

---

# GitHub講座基礎

荒木 寛厚 20200524

---

# 講座の概要

## GitHubとは？

### GitHubとgitの違い

何がどう便利で使われているのか、機能の概要と基本操作を理解する。

個人利用でのファイルのバージョン管理、データの公開方法を理解する。

※そこから先は企業や開発チームによって使い方やファイル管理の運用がこととなりますので、必要に応じて追加の学習をしてください。

---

---

# 学習の環境について

## GigHubアカウント

<https://github.com/>

↓↓取得方法

<https://docs.google.com/presentation/d/1H4YinCVWEyl29hcLmg-qKrl-XCmNrCCq4hldDoJ5zUU/edit?usp=sharing>

## gitが利用できるGUIアプリ

※以下のどれかを利用します。

- Visual Studio Code
  - Soucre Tree
  - Github Desktop など
-

---

# 学習のステップ

---

---

# 学習のステップ

## GitHubとgitとは

GitHubとはgitをオンライン上で管理するサービスです。具体的にどのように利用するのか、確認しましょう。

## リポジトリの作成

GitHub上でリモートリポジトリを作成します。その後ローカルでリポジトリを作成し接続します。

## ファイルのバージョン管理とデータの公開

ローカルで管理しているデータをリモートにプッシュしましょう。

---

---

# Git Hub/gitとは

---

# GitHubとは

**GitHubは、その名の通り、「Git」の「ハブ:拠点・中心・集まり」です。**

GitHubは、Gitの仕組みを利用して、世界中の人々が自分の作品 (プログラムコードやデザインデータなど )を保存、公開することができるようにしたウェブサービスの名称です。

GitHubに作成されたりポジトリ (保存庫のようなもの) は、基本的にすべて公開されますが、有料サービスを利用すると、指定したユーザーからしかアクセスができないプライベートなレポジトリを作ったりすることができます。

参考: <https://www.sejuku.net/blog/7901>



# gitとは

Gitはプログラムのソースコードなどの変更履歴を記録・追跡するための分散型の「バージョン管理システム」です。

- ファイルの変更履歴が管理できる
- 過去のファイルに戻せる
- チームで共有できる

参考: <https://www.sejuku.net/blog/5756>





# gitとは

Gitはプログラムのソースコードなどの変更履歴を記録・追跡するための分散型の「バージョン管理システム」です。

- ファイルの変更履歴が管理できる
- 過去のファイルに戻せる
- チームで共有できる



ファイルのバックアップを大量に  
持ったり、  
過去の変更箇所を誰かに説明し  
たりする必要がなくなる。

参考: <https://www.sejuku.net/blog/5756>

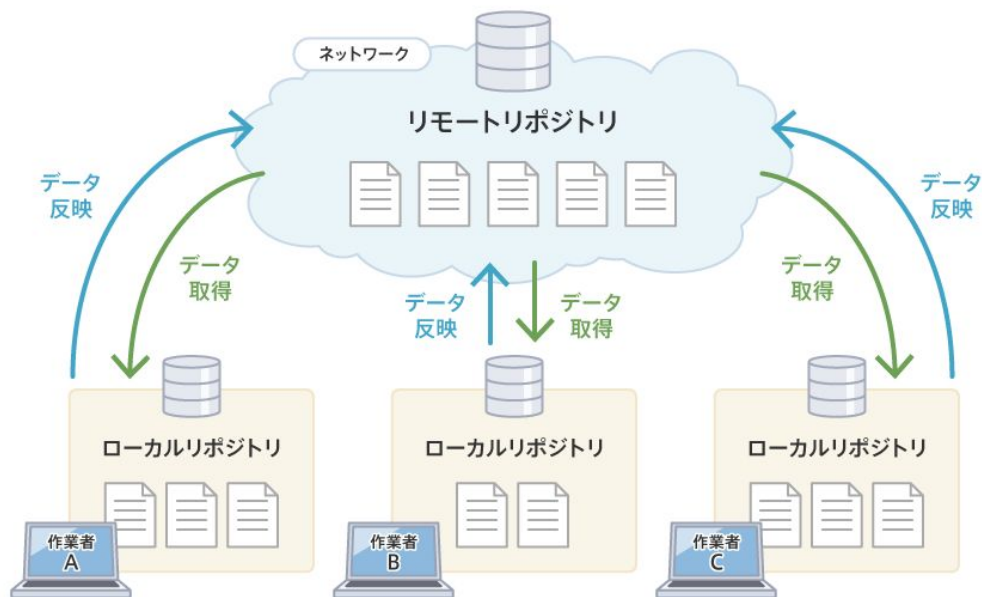


---

# リポジトリの関係

---

# ローカルリポジトリとリモートリポジトリ

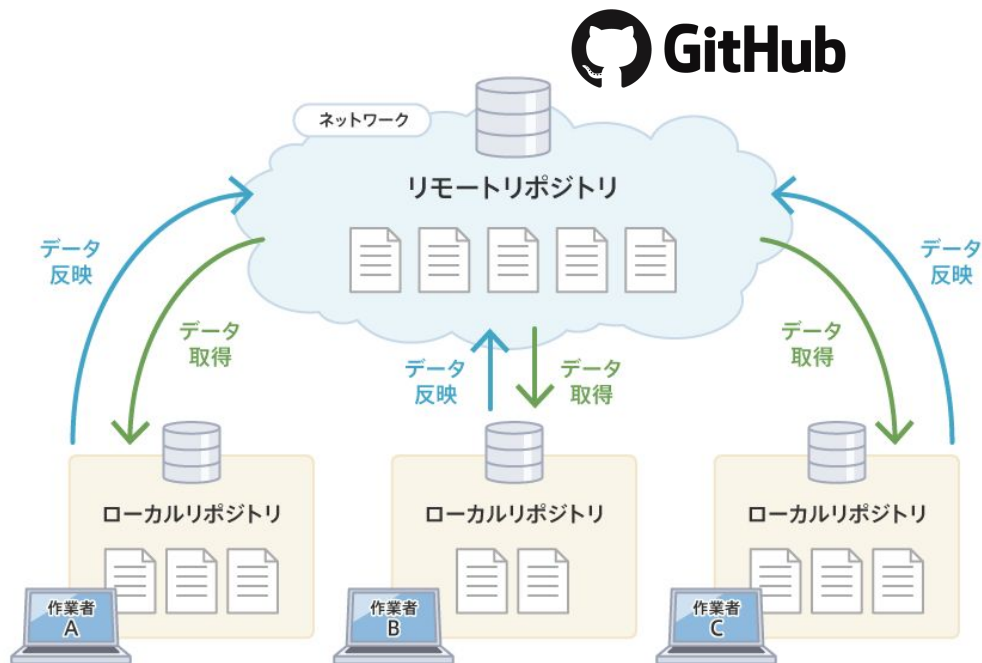


リポジトリとは、ファイルやディレクトリの状態を保存する場所です。

参考

: <https://techacademy.jp/magazine/6235>

# ローカルリポジトリとリモートリポジトリ



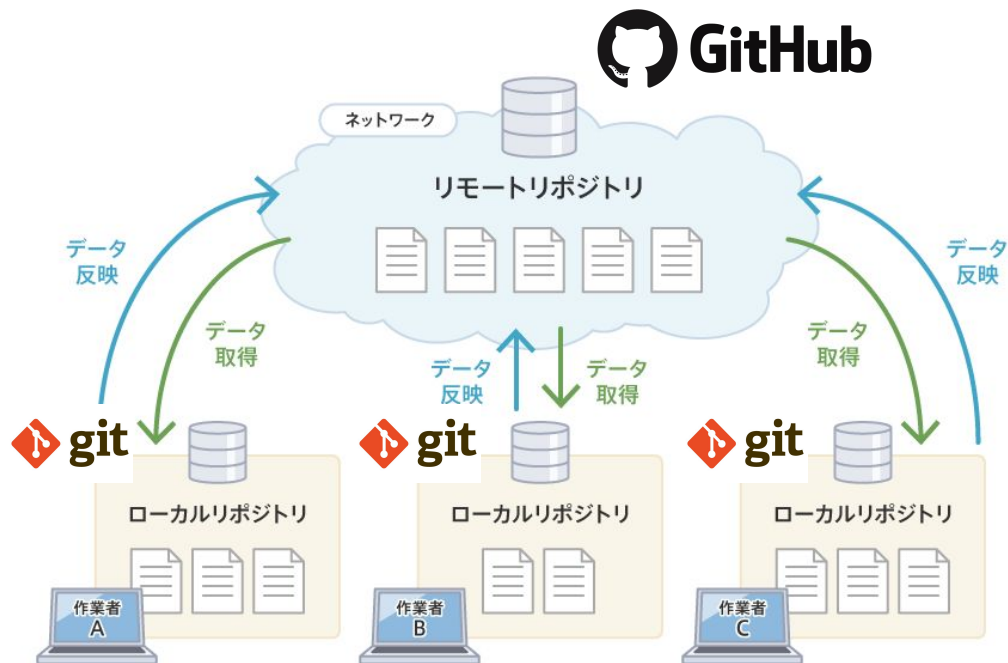
リポジトリは自分のマシン内にある「ローカルリポジトリ」とサーバなどネットワーク上にある「リモートリポジトリ」の2箇所にあります。

- ・リモート: GitHub
- ・ローカル: 自分のマシン

参考

: <https://techacademy.jp/magazine/6235>

# ローカルリポジトリとリモートリポジトリ



基本的にローカルリポジトリで作業を行い、その作業内容をリモートポジトリへプッシュする流れで行います。

データの取得や反映、自分のマシン上でのファイルの変更履歴の管理は gitを利用して行われます。

参考

: <https://techacademy.jp/magazine/6235>

---

# リポジトリの作成

---

# リモートリポジトリの作成 (GitHub)

The screenshot shows the GitHub homepage. At the top is a dark navigation bar with the GitHub logo, a search bar, and links for Pull requests, Issues, Marketplace, and Explore. On the left, there are sections for 'Create your first project' and 'Working with a team?'. In the 'Create your first project' section, the 'Create repository' button is highlighted with a red rectangular box. Below it is a link for 'Import repository'. In the 'Working with a team?' section, there is a link for 'Create an organization'. The main content area features a large light blue box with the text 'Learn Git and GitHub without any code!' and a green 'Read the guide' button. Below this is a 'Start a project' button. At the bottom of the main area, there is a section for 'Discover interesting projects and people to populate your personal news feed.' On the right side, there is a 'News from GitHub Satellite' notification and a list of repositories under the heading 'Explore repositories', including 'neherlab/covid19\_scenarios', 'auth0/auth0-spa-js', and 'bootstrap-vue/bootstrap-vue'.

Create repositoryボタンをクリックし、リモートリポジトリを作成しましょう。

※過去にリポジトリの制作をしている場合は、newボタンをクリックし、新規で作成してください。

# リモートリポジトリの作成

Pull requests Issues Marketplace Explore

## Create a new repository

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

Owner: Attsu-gmail

Repository name: lesson01\_test

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

Description (optional): My first repository

☒ Public  
Anyone 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

Repositoryは複数持つ事ができます。  
Repository nameを決めて入力しましょう。

Description (リポジトリの説明)は任意で入力してください。

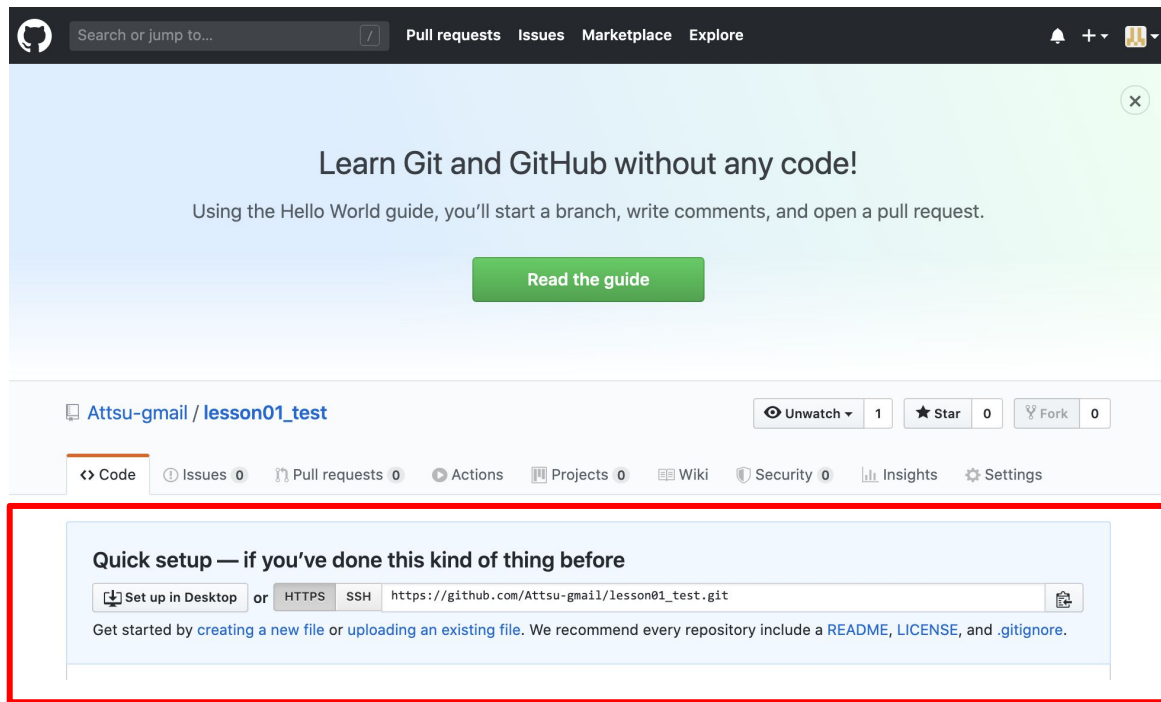
公開設定でPublicをチェックしたまま、  
Create Repositoryをクリックして完了です。

注意: Initialize this repository with a READMEの  
チェックを入れないようにしましょう。  
この後のローカルリポジトリで用意したファイルをアップ  
する作業ができなくなります。

リモートリポジトリのデータと履歴をローカルに構築する  
場合「クローン」はチェックを入れても大丈夫です。



# リモートリポジトリの作成

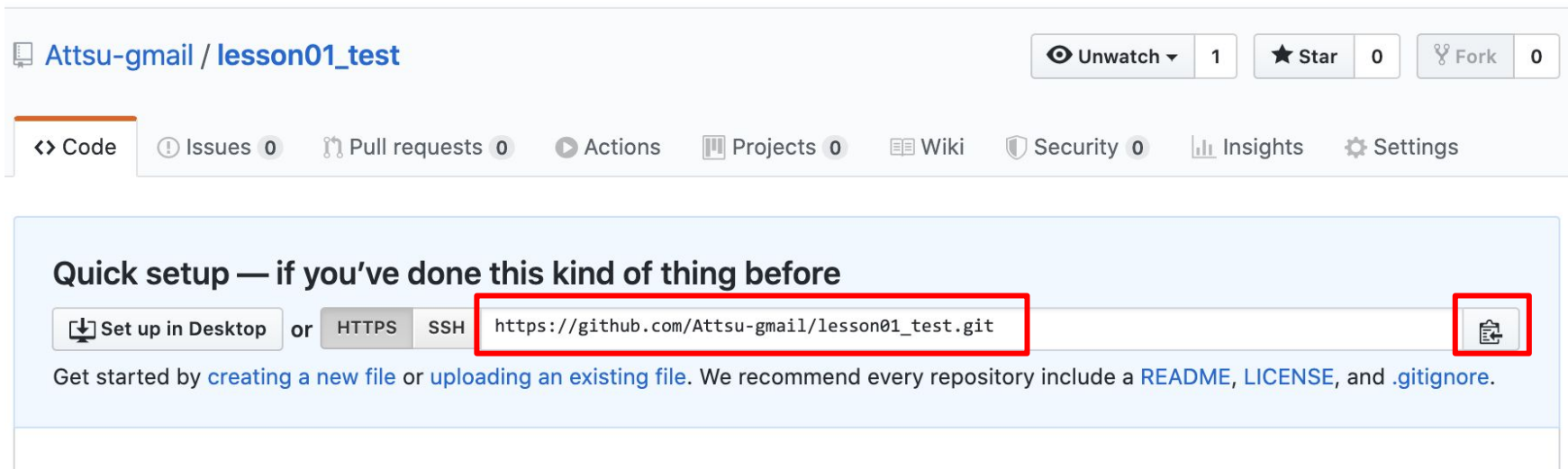


リモートリポジトリが作られるとこのような画面が表示されます。

この時点ではリモートリポジトリで管理しているファイルは何もありません。

画面下部に記載されている接続用のURLを利用して、ローカルリポジトリを作成します。

# ローカルリポジトリの作成



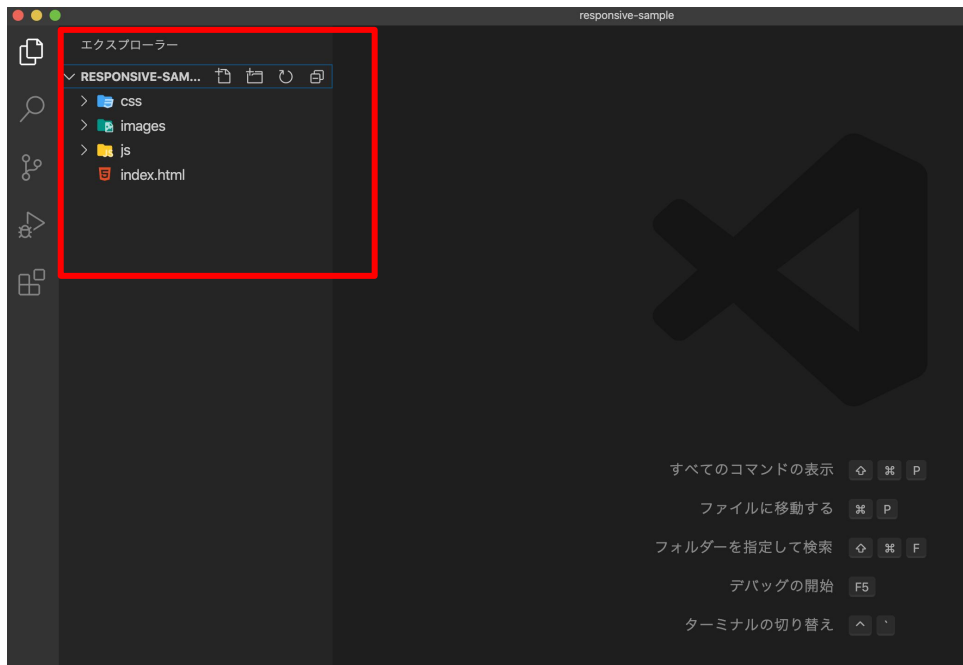
The screenshot shows the GitHub interface for a repository named 'lesson01\_test' by the user 'Attsu-gmail'. At the top, there are buttons for 'Unwatch' (1), 'Star' (0), and 'Fork' (0). Below this is a navigation bar with tabs for 'Code', 'Issues' (0), 'Pull requests' (0), 'Actions', 'Projects' (0), 'Wiki', 'Security' (0), 'Insights', and 'Settings'. The main content area is titled 'Quick setup — if you've done this kind of thing before'. It features three options: 'Set up in Desktop', 'HTTPS', and 'SSH'. The 'SSH' option is selected, and its corresponding URL, 'https://github.com/Attsu-gmail/lesson01\_test.git', is displayed in a text box. A red rectangle highlights this URL. To the right of the text box is a copy icon, also highlighted with a red rectangle. Below the text box, there is a line of text: 'Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).'

リモートリポジトリの接続情報が必要になります。

予め**URLをコピー**しておきましょう。

接続方法いろいろありますが、今回は HTTPSのURLをコピーしてください。

# ローカルリポジトリの作成 (VsCode)

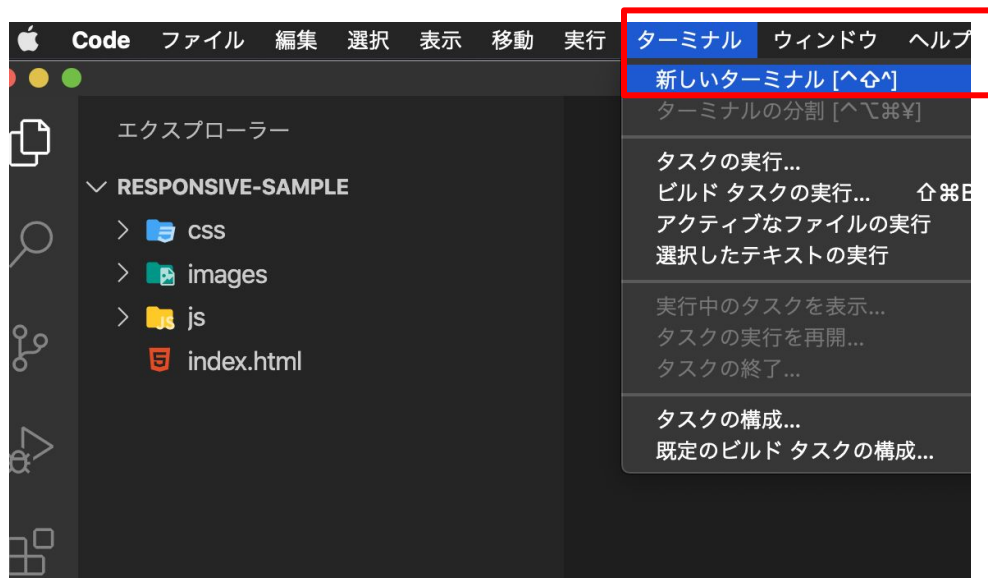


ローカルで管理したいファイルを VsCode でフォルダごと開きます。

**ショートカット: Ctrl+O (開く)**

今回は最終課題のフォルダを開き、ローカルリポジトリとして管理します。

# ローカルリポジトリの作成 (VsCode)



ターミナルを使って、gitが利用できるよう環境設定を行います。

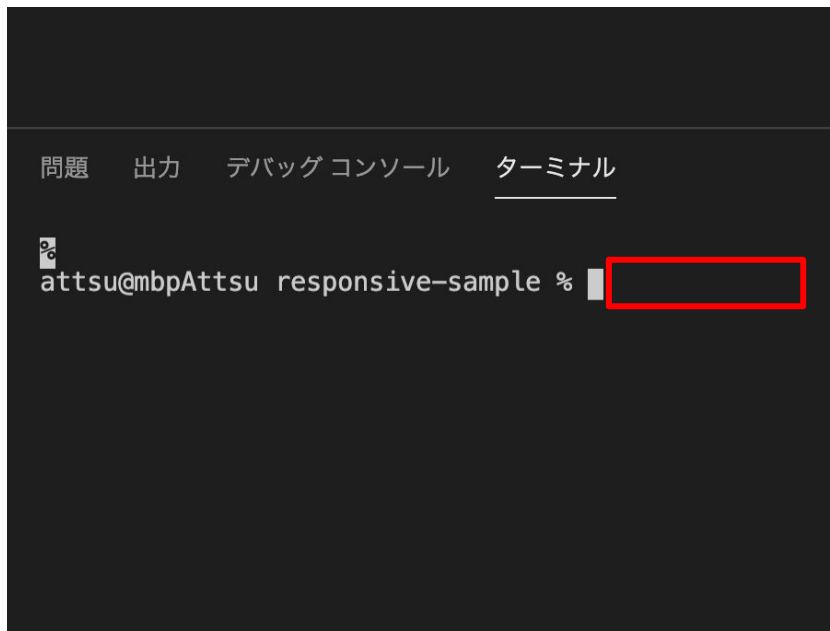
VsCodeのメニュー

**ターミナル > 新しいターミナル**

※英語表記の場合

"Terminal - > New Terminal"

# ローカルリポジトリの作成 (VsCode)



## Gitの初期化

※この内容は一回やれば OK

ターミナルで以下のコマンドを入力

**git --version** gitがそもそも利用可能かどうか

**git config --global user.name "各自の登録名"**

**git config --global user.email 各自の登録メール**

**git config -l ※登録内容の確認**

※windowsの方は別途、gitのインストールが必要な場合があります。こちらを [参照](#) してください

# ローカルリポジトリの作成 (VsCode)



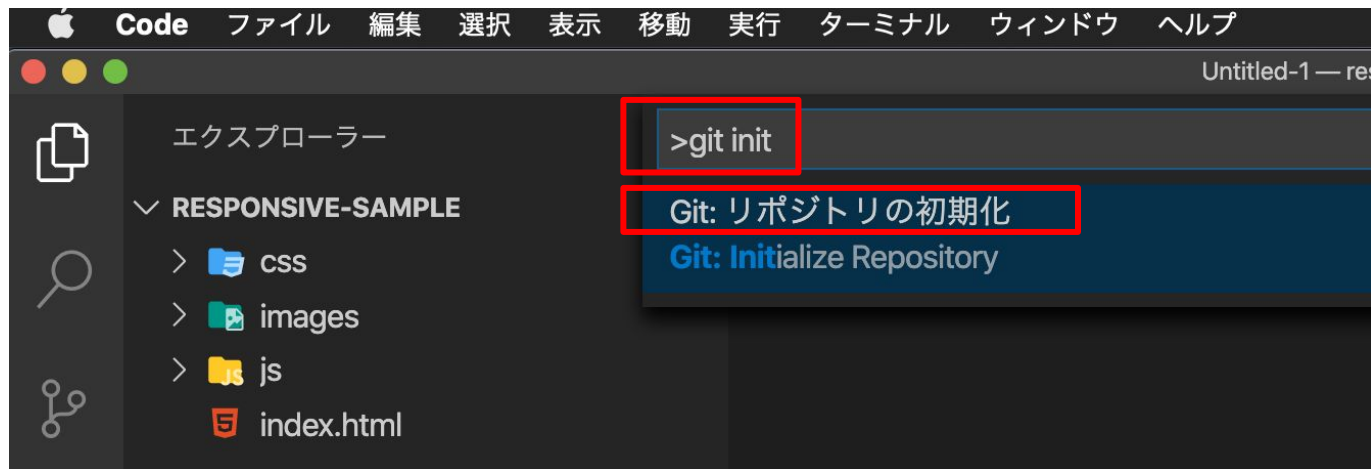
コマンドパレットを使って、ローカルリポジトリの設定を行います。

VsCodeのメニュー

**表示 > コマンドパレット**

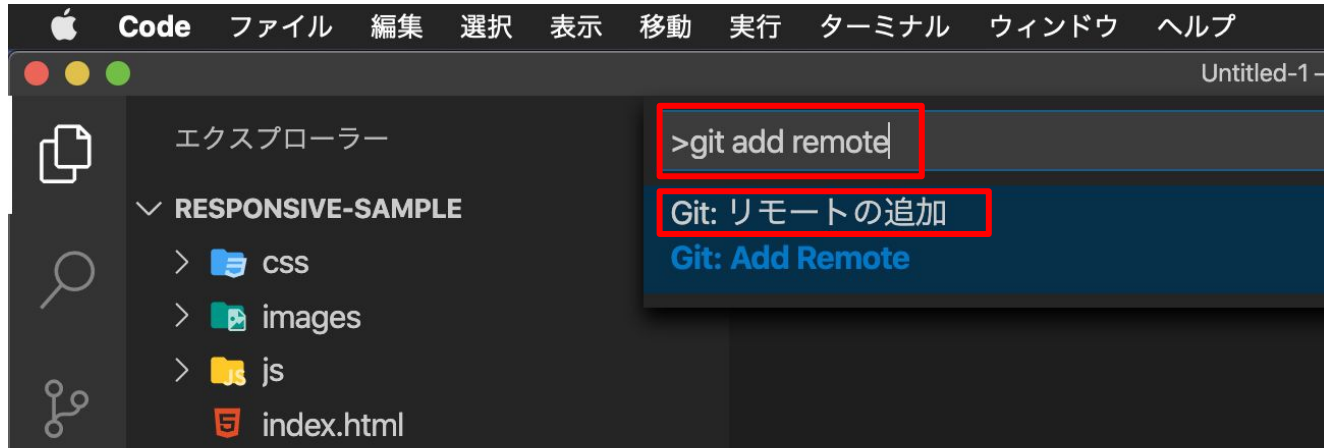
※ショートカット: Shift + Command + P

# ローカルリポジトリの作成 (VsCode)



VsCodeで開いているフォルダをローカルリポジトリとして登録します。  
コマンドパレットに"**git init**"と入力し、  
入力欄の下に候補として表示される「**リポジトリの初期化**」のコマンドを選択

# ローカルリポジトリの作成 (VsCode)



先程、Githubページでコピーしたの接続用の URLを登録します。

コマンドパレットに"**git add remote**"と入力し、入力欄の下に候補として表示される「**リモートの追加**」のコマンドを選択。



# ローカルリポジトリの作成 (VsCode)



```
>origin
```

この後、リモートの名前を求められるので、"**origin**"と入力しEnterを押してしてください。  
**origin**: デフォルトのリポジトリの場所 (URL)の別名



```
>https://github.com/Attsu-gmail/lesson01_test.git
```

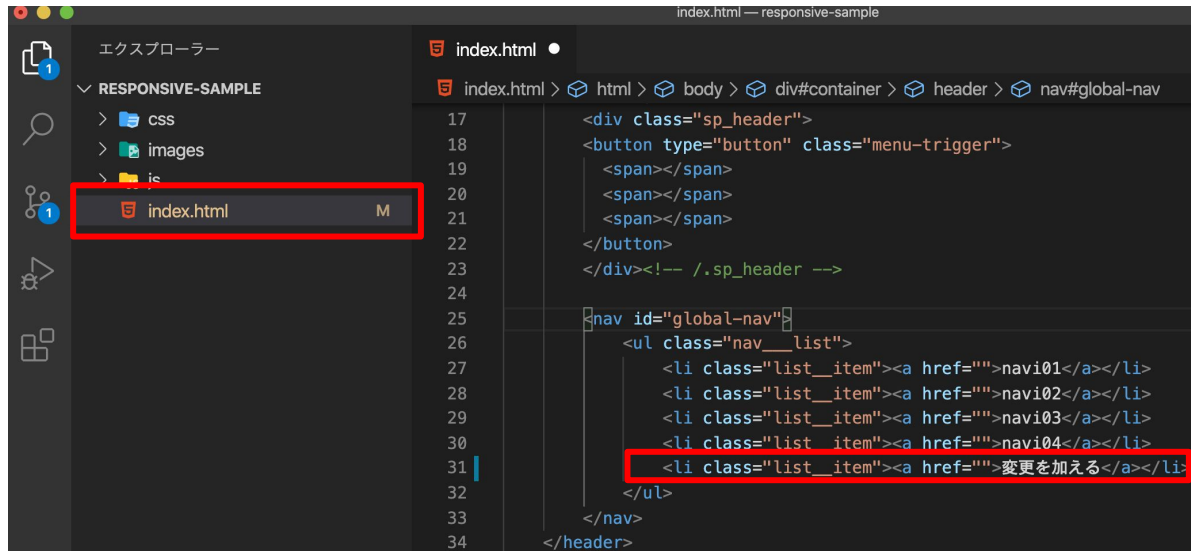
次に、リモートのURLを求められるので、GitHubでコピーした"**URL**"を貼り付けEnterを押してしてください。

---

# ローカルファイルの管理

---

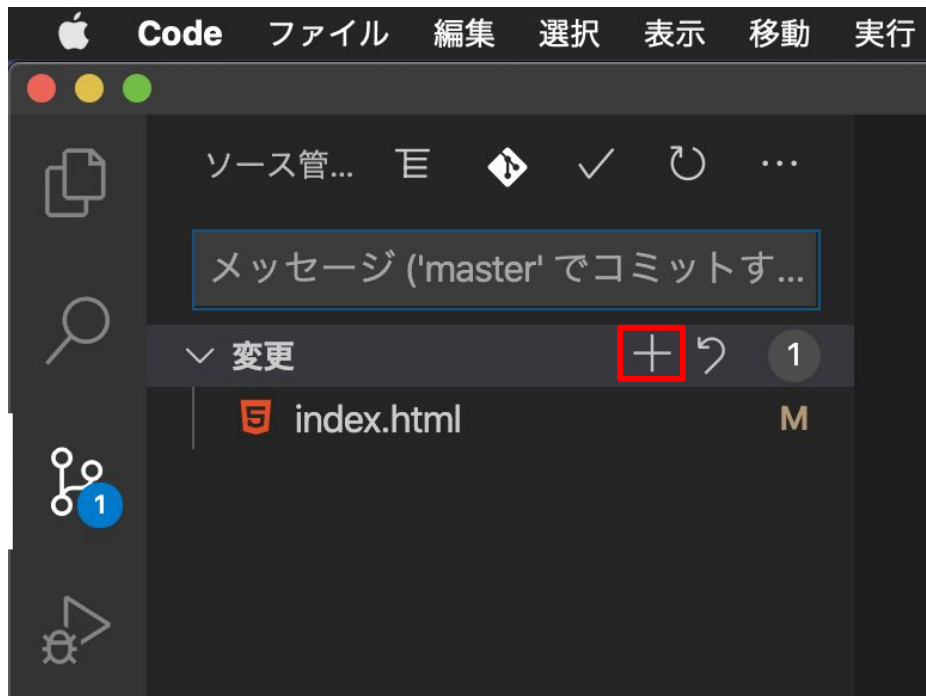
# ファイルの管理(更新からリモートへのアップについて)



ファイルを更新し、gitに更新履歴を登録します。

例えばhtmlファイルを更新して保存をすると、ファイル名に **Mのマーク(modified)**が表示されます。更新されたが、記録がされていないファイルであることが表示から確認できます。

# ファイルの管理 (ステージング、コミット、プッシュについて)



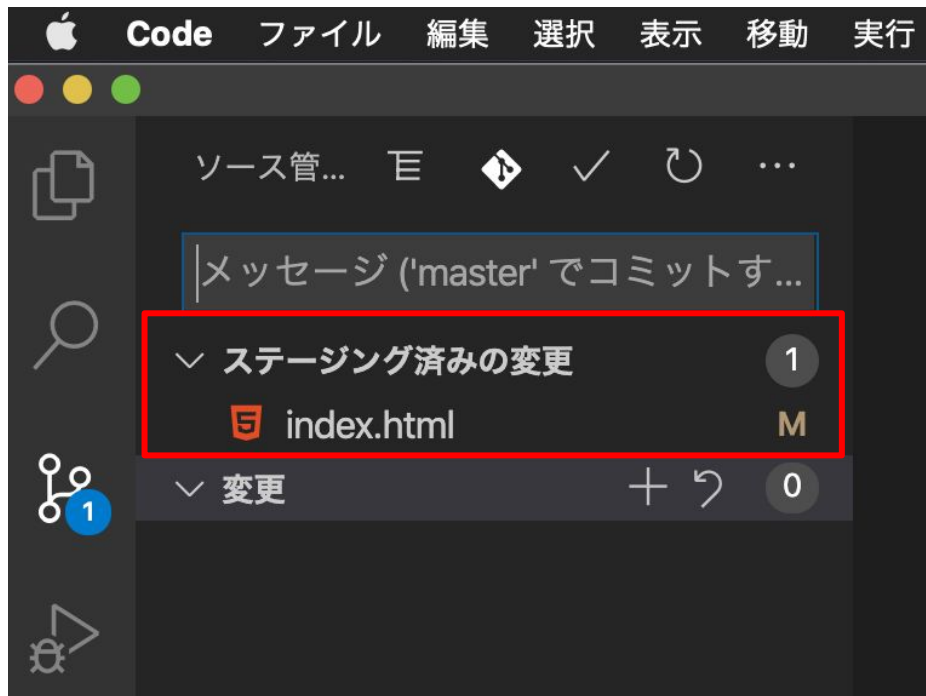
Gitに更新履歴を登録する対象のファイルを選定します。

更新の対象になっているファイルが、変更のタブの中に一覧表示されます。  
**+マーク**を押すと対象として選択が完了します。

**Mのマーク(modified)**が付いたファイルや新しく追加されたファイルなど、記録の対象とすることが選択することができます。

この記録対象として選択された状態にあることを**ステージング**と呼びます。

# ファイルの管理 (ステージング、コミット、プッシュについて)

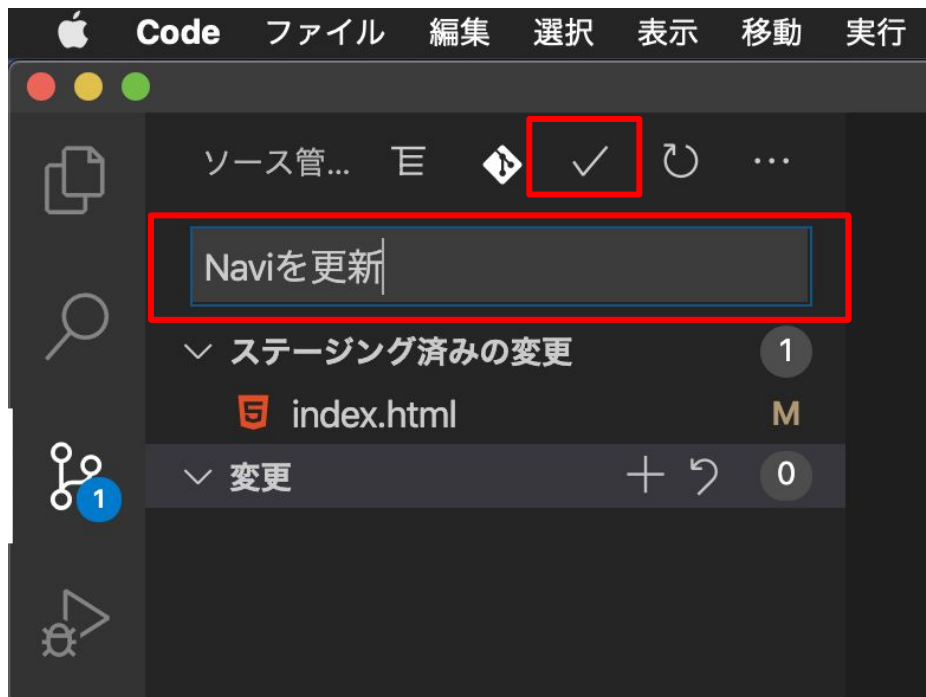


「**ステージング済みの変更**」に並んでいるファイルをgitに記録することができます。

画面上部にある **✓ チェックマーク** をクリックすると記録されますが、必ず記録する際のメモを残しましょう。

この記録する作業のことを **コミット** と呼びます。

# ファイルの管理(ステージング、コミット、プッシュについて)

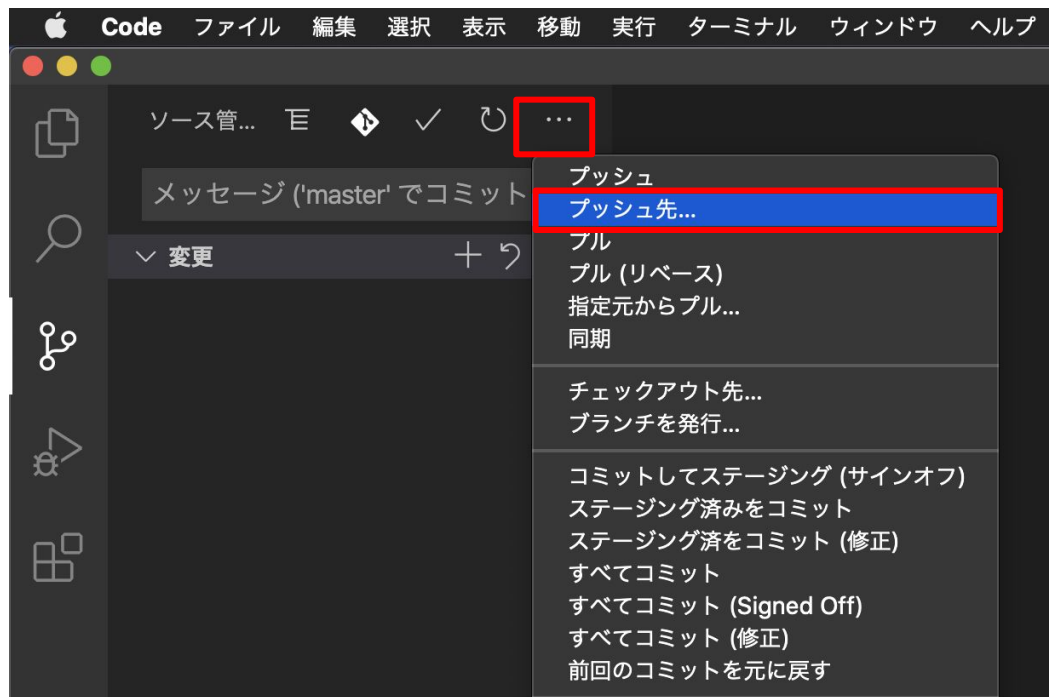


「**ステージング済みの変更**」に並んでいるファイルをgitに記録することができます。

画面上部にある **✓チェックマーク** をクリックすると記録されますが、必ず記録する際のメモを残しましょう。

この記録する作業のことを **コミット** と呼びます。

# ファイルの管理 (ステージング、コミット、プッシュについて)



「...」のマークをクリックすると。  
gitの操作に関するメニューが表示されます。

メニュー上部にある「**プッシュ先**」をクリックし、リモートリポジトリへ更新履歴とともに更新データをアップしましょう。

ローカルデータからリモートへのアップの操作をgitでは**プッシュ**と呼びます。

# ファイルの管理 (ステージング、コミット、プッシュについて)



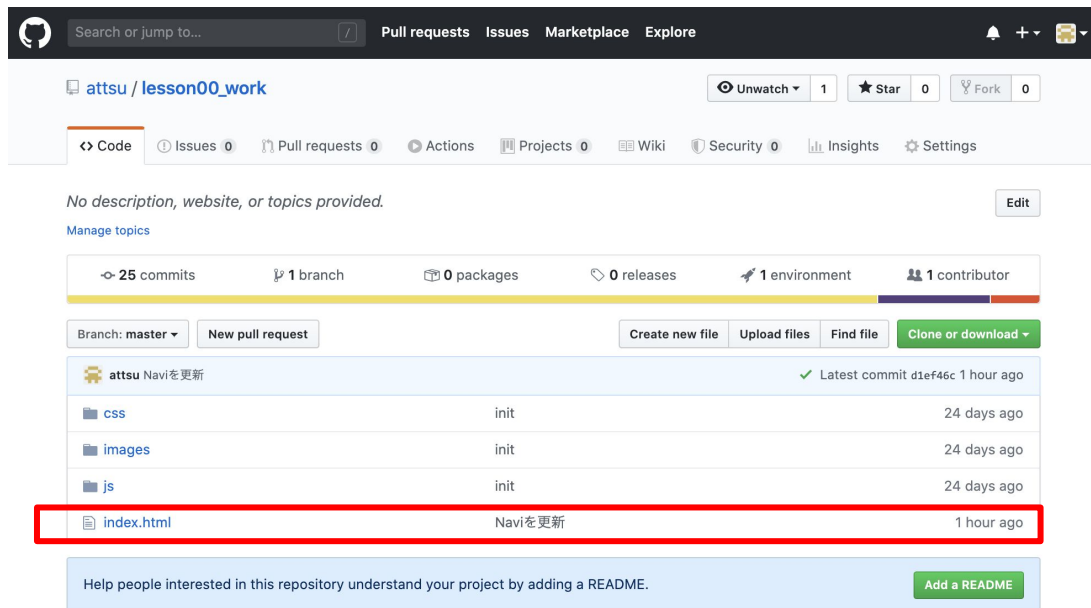
プッシュ先として”**origin**”を選択

※初回のみ、ユーザー名とパスワードが求められます。

※パスワードの入力では文字が非表示となるが、入力続けて Enterキーを押してください。



# ファイルの管理 (ステージング、コミット、プッシュについて)



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

attsu / lesson00\_work

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security 0 Insights Settings

No description, website, or topics provided. Edit

Manage topics

25 commits 1 branch 0 packages 0 releases 1 environment 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

attsu Naviを更新 Latest commit d1ef46c 1 hour ago

css	init	24 days ago
images	init	24 days ago
js	init	24 days ago
index.html	Naviを更新	1 hour ago

Help people interested in this repository understand your project by adding a README. Add a README

ファイルがアップされ、リモートリポジトリ上の表示と履歴が更新されました。

---

# コミットの履歴について

---

# コミットの履歴について (GitHub)

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

attsu / lesson00\_work

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security 0 Insights Settings

No description, website, or topics provided. Edit

Manage topics

25 commits 1 branch 0 packages 0 releases 1 environment 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

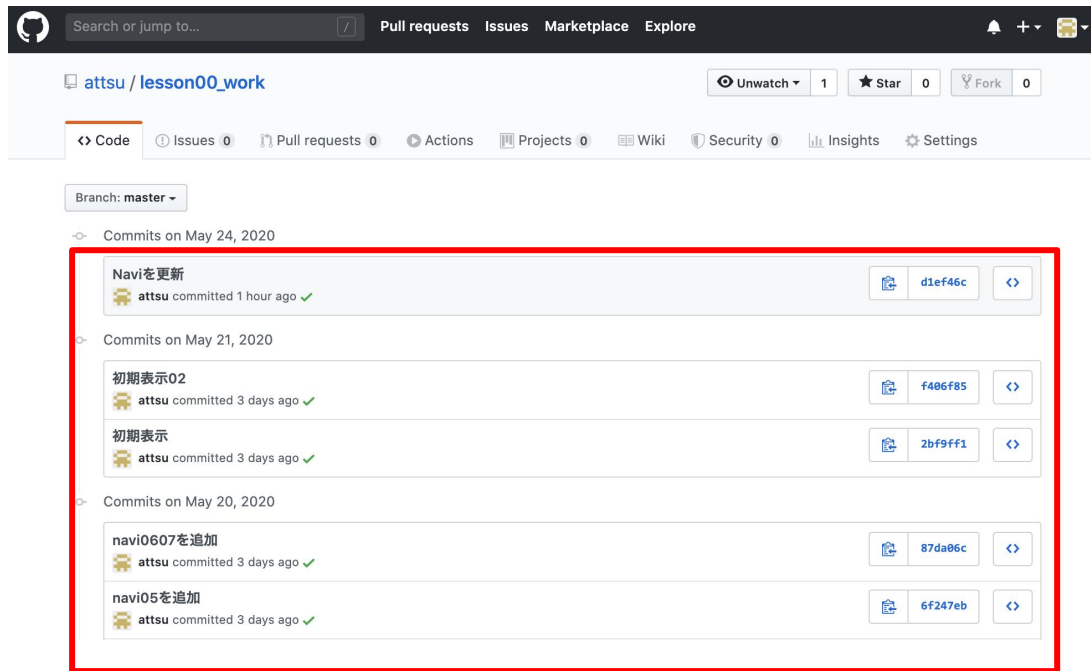
attsu Naviを更新 Latest commit d1ef46c 1 hour ago

css	init	24 days ago
images	init	24 days ago
js	init	24 days ago
index.html	Naviを更新	1 hour ago

Help people interested in this repository understand your project by adding a README. Add a README

Commitsのタブで、これまでのコミットの履歴を確認することができます。

# コミットの履歴について (GitHub)



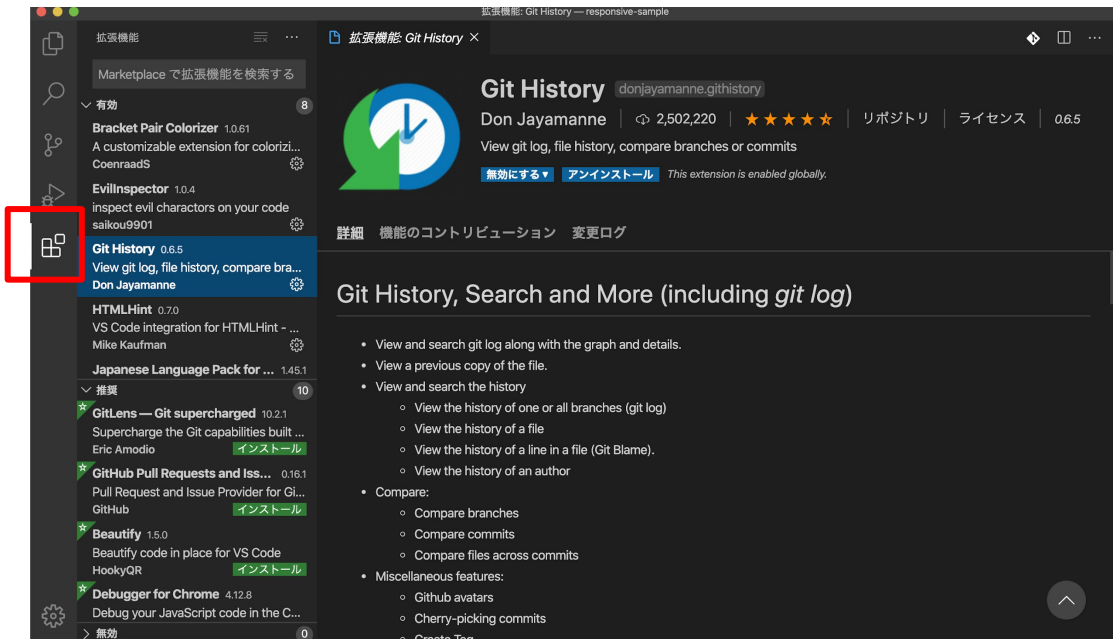
The screenshot shows the GitHub interface for the repository `attsu / lesson00_work`. The page displays the commit history for the `master` branch. A red rectangular box highlights the commit history section, which includes the following commits:

- Commits on May 24, 2020**
  - Naviを更新**  
attsu committed 1 hour ago ✓  
Commit hash: `d1ef46c`
- Commits on May 21, 2020**
  - 初期表示02**  
attsu committed 3 days ago ✓  
Commit hash: `f486f85`
  - 初期表示**  
attsu committed 3 days ago ✓  
Commit hash: `2bf9ff1`
- Commits on May 20, 2020**
  - navi0607を追加**  
attsu committed 3 days ago ✓  
Commit hash: `87da06c`
  - navi05を追加**  
attsu committed 3 days ago ✓  
Commit hash: `6f247eb`

これまでのコミットの履歴を確認することができます。

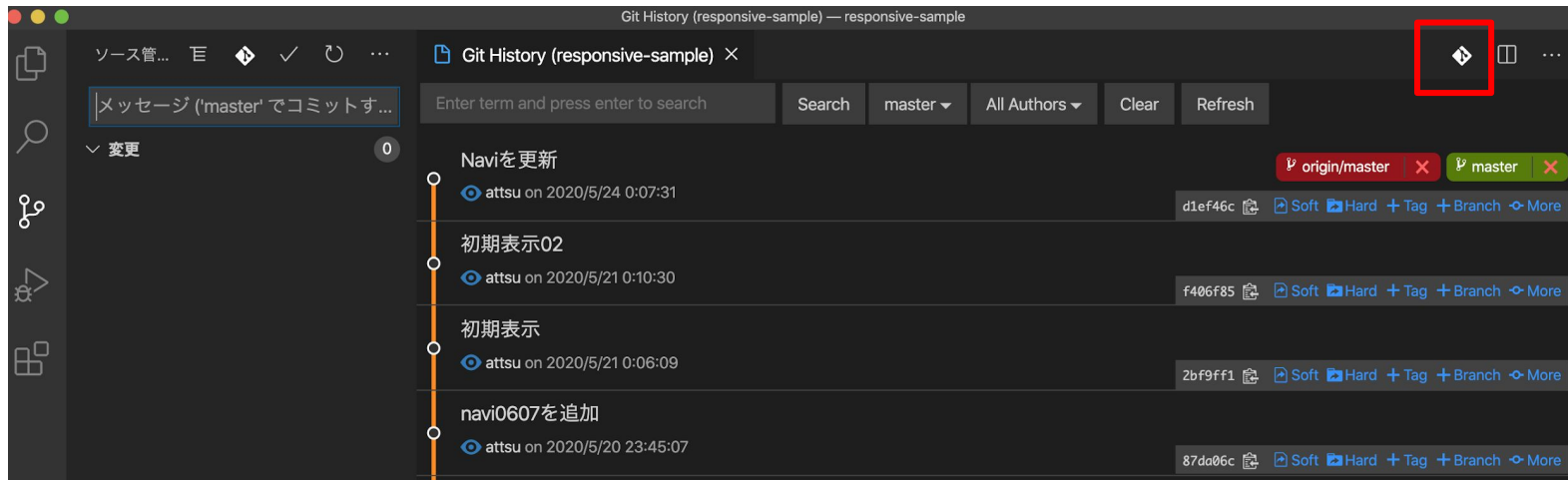
ローカルリポジトリのコミット履歴がリモートにも記録された形になっています。

# コミットの履歴について (VsCode)



ローカルリポジトリのコミット履歴を確認するには、プラグインの Git History を利用すると良いでしょう。

# コミットの履歴について (VsCode)



画面右上のアイコンをクリックすると、これまでのコミット履歴を確認することができます。  
表示メニューのコマンドパレット( $\text{⌘}$  +shift+p)から "**view History**"と入力することでも同様に確認ができます。

---

# Git Hubでサイトを公開する

---

# サイトの公開について (GitHub)

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

attsu / lesson00\_work

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security 0 Insights Settings

No description, website, or topics provided. Edit

Manage topics

25 commits 1 branch 0 packages 0 releases 1 environment 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

attsu Naviを更新 Latest commit d1ef46c 2 hours ago

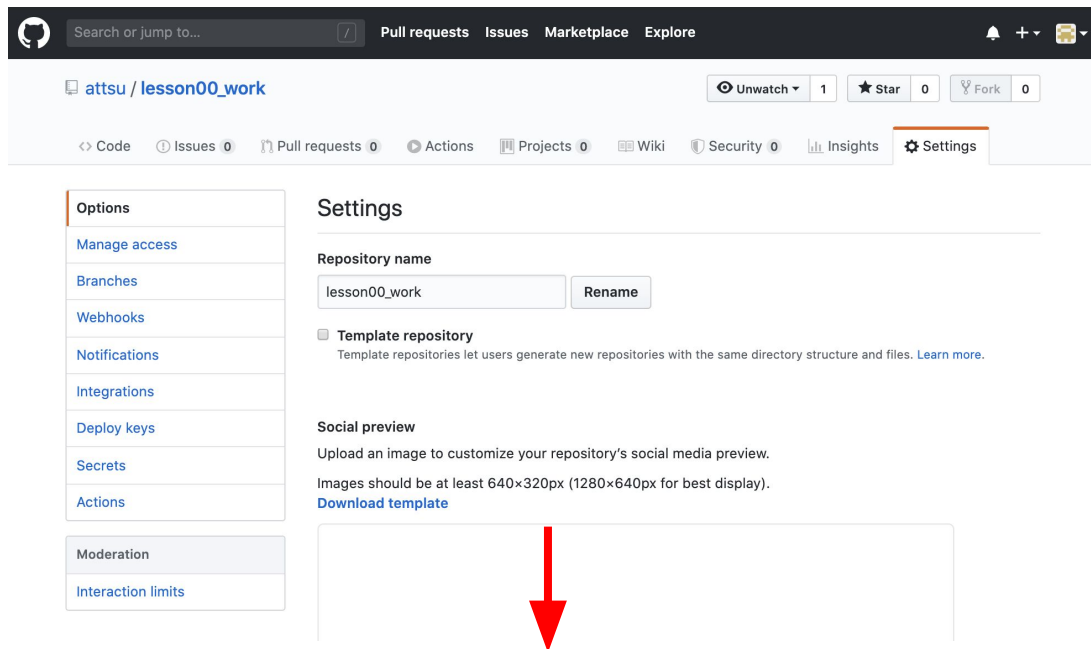
CSS	init	24 days ago
images	init	24 days ago
js	init	24 days ago
index.html	Naviを更新	2 hours ago

Help people interested in this repository understand your project by adding a README. Add a README

GitHubのSettingsページで、公開設定が必要です。



# サイトの公開について (GitHub)



The screenshot shows the GitHub interface for the repository 'attsu / lesson00\_work'. The top navigation bar includes links for Pull requests, Issues, Marketplace, and Explore. Below the repository name, there are buttons for Unwatch (1), Star (0), and Fork (0). The main navigation bar shows links for Code, Issues (0), Pull requests (0), Actions, Projects (0), Wiki, Security (0), Insights, and Settings (highlighted with an orange bar). On the left sidebar, the 'Options' section is expanded, showing links for Manage access, Branches, Webhooks, Notifications, Integrations, Deploy keys, Secrets, and Actions. Below this is the 'Moderation' section with a link for Interaction limits. The main content area is titled 'Settings' and contains the 'Repository name' field (lesson00\_work) with a 'Rename' button. Below this is the 'Template repository' section, which is currently unchecked. The 'Social preview' section is also visible, with instructions to upload an image to customize the repository's social media preview. A large red arrow points down to the 'Social preview' section.

Options

- Manage access
- Branches
- Webhooks
- Notifications
- Integrations
- Deploy keys
- Secrets
- Actions

Moderation

- Interaction limits

## Settings

Repository name

lesson00\_work [Rename](#)

☐ **Template repository**  
Template repositories let users generate new repositories with the same directory structure and files. [Learn more.](#)

**Social preview**  
Upload an image to customize your repository's social media preview.  
Images should be at least 640×320px (1280×640px for best display).  
[Download template](#)

Settingsページで、画面下部にスクロールし、公開設定を行います。

# サイトの公開について (GitHub)

## GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

✓ Your site is published at [https://attsu.github.io/lesson00\\_work/](https://attsu.github.io/lesson00_work/)

### Source

Your GitHub Pages site is currently being built from the master branch. [Learn more.](#)

master branch ▾

### Theme Chooser

Select a theme to publish your site with a Jekyll theme. [Learn more.](#)

Choose a theme

### Custom domain

Custom domains allow you to serve your site from a domain other than attsu.github.io. [Learn more.](#)

Save

### ☒ Enforce HTTPS

— Required for your site because you are using the default domain (attsu.github.io)

GitHub Pagesの項目で、Sourceの設定が「**master branch**」になっていると、サイトの URL が表示されます。  
こちらをブラウザでアクセスし、表示の確認を行ってください。

**none**などになっている場合は、URLが表示されません。

## 公開URLの形式

**https://アカウント名.github.io/リポジトリ名/**