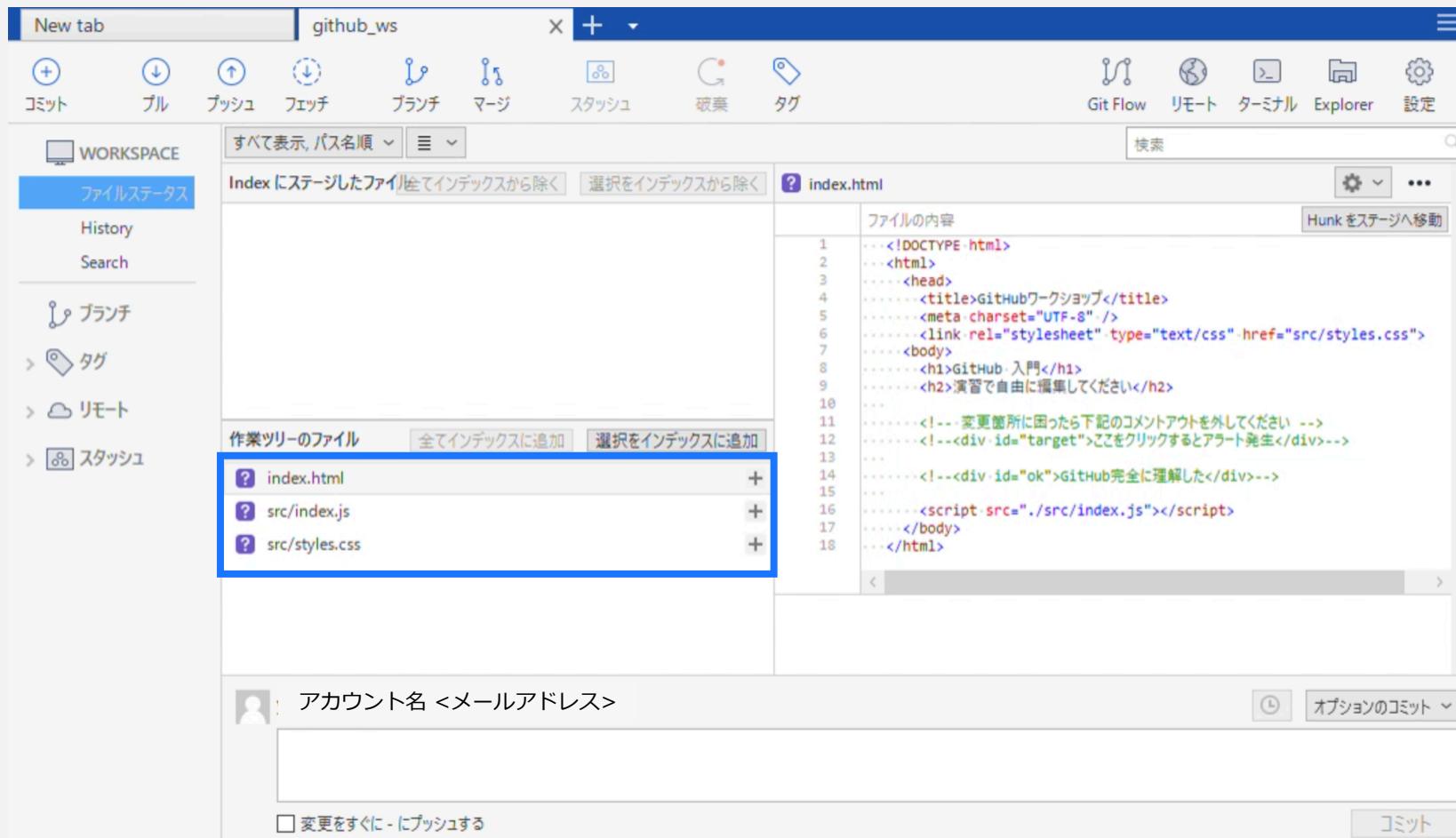


GitHubへの疎通確認

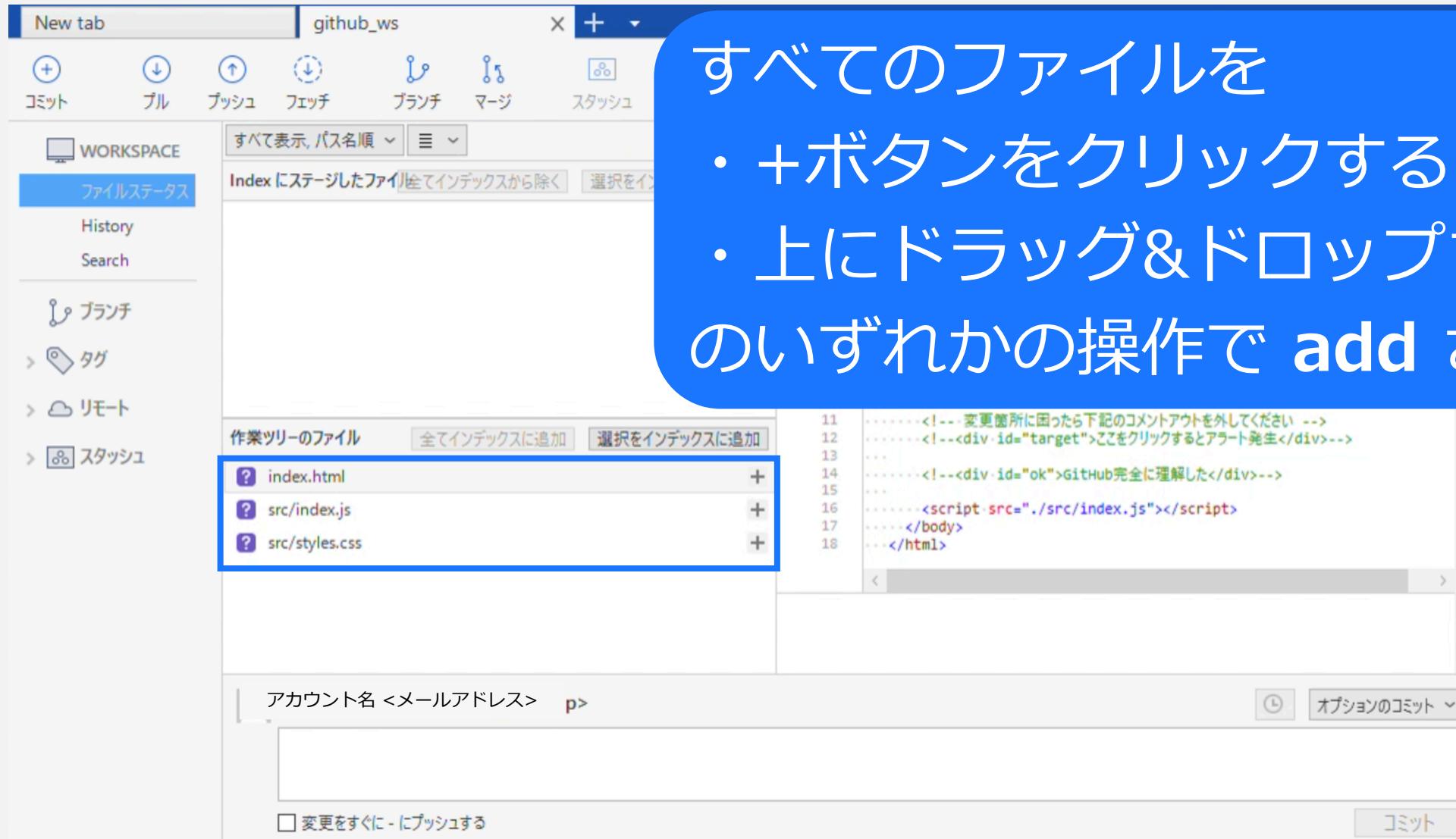
- GitHub に演習ファイルをアップロードして疎通確認を行います
- Git の基本操作 add/commit/push を使用します

GitHubへの疎通確認 (add)

- SourceTree で演習ファイルがあることを確認



GitHubへの疎通確認 (add)

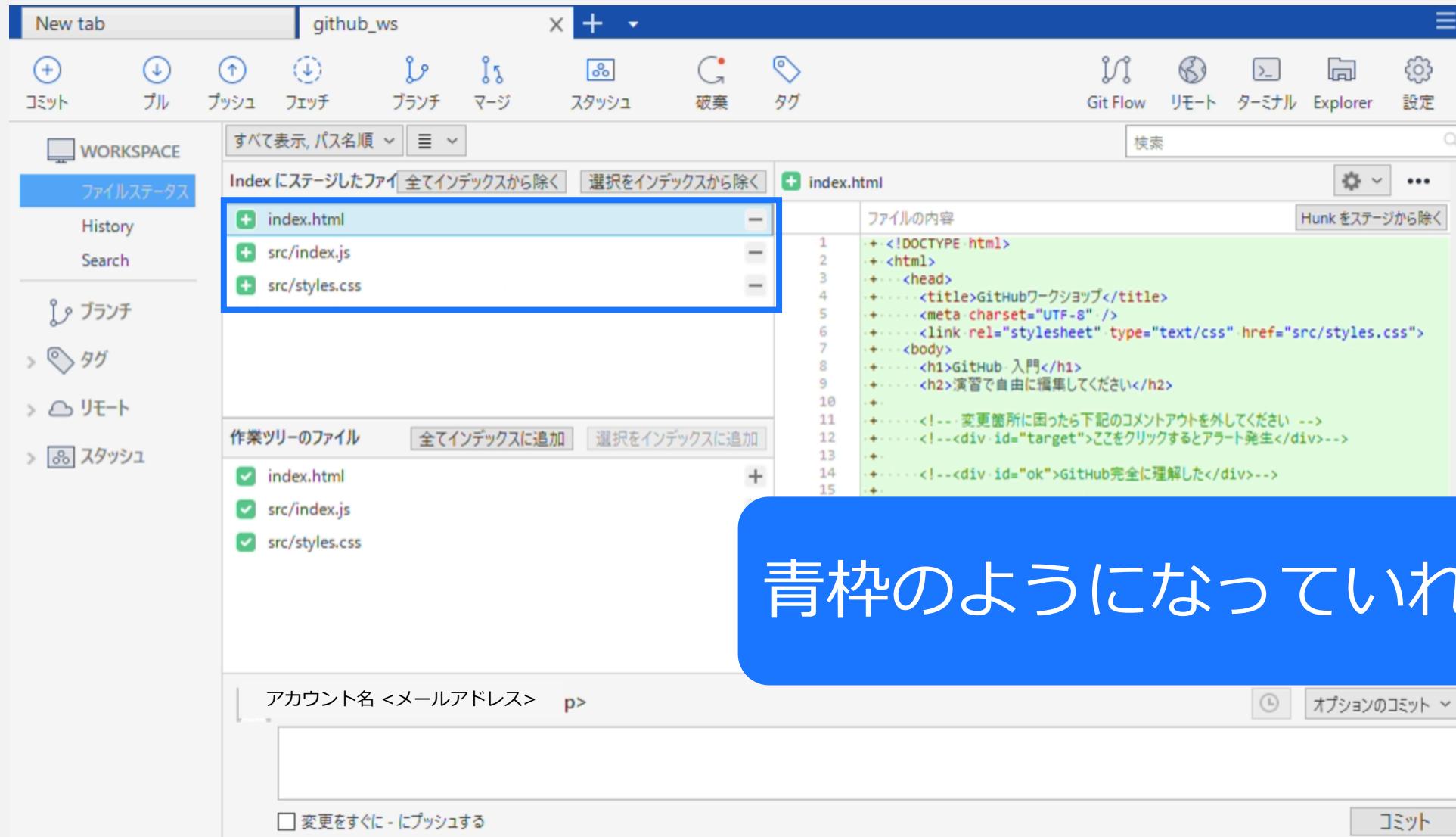


すべてのファイルを

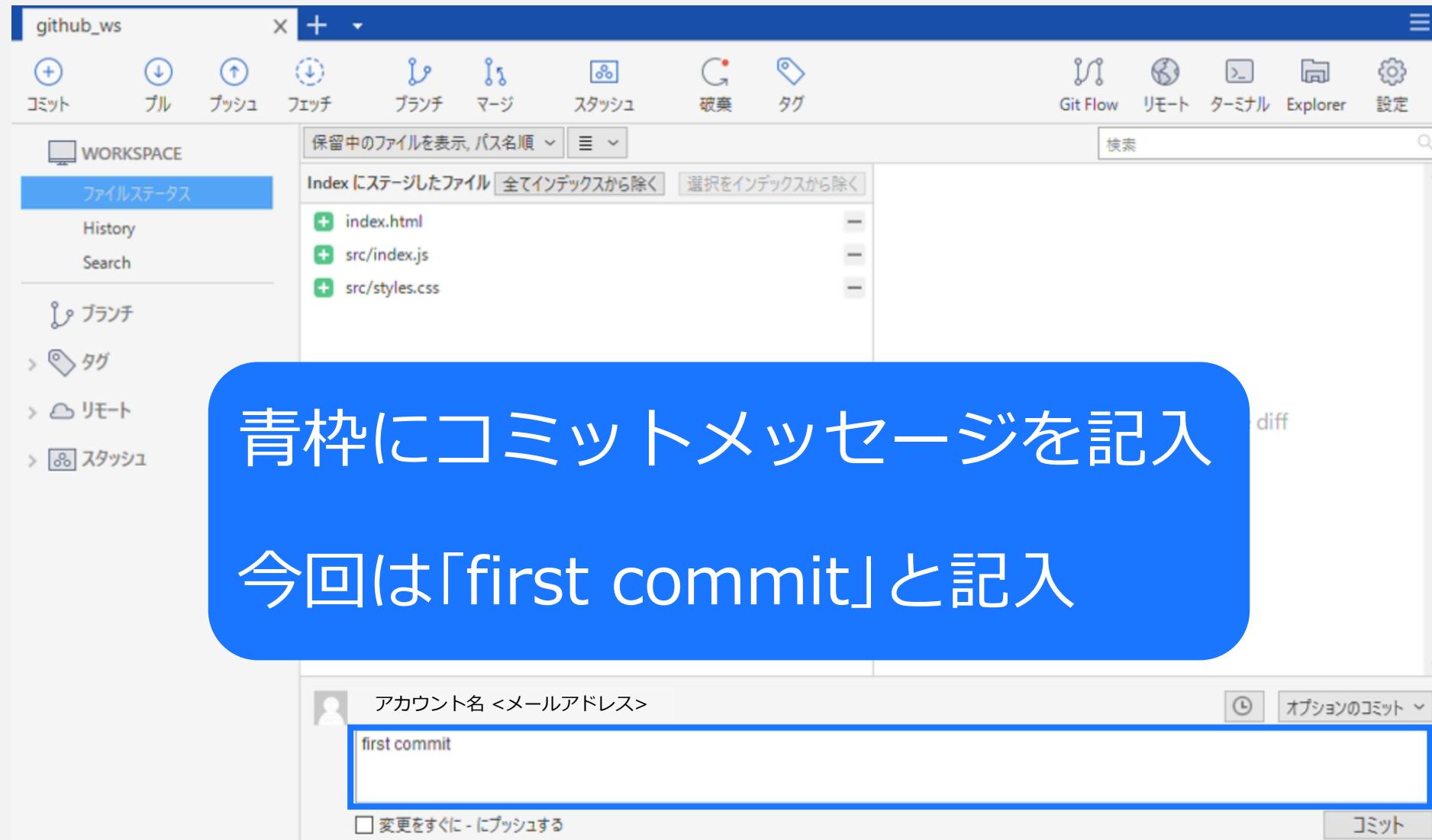
- ・+ボタンをクリックする
- ・上にドラッグ&ドロップする

のいずれかの操作で **add** される

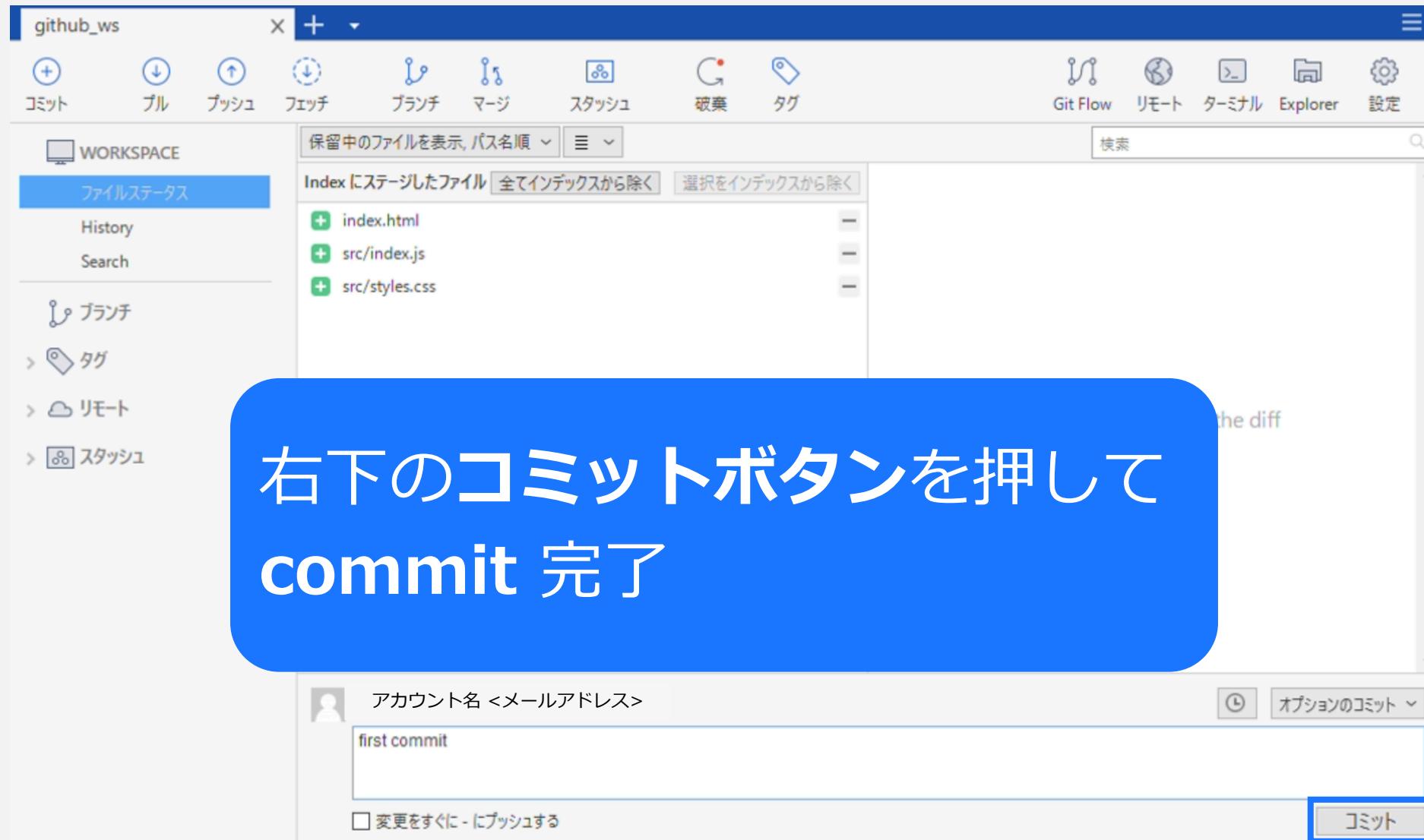
GitHubへの疎通確認 (add)



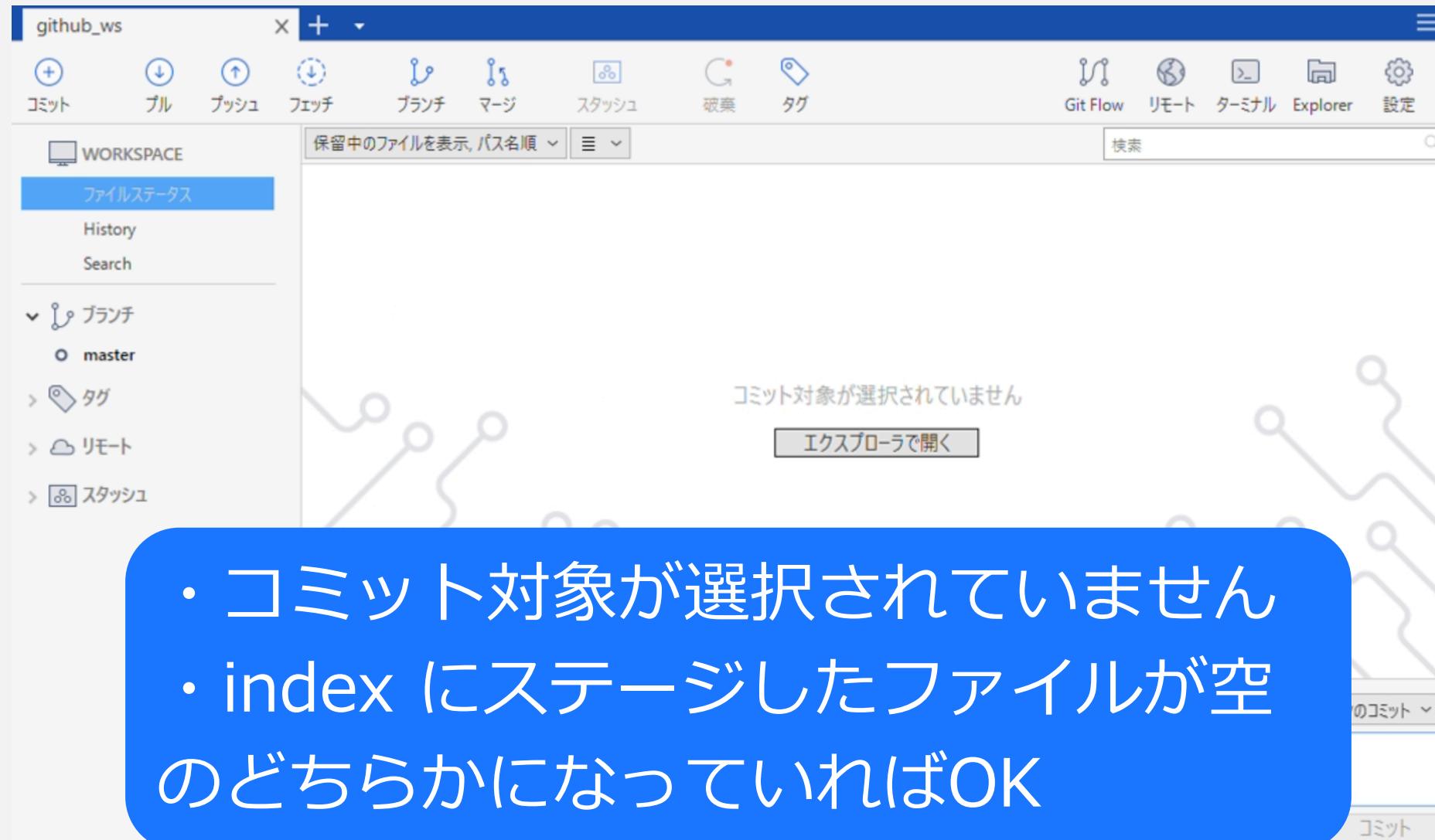
GitHubへの疎通確認 (commit)



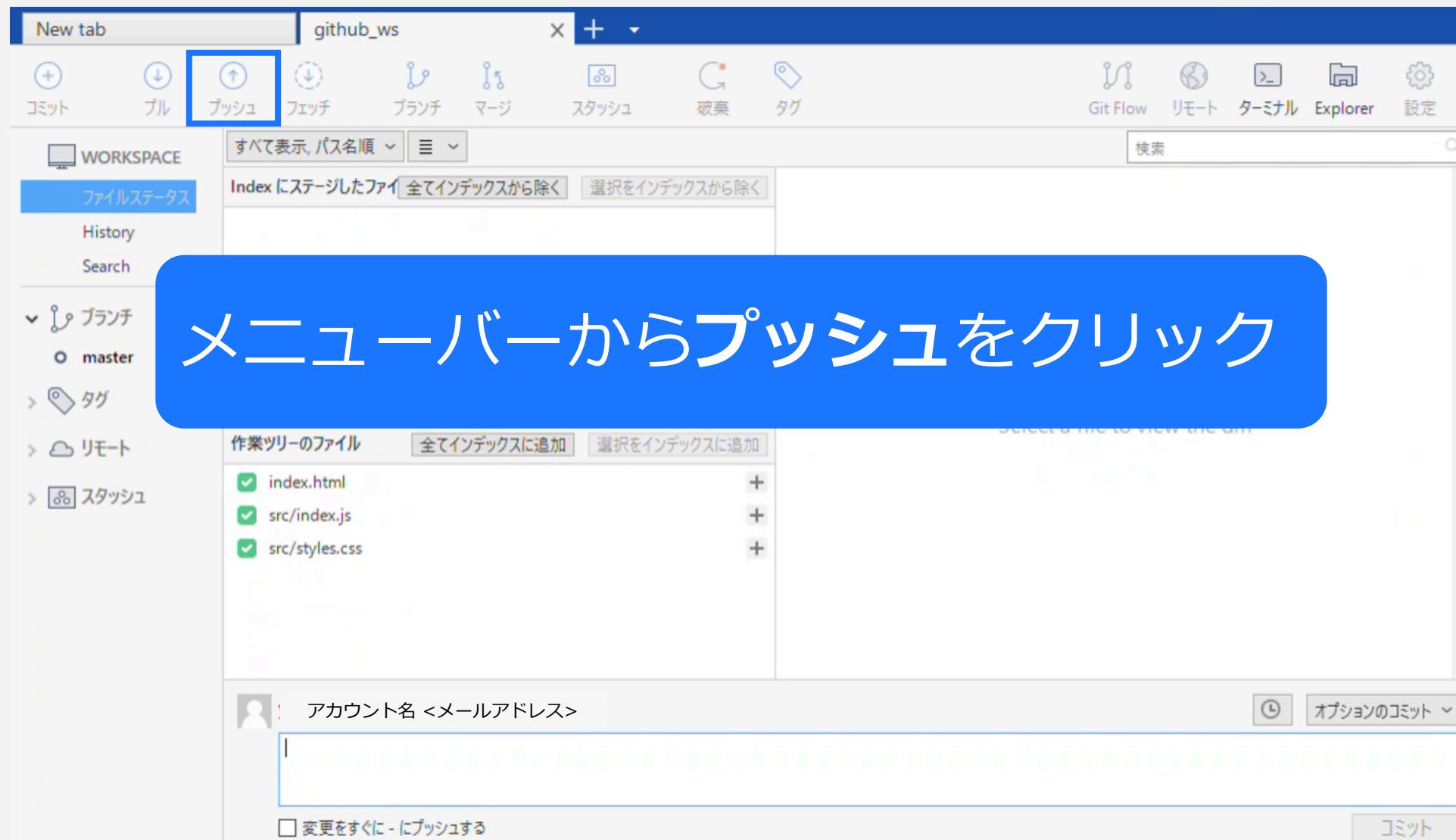
GitHubへの疎通確認 (commit)



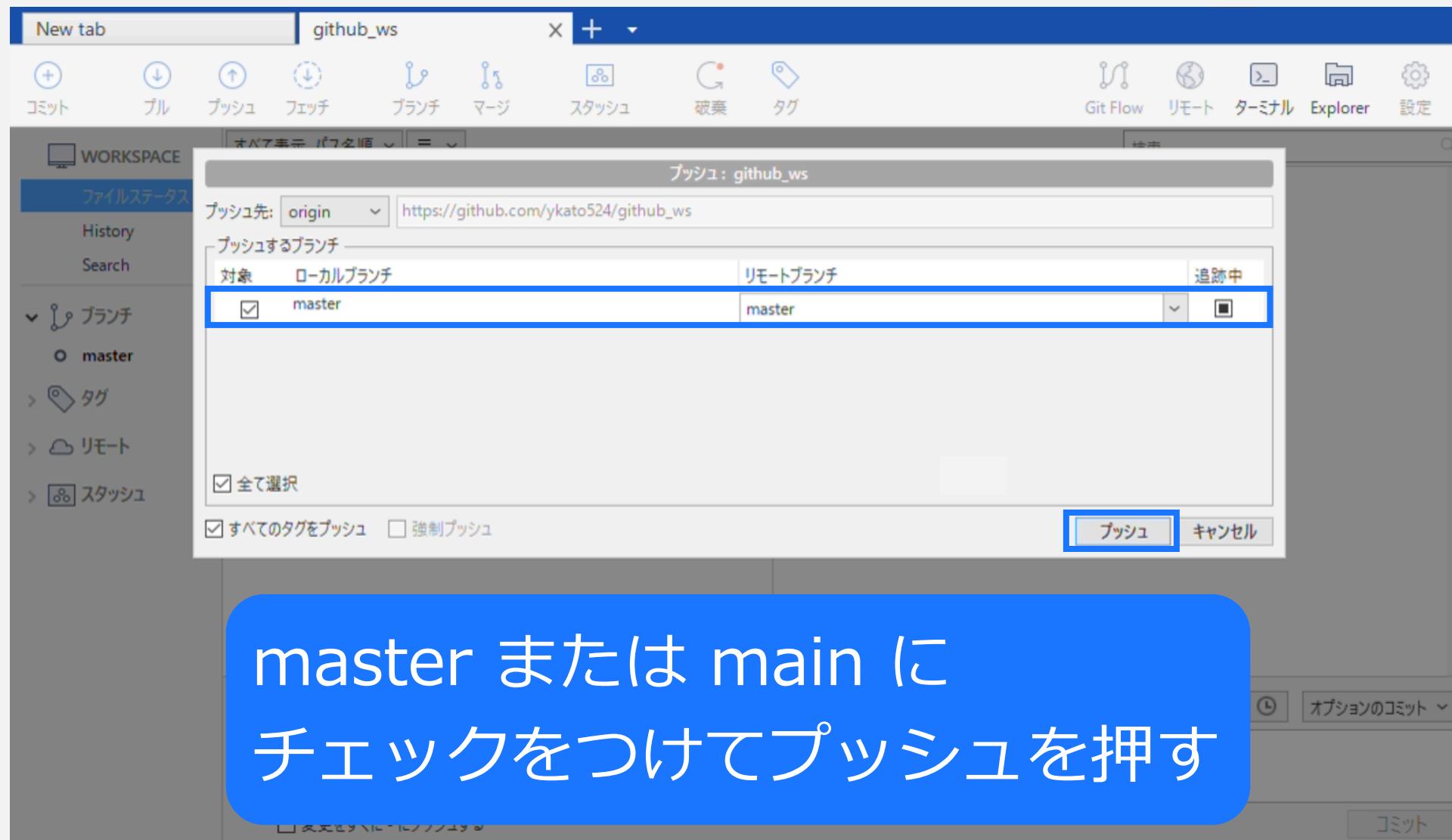
GitHubへの疎通確認 (commit)



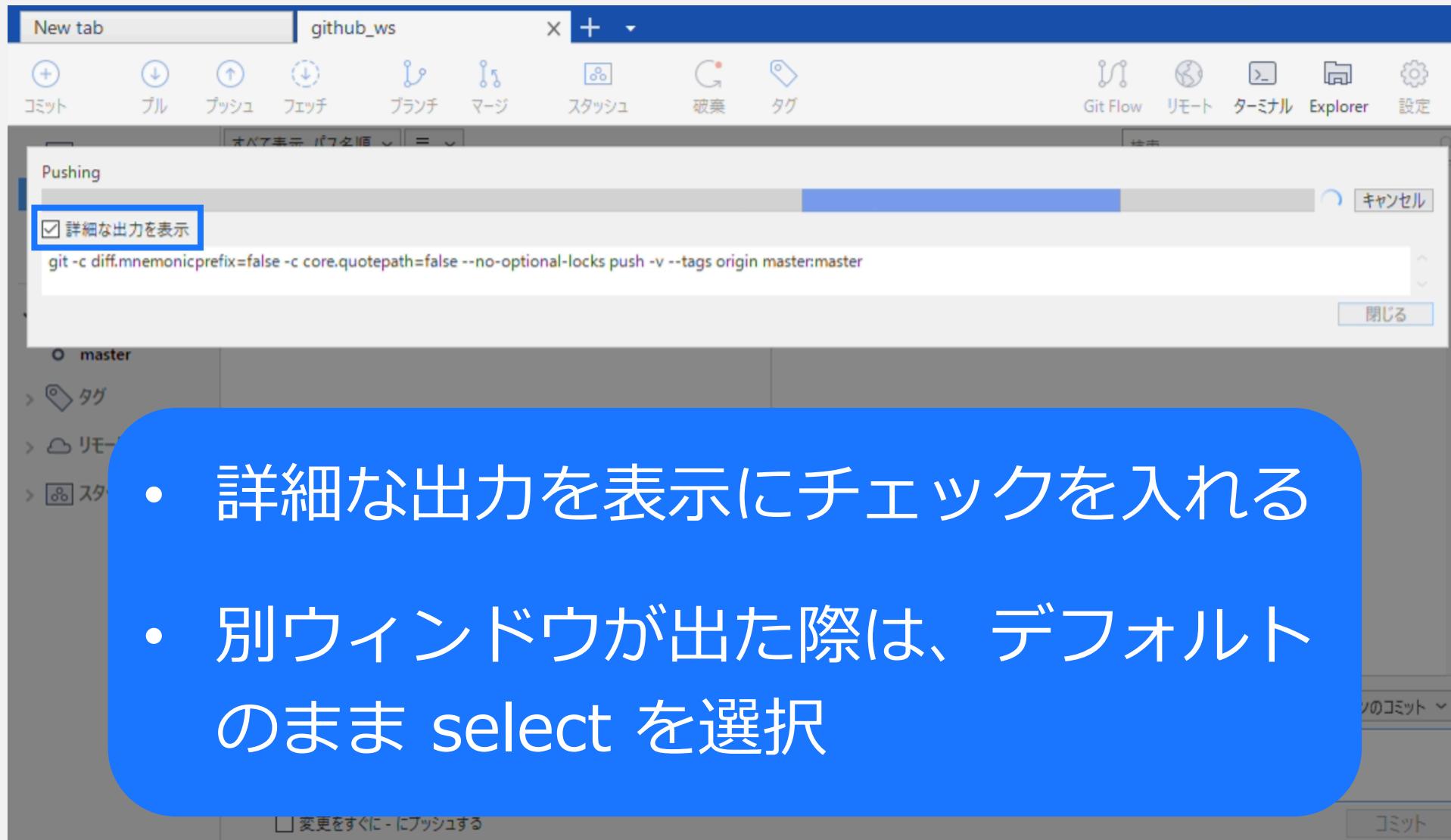
GitHubへの疎通確認 (push)



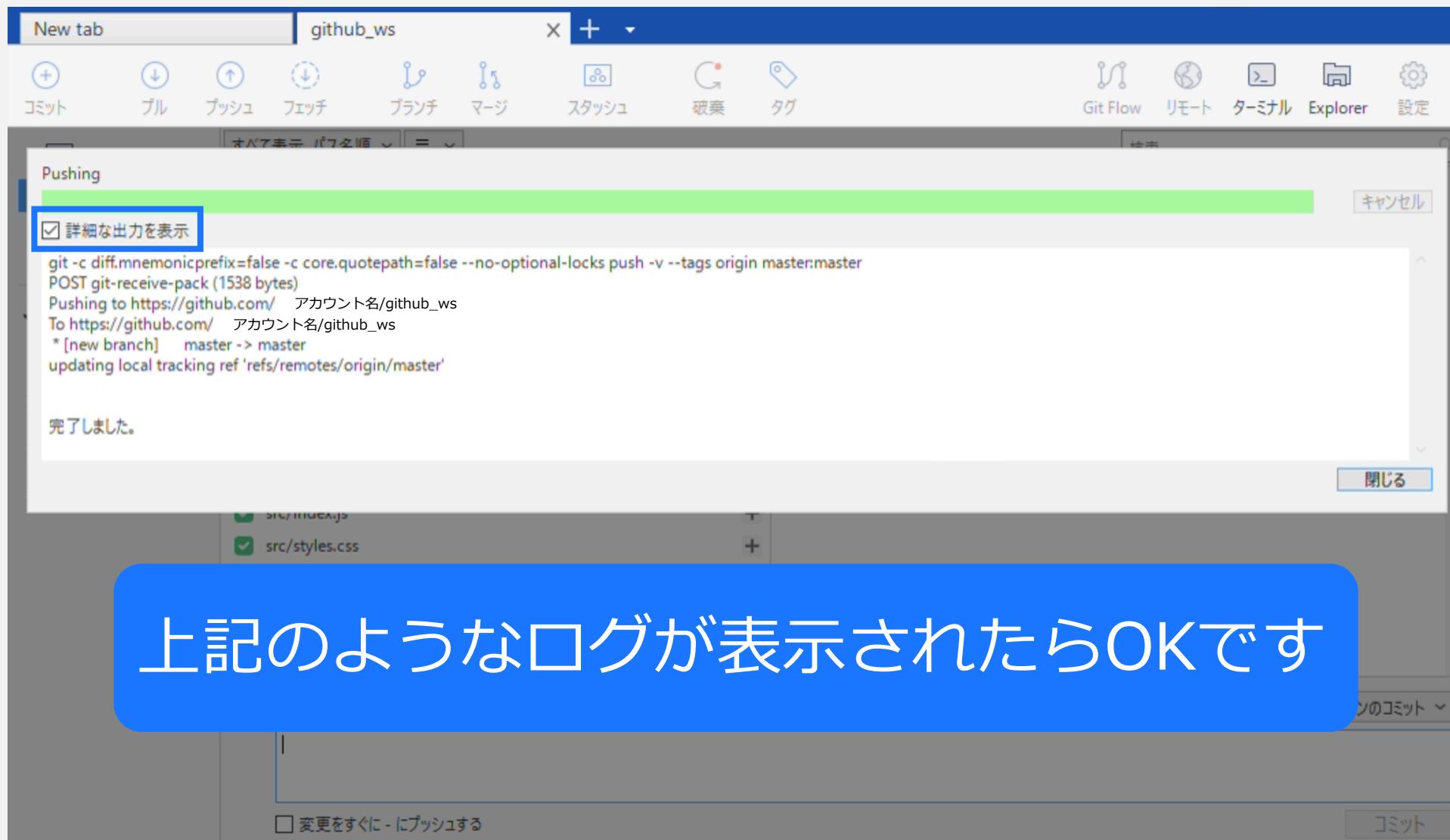
GitHubへの疎通確認 (push)



GitHubへの疎通確認 (push)



GitHubへの疎通確認 (push)



GitHub に反映されているか確認

The screenshot shows a GitHub repository page for the user 'github_ws'. The repository name is 'github_ws'. The page includes a search bar, navigation links for Pull requests, Issues, Marketplace, and Explore, and a notifications icon. Below the header, there are buttons for Unwatch (with 1 notification), Star (0 stars), and Fork (0 forks). The main content area shows the 'Code' tab selected, with 'Issues', 'Pull requests', and 'Activity' tabs also present. A blue callout bubble contains the text '演習ファイルが上がつていればOK'. The commit history shows one commit from 'github_ws' adding 'index.html & src' files. The commit details show '50cdc45 14 minutes ago' and '1 commits'. The commit log lists 'src' and 'index.html' with their respective add actions and timestamps. To the right of the commit history, there is a section for 'No description, website, or topics provided.' and a 'Releases' section indicating 'No releases published' with a link to 'Create a new release'. At the bottom, there is a prompt to 'Add a README'.

演習ファイルが上がつていればOK

Search or jump to... /

Pull requests Issues Marketplace Explore

Unwatch 1 Star 0 Fork 0

Code Issues Pull requests Activity

master 1 branch 0 tags

50cdc45 14 minutes ago 1 commits

src Add index.html & src 14 minutes ago

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

No description, website, or topics provided.

Releases

No releases published Create a new release

Add a README

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

Pushがうまくいかない時②

- Push時のURLの問題

- SourceTreeの[設定] -> [リモート]からoriginのURLを確認し、URLの「github.com」の前が「ユーザー名@」となっていた場合、この「ユーザー名@」を消すとうまくいく場合があります
- 変更後のURL : https://github.com/{自分のユーザー名}/github_ws
- 上記をやってもうまくいかない場合、「Sourcetree と Github を連携」でやったOauth認証の部分をBasic認証に変えて試してみましょう



事前準備は以上となります
お疲れさまでした