# GitHub使用マニュアル（未経験者向け）

※ コード編集・Push・Pull Requestを用いないWeb操作ベースの入門ガイド

作成：古橋ゼミ4年佐藤愛妃（2025.04.06）

## はじめに

古橋研究室に所属したゼミ生にとって、最初の関門はGitHubGitHubを使いこなすことかもしれません。プロジェクト管理〜卒論に至るまで、成果物をGitHub上で提出するので、ぜひ早めに使い方をマスターしましょう！

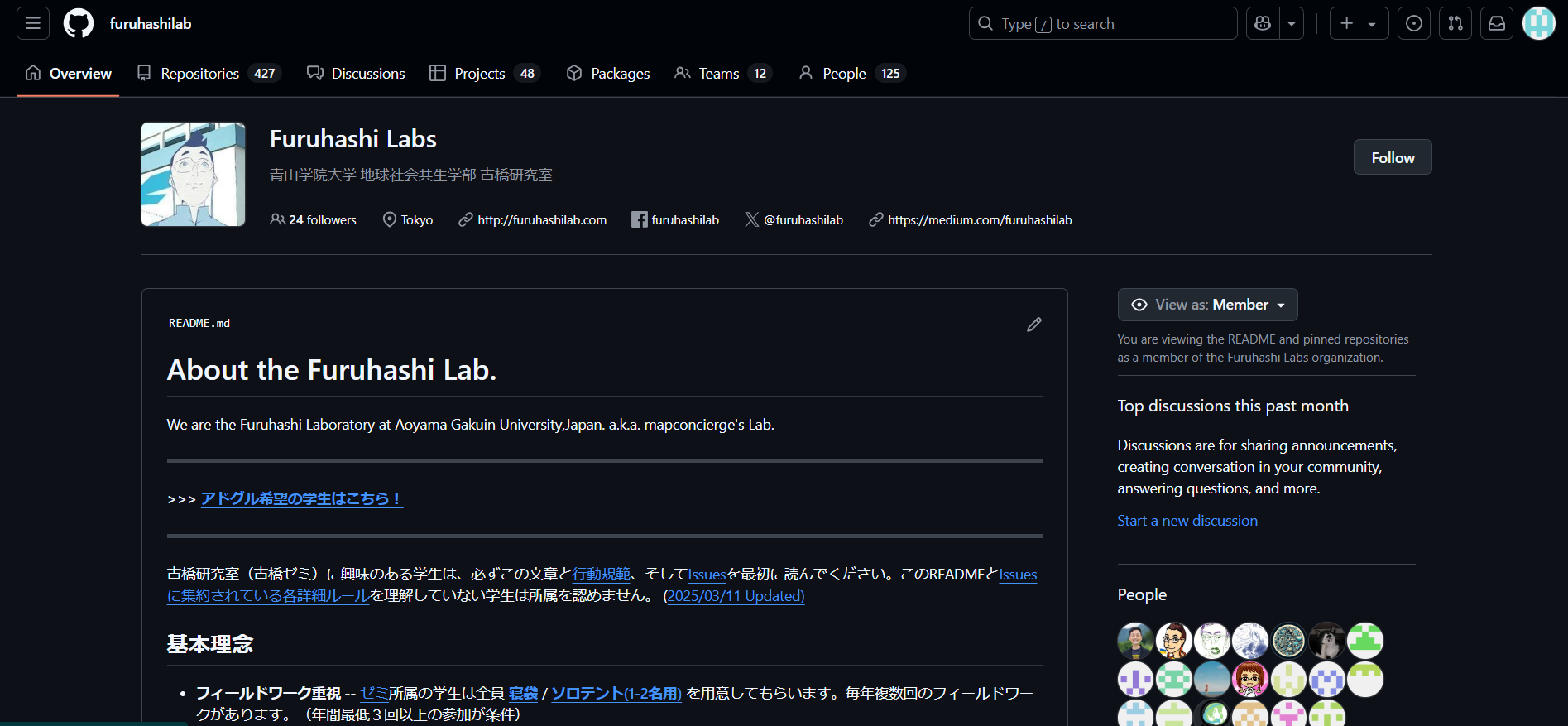
このマニュアルが皆さんの役に立てば嬉しいです(^▽^)/

## 1. GitHubとは？

GitHubは、ファイルの保存・共同作業・プロジェクト管理をWeb上で行えるサービスです。特に「誰が」「いつ」「どんな変更をしたか」を記録するバージョン管理機能があり、研究・課題・開発などさまざまな目的に利用できます。

Furuhashi Labs

<https://github.com/furuhashilab>



## 2. アカウントの作成

1. <https://github.com> にアクセスする。
2. 「Sign up」より新規アカウントを作成する。
3. メールアドレス、パスワード、ユーザー名を登録する。（メールアドレスは個人でも学校のドメインのアドレスでもOK）
4. 届いた確認メールのリンクをクリックし、登録を完了する。

## 3. リポジトリの作成方法

サンプルリポジトリ：<https://github.com/furuhashilab/GitHub-manual-beginners>　（作成2025.04.08 佐藤愛妃）

### 3.1 リポジトリとは？

リポジトリは、1つのプロジェクトに関するファイルをまとめて保管する場所です。フォルダのような役割を持っています。

### 

### 3.2 作成手順

Furuhashi labsのRepositoryを画面に表示し、右上の「＋」マーク →「New repository」をクリックします。

### 

次の項目を入力します：

* + Repository template：**No template**でOKです。
  + Repository name（必須）：例 GitHub-manual-biginners

※タイトルには、スペース使用不可（「-」に変換されます）。すべて英語で記入。

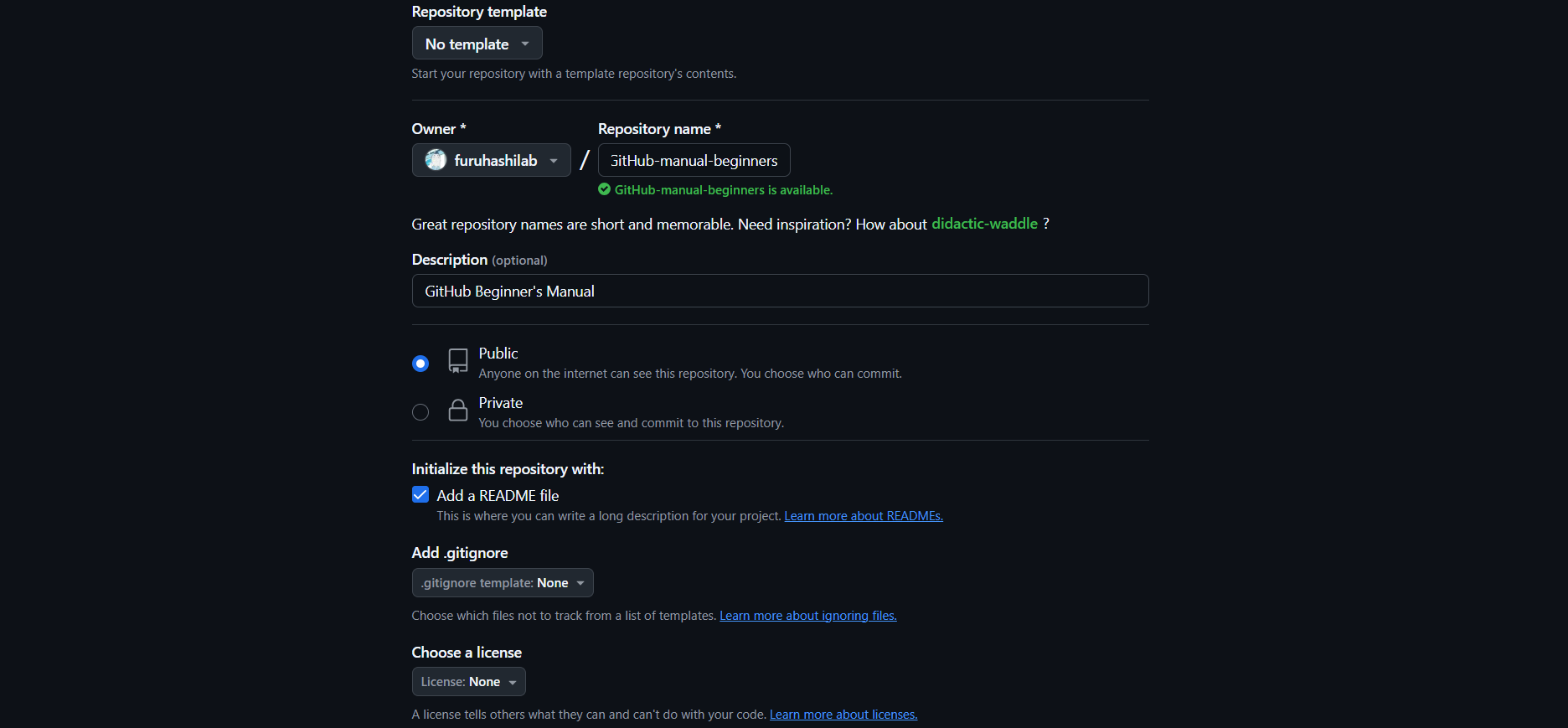
※OwnerはfuruhashilabのままでOK

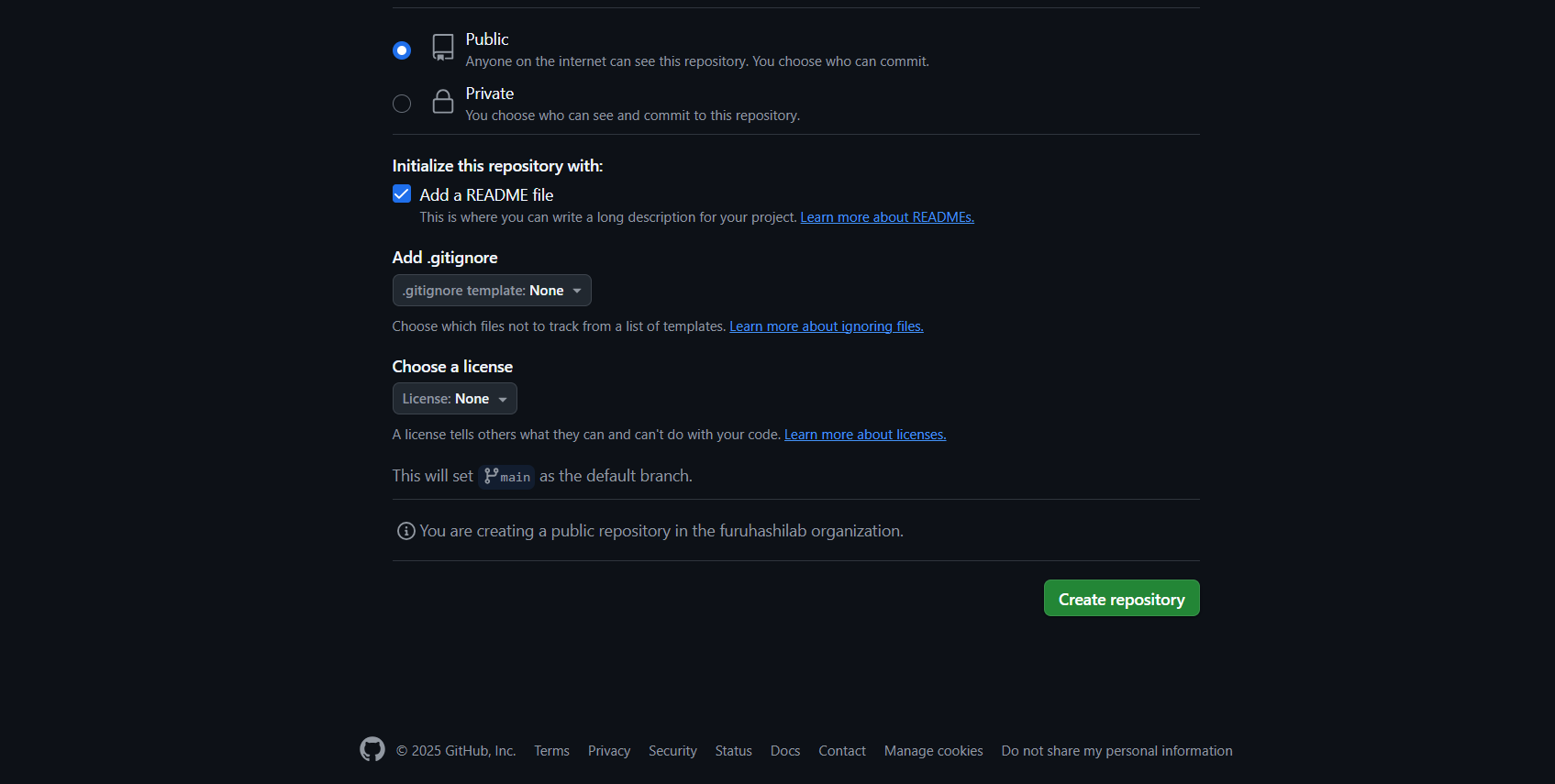
* + Description（任意）：例 リポジトリ内容の要約
  + Public / Private：ゼミのプロジェクトは基本**Public**です！
  + Initialize this repository with: **✅ 「Add a README file」にチェック**を入れてください。

※これにチェックすることで、Web上で編集できる初期ファイルが作成されます！後から追加することも可能です。

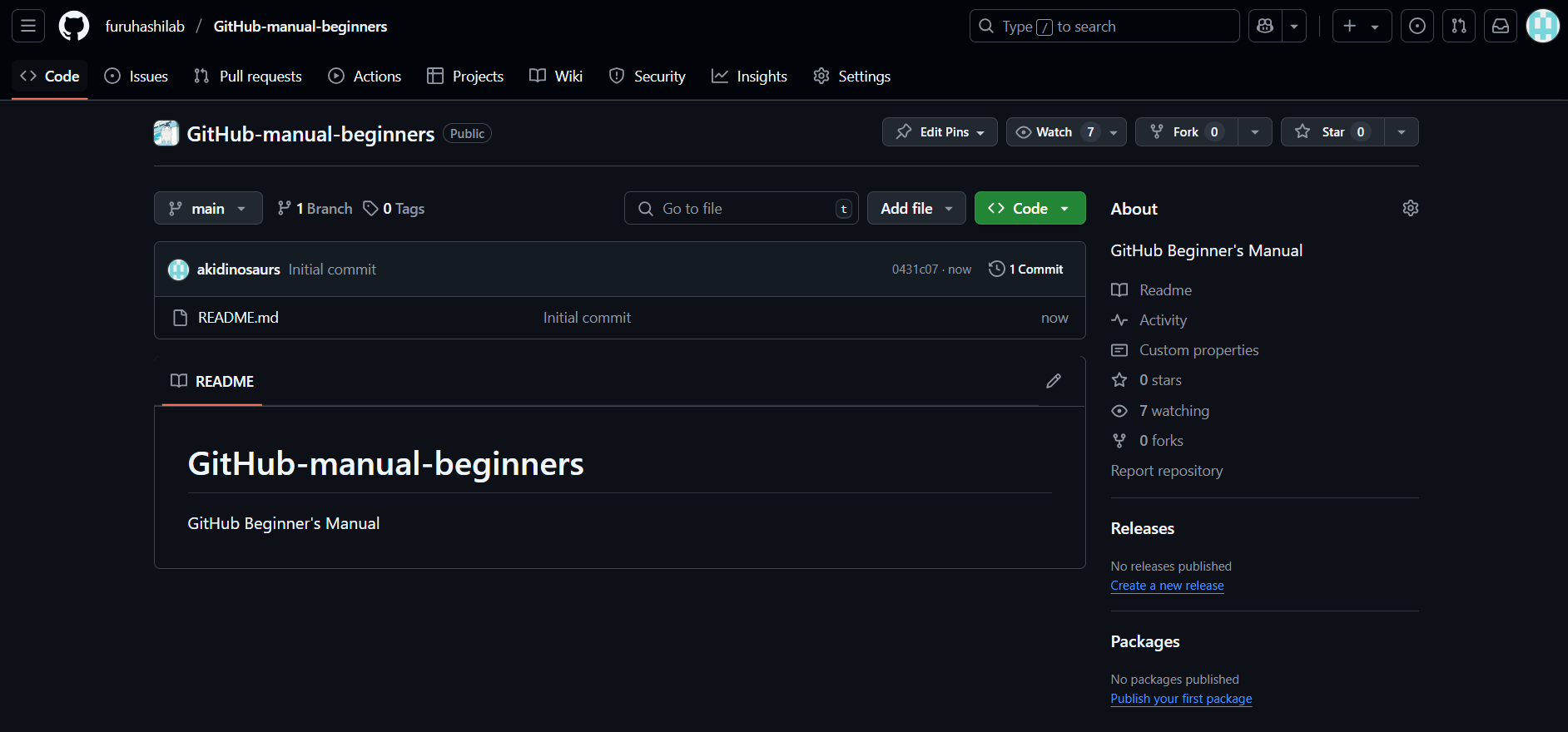
* + Add .gitignore：特に何もしなくて大丈夫です。
  + Choose a license：あとで、cc by 4.0のライセンスファイルを追加するので、これも触らなくて大丈夫です。

サンプルでつくったリポジトリ↓



右下の「Create repository」ボタンを押すと、作成完了です。

↓これで、リポジトリが立ちました。



ここまでお疲れ様です！

次は、ページ内のコンテンツとなるファイルやデータの編集方法を説明します！

## 4. READMEの編集方法

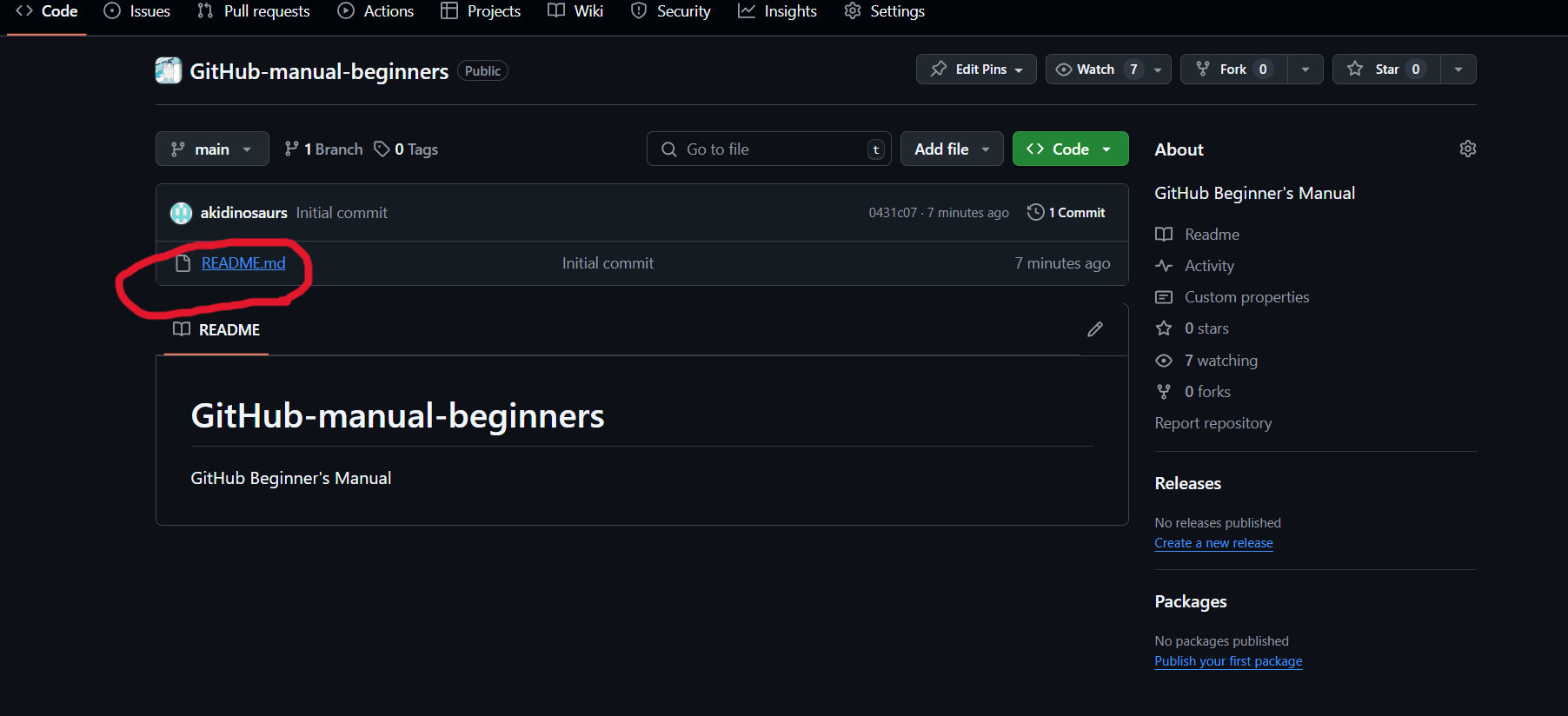
## READMEとは？

README（リードミー）ファイルは、プロジェクトの説明や使い方、目的などを記載するファイルです。リポジトリを開いたときに最初に表示されるため、名刺や案内板のような役割を持っています。そして、GitHubでページを編集するときは、**Markdown**という書き方を使用します！

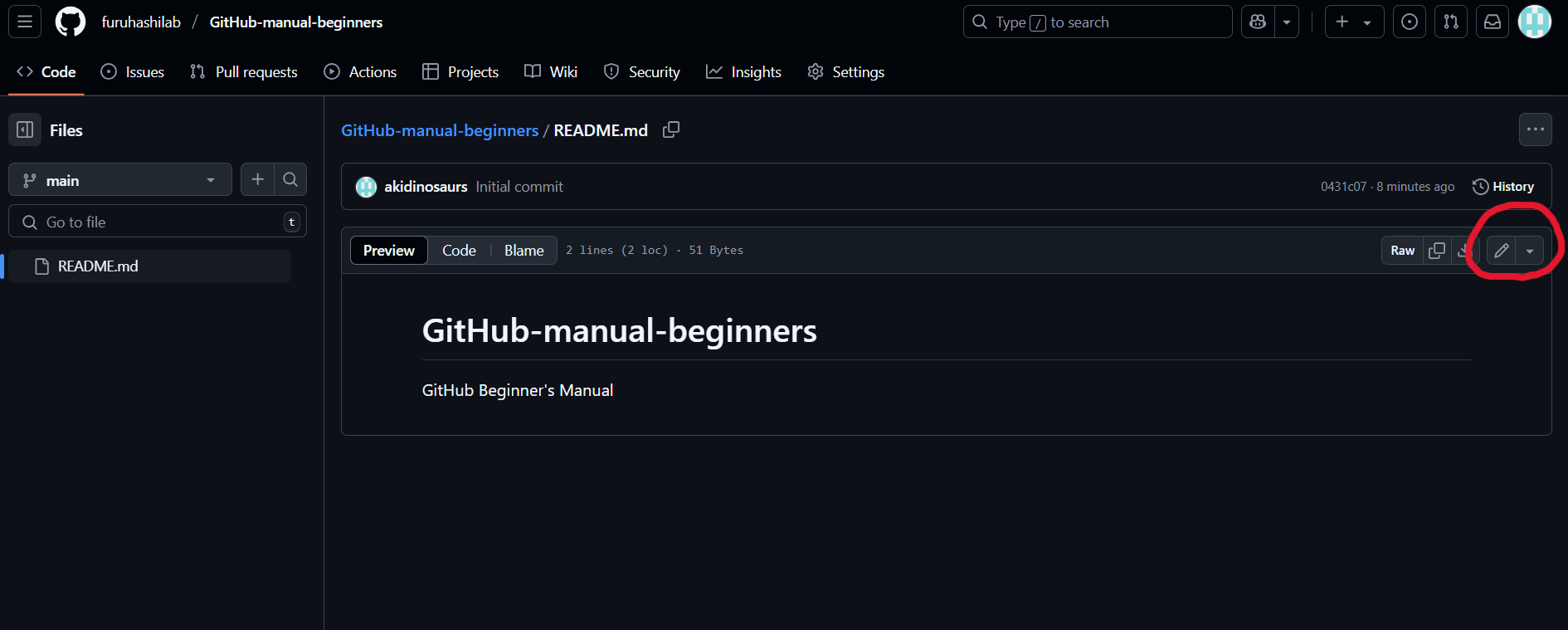
### 編集方法（Web上で操作する場合）

GitHubで対象のリポジトリを開きます。

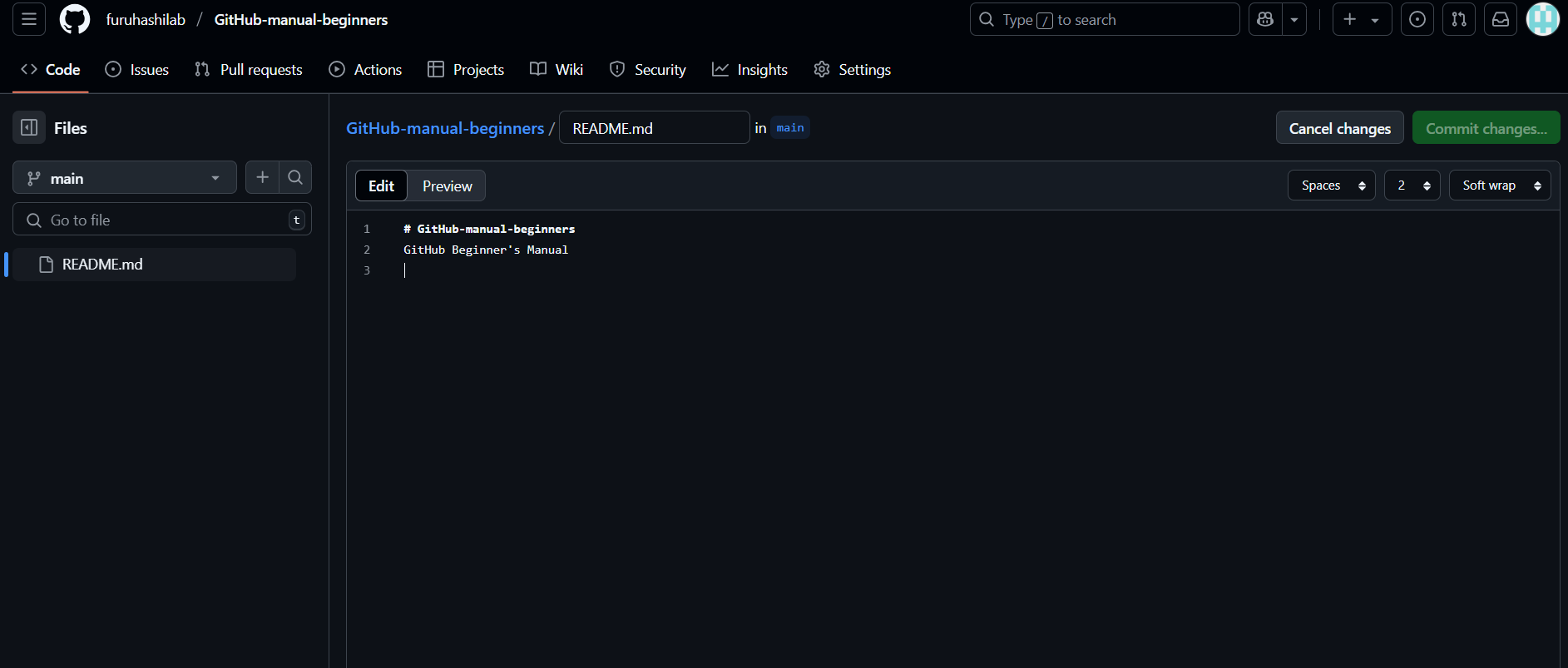
画面中央にある **README.md というファイルをクリック**します。

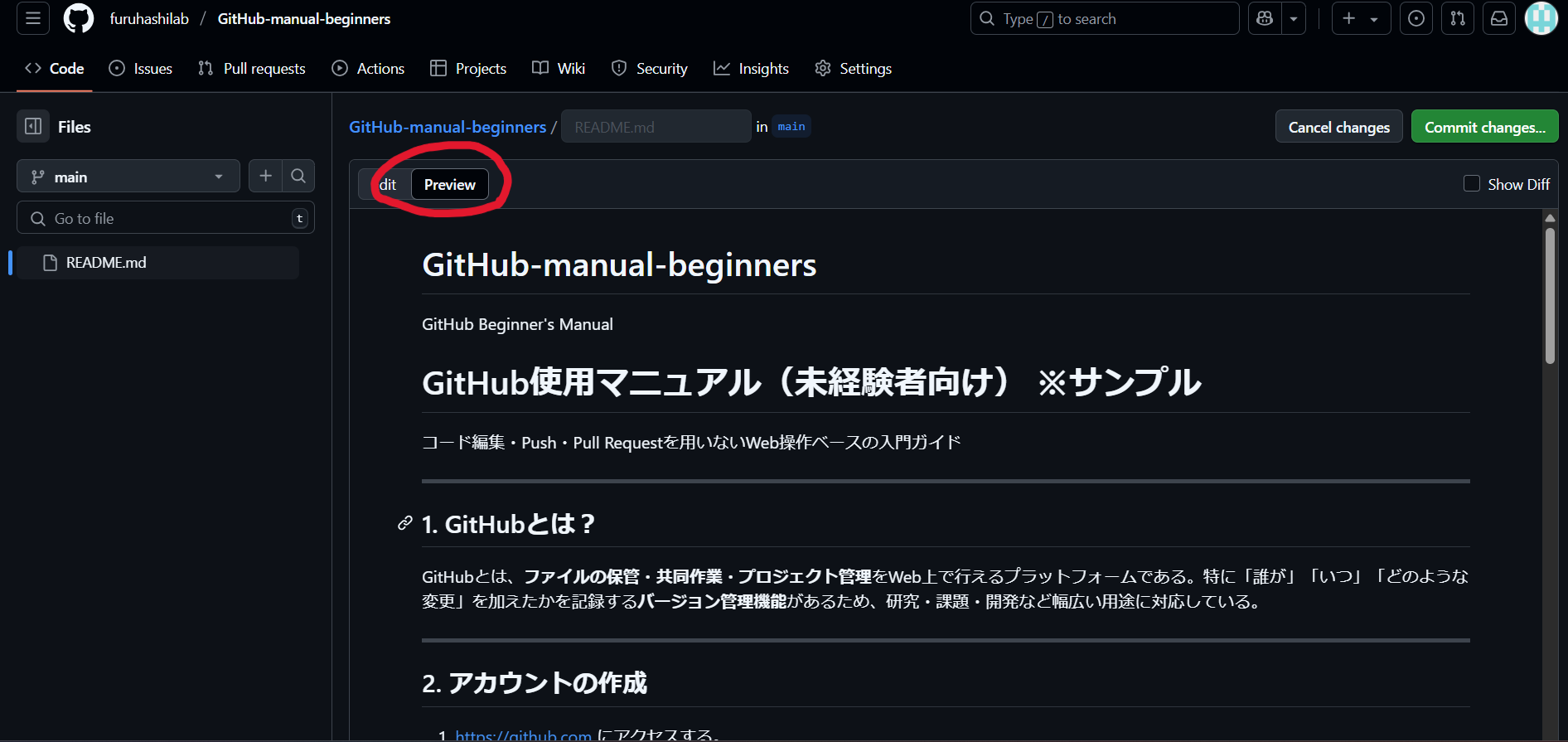


**右上の「✏️ Edit this file」ボタンをクリック**します。

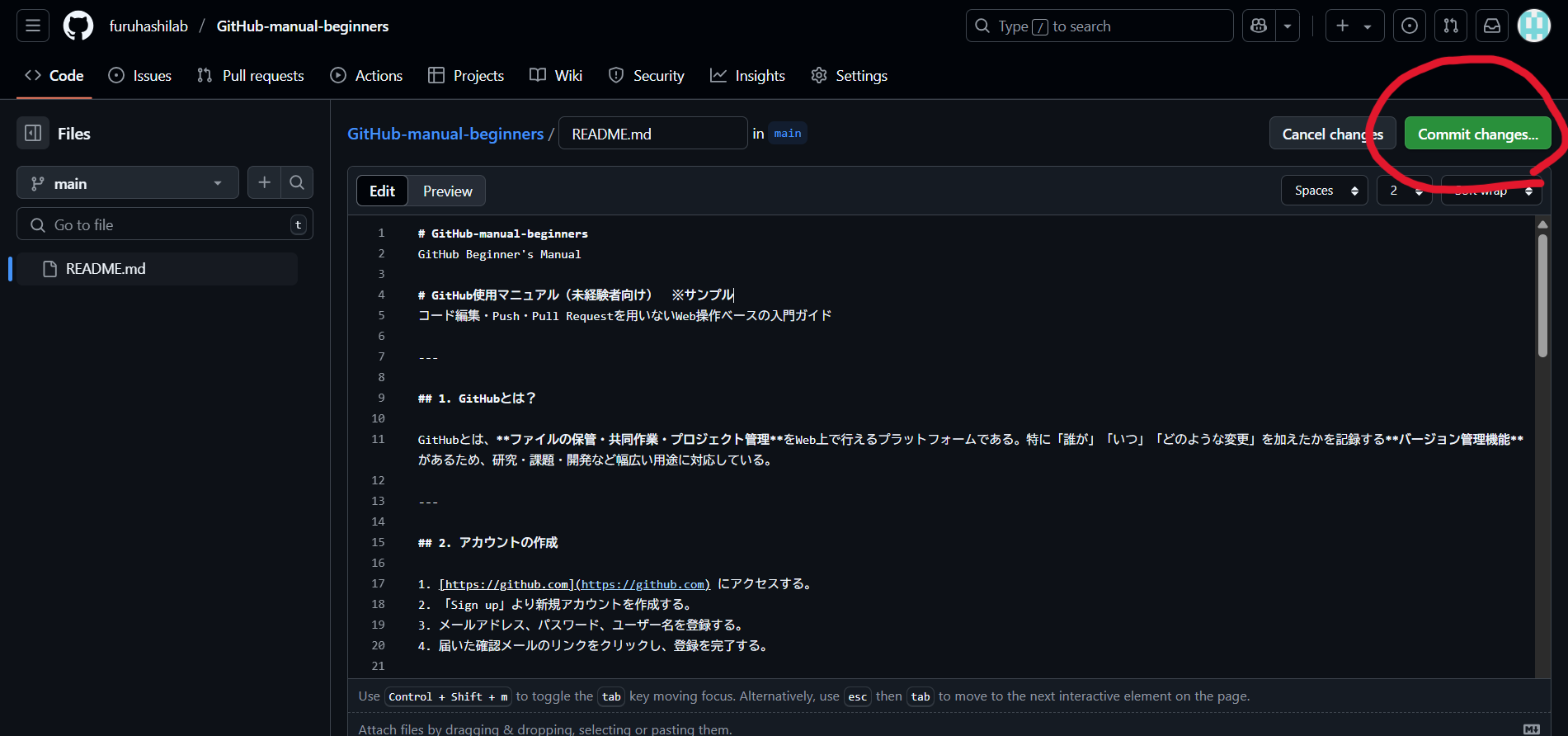


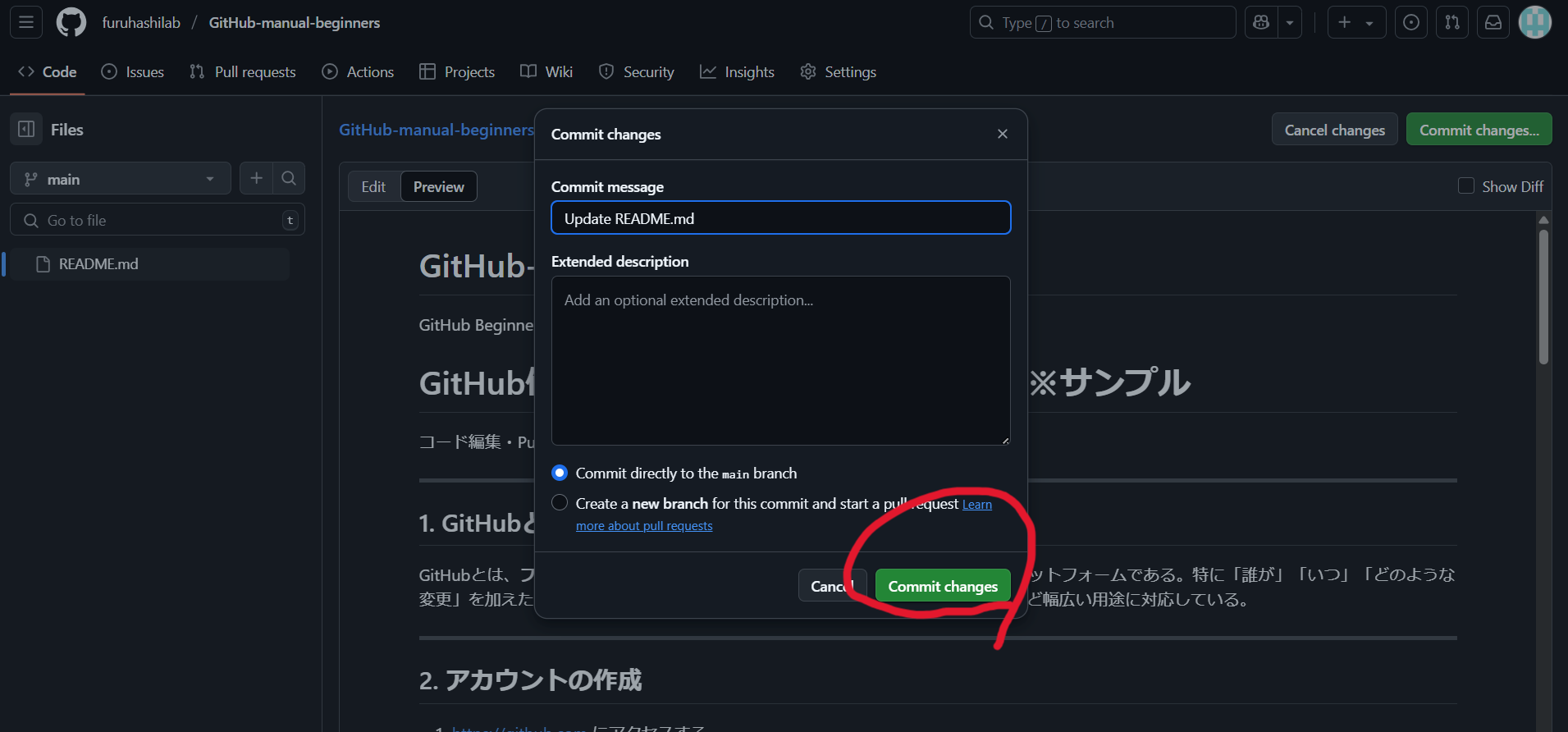
内容を編集します（**Markdown方式**で記述されます）。



編集を反映する前に、「Edit」横の**「Preview」**を選択して、出来栄えを確認してみてください！とても便利な機能です(⌒∇⌒)

「Commit changes」ボタンを押すと、編集内容が保存されます。



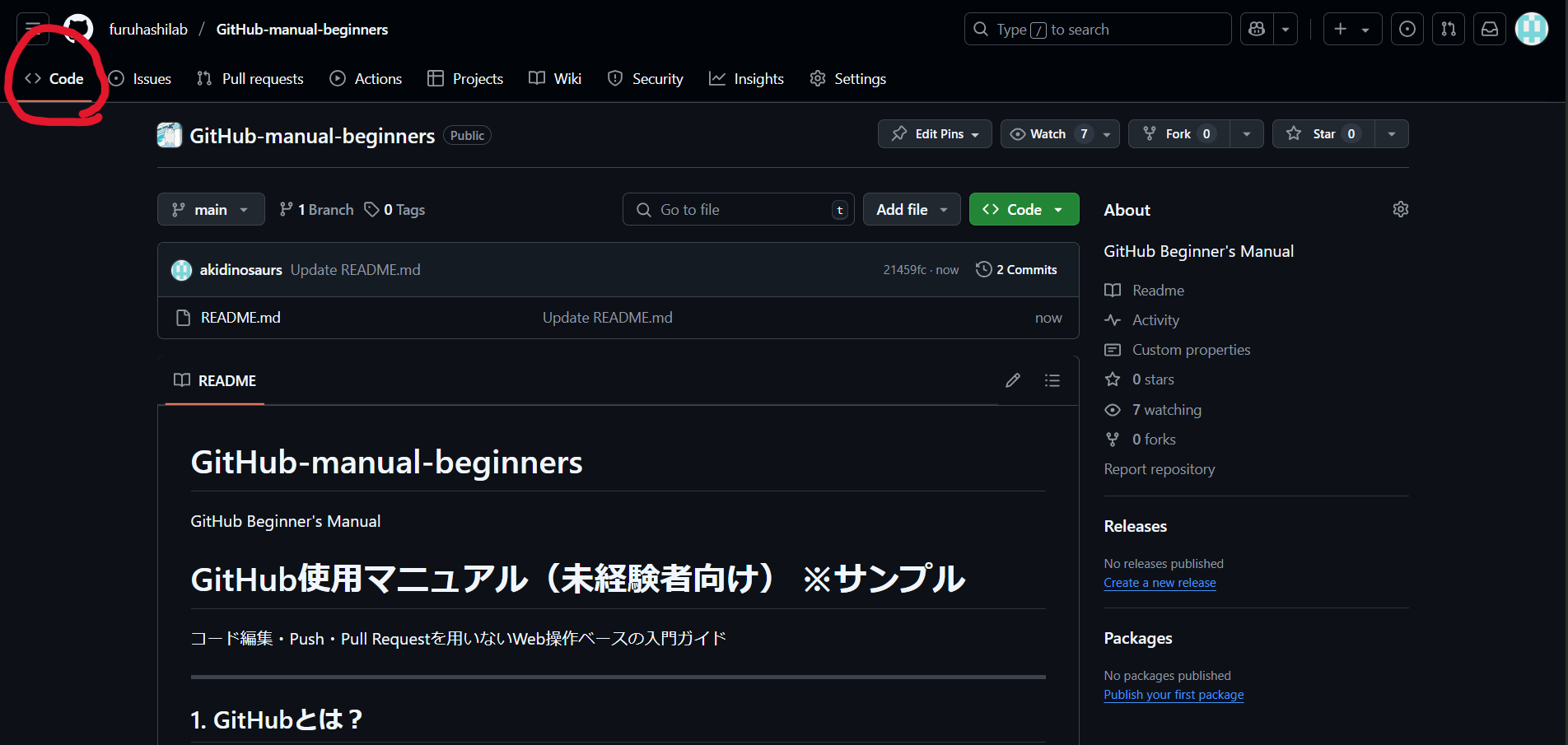


編集内容が反映され、READ. meが最新バージョンにアップデートされます



## 

## 「code」リポジトリ内のホーム画面



**code**は、**リポジトリ内のホーム画面**です！このcodeには、上部分にリポジトリに追加されているデータやファイルが置かれ、その下にREADMEが表示されるようになってます。

## MarkDownについて

Markdownとは、見出し・箇条書き・表などをシンプルに記述できる文書フォーマットです。GitHubではREADMEやIssue、Projectなどで使えます！

以下に、よく使う記法と参考になるwebサイトのURLを載せています。ご参照ください。

## **よく使う記法一覧**

| **機能** | **書き方** | **表示例** |
| --- | --- | --- |
| 見出し | # 見出し1 ～ ###### | 見出し1 |
| 太字 | \*\*太字\*\* | **太字** |
| 斜体 | \*斜体\* | *斜体* |
| リスト | - アイテム | - アイテム |
| 番号付き | 1. 項目 | 1. 項目 |
| リンク | [名前](URL) | [GitHub](https://github.com) |
| コード | `コード` | コード |
| 引用 | > 引用文 | > 引用文 |
| 区切り線 | --- または \*\*\* | ――― |

参考ページ：

1 <https://qiita.com/Qiita/items/c686397e4a0f4f11683d>

2 <https://docs.github.com/ja/get-started/writing-on-github/getting-started-with-writing-and-formatting-on-github/basic-writing-and-formatting-syntax>

（2025.04.06閲覧）

### 注意点

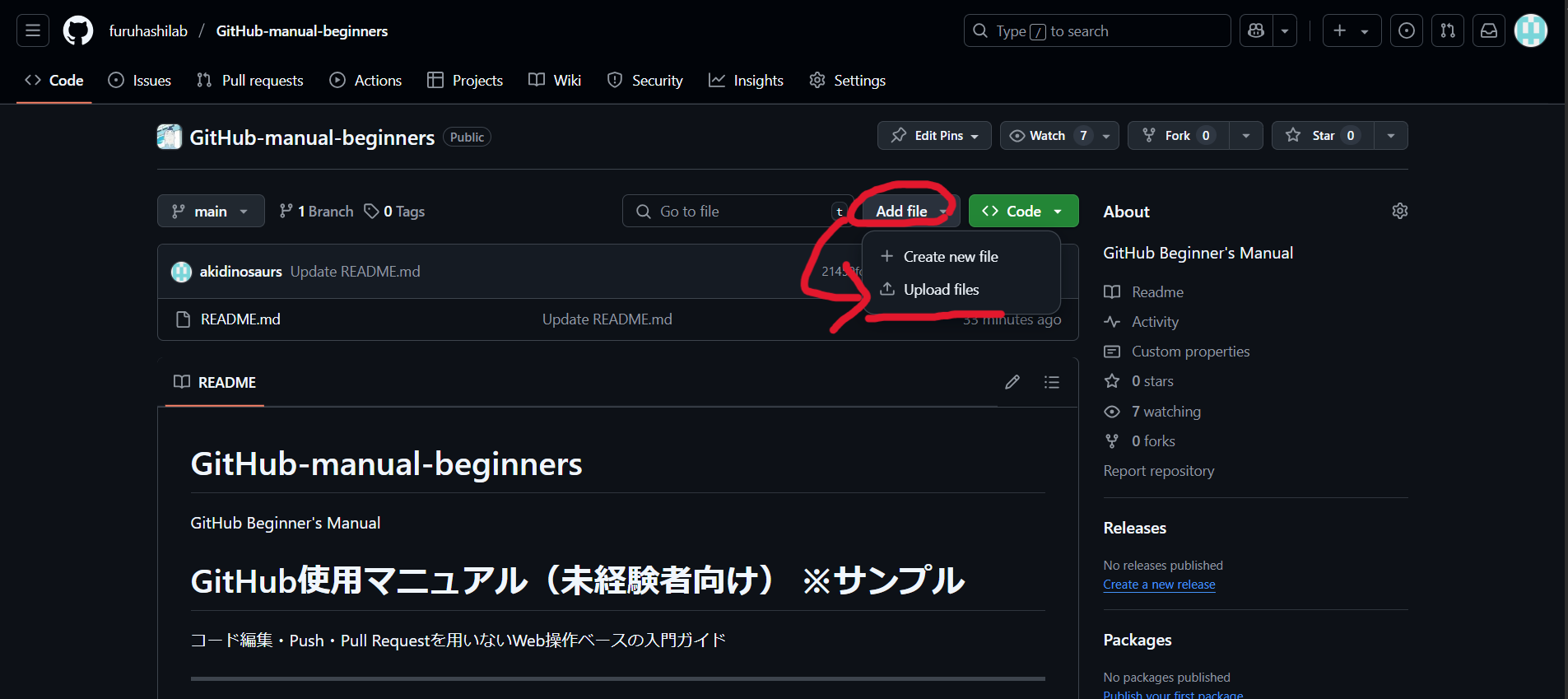
* .mdファイル（Markdownファイル）は、基本的にテキストエディタで編集可能です。
* GitHub上で編集すれば、自動でMarkdown形式として反映されます。
* 書き方に迷ったときは、既存のREADMEを参考にするのが近道です。

## 

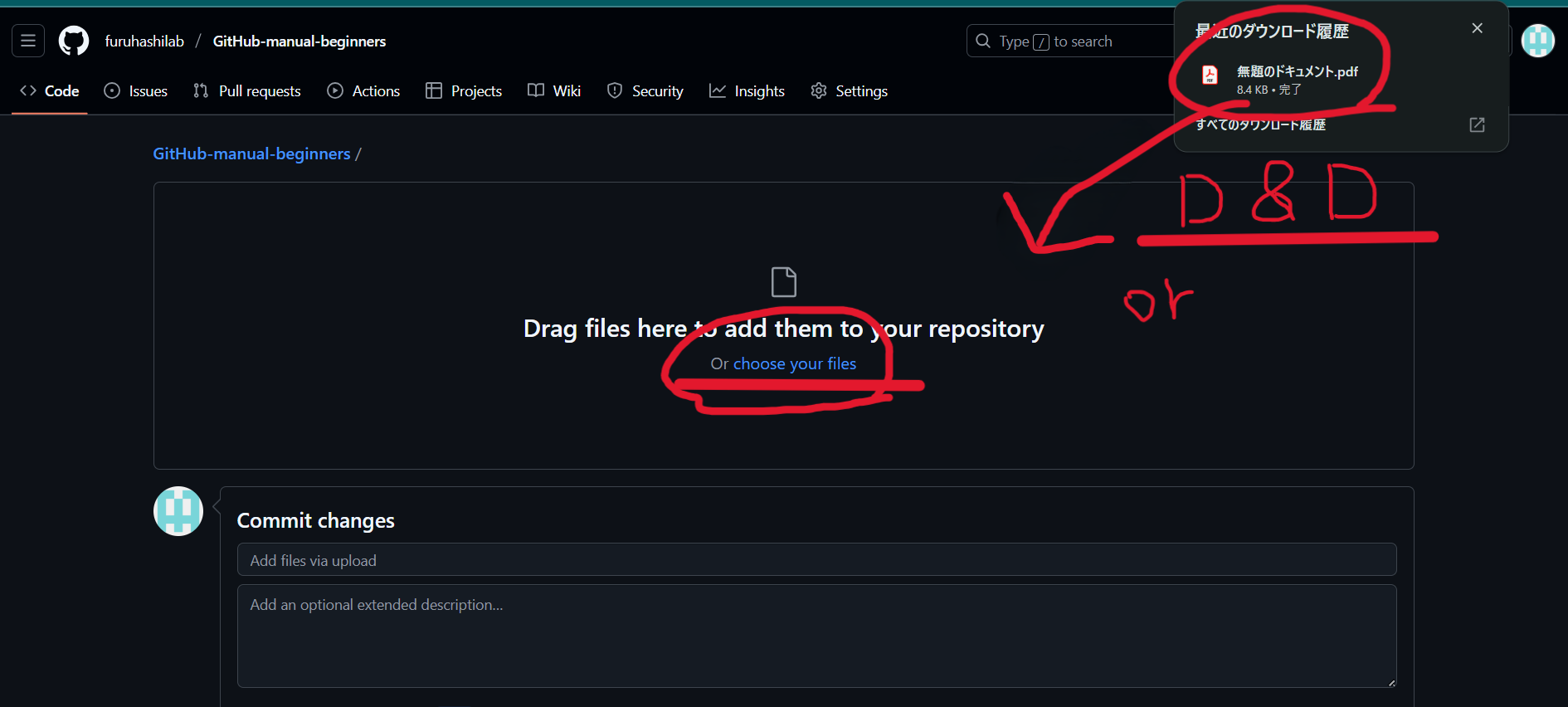
## 6.ファイルのアップロードと編集

### 6.1 ファイルをアップロードする方法

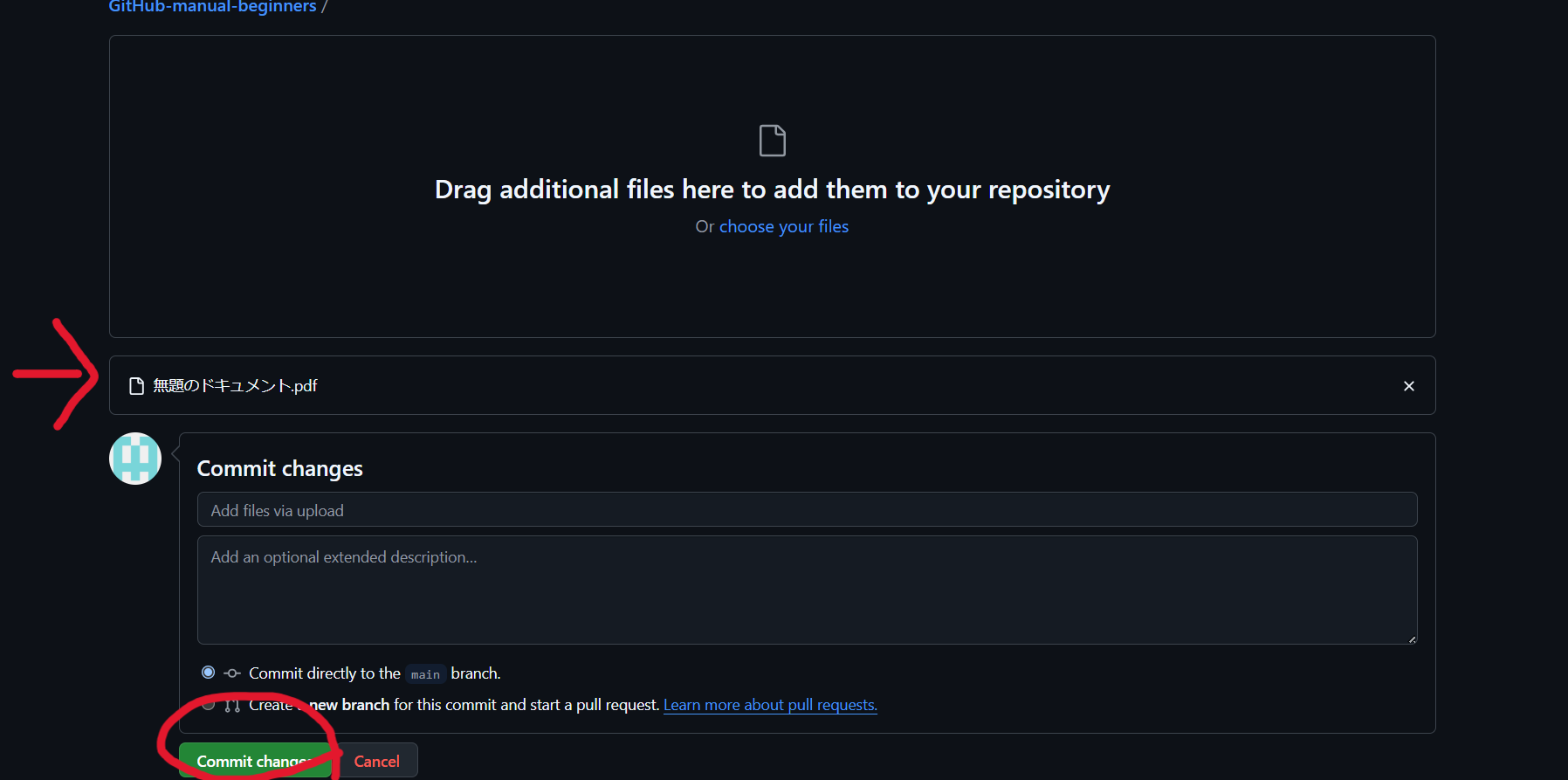
作成したリポジトリを開きます。

「Add file」→「Upload files」をクリックします。

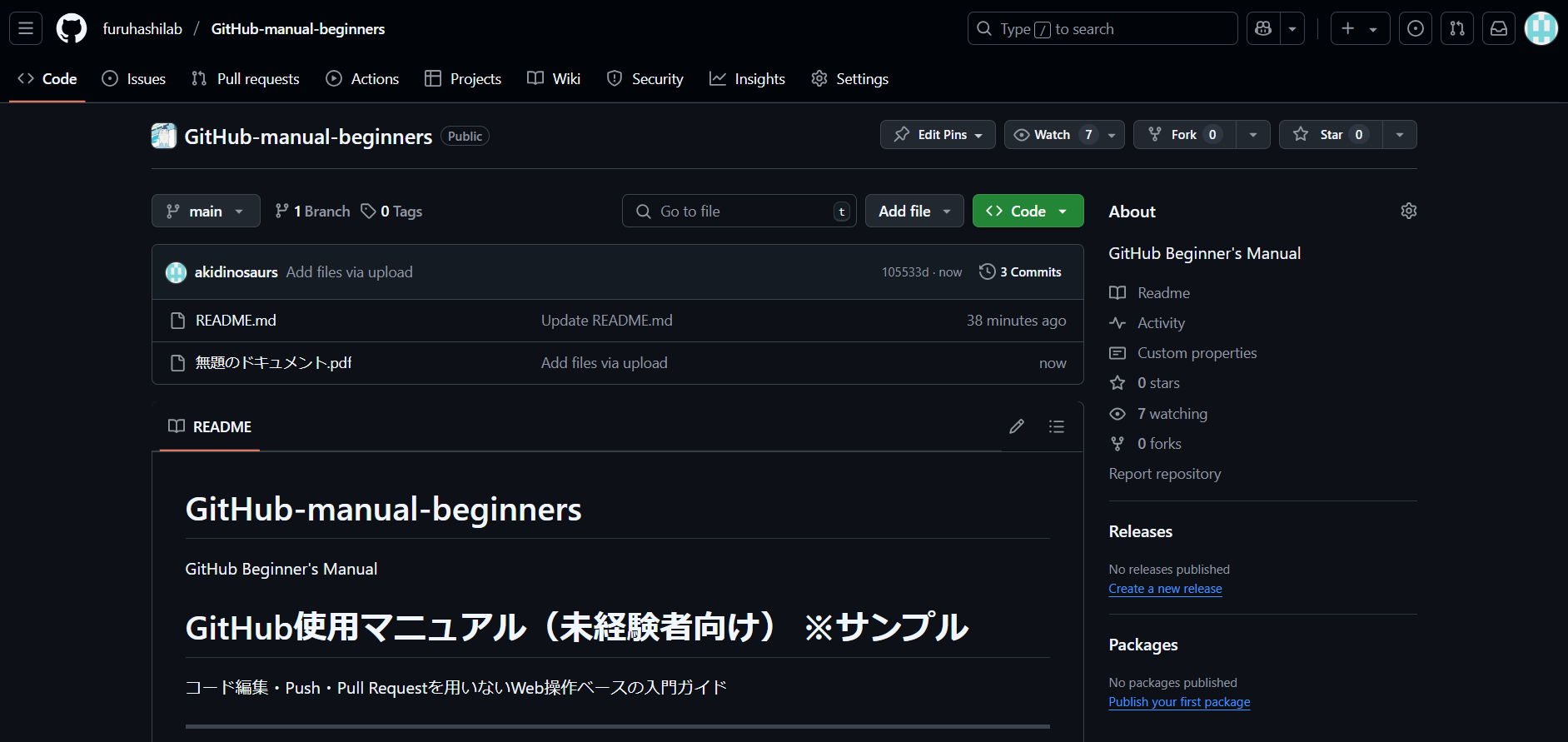
アップロードしたいファイルをドラッグ＆ドロップか、参照などから選択します。



下にある「Commit changes」ボタンを押すと保存されます。



これで追加完了です。READMEファイルの下に追加されました！



### 6.2 ファイルをWeb上で編集もできます（〇:txt, ×:pdf）

リポジトリ内のファイルをクリックします。

右上の「✏️ Edit this file」ボタンをクリックします。

内容を編集したあと、「Commit changes」ボタンで保存します。

### 

## 7. Issueを使ったタスク管理

### 7.1 Issueとは？

### Issueは、**やることリスト（ToDo）やメモ、質問などを記録・共有できる掲示板機能**です。1人での作業にも、チーム作業にも役立ちます。ゼミでGitHubでプロジェクトを行う場合は、よくissueを立てて、進捗確認やタスクごとの作業を行います。

### 7.2 Issueを作成する手順

### リポジトリ内の**「Issues」**タブをクリックします。

### 

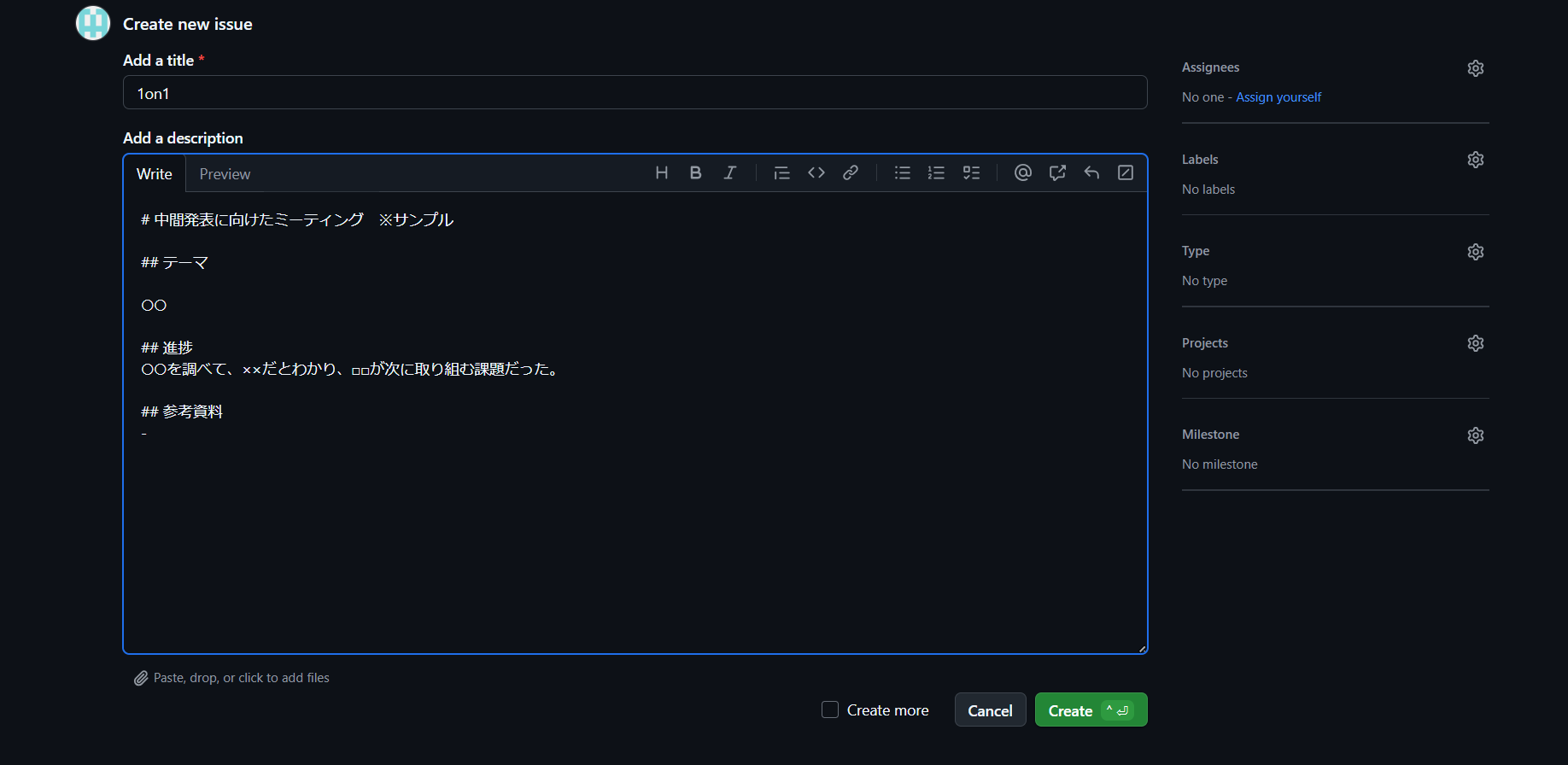
### **「New issue」**ボタンを押します。

### 

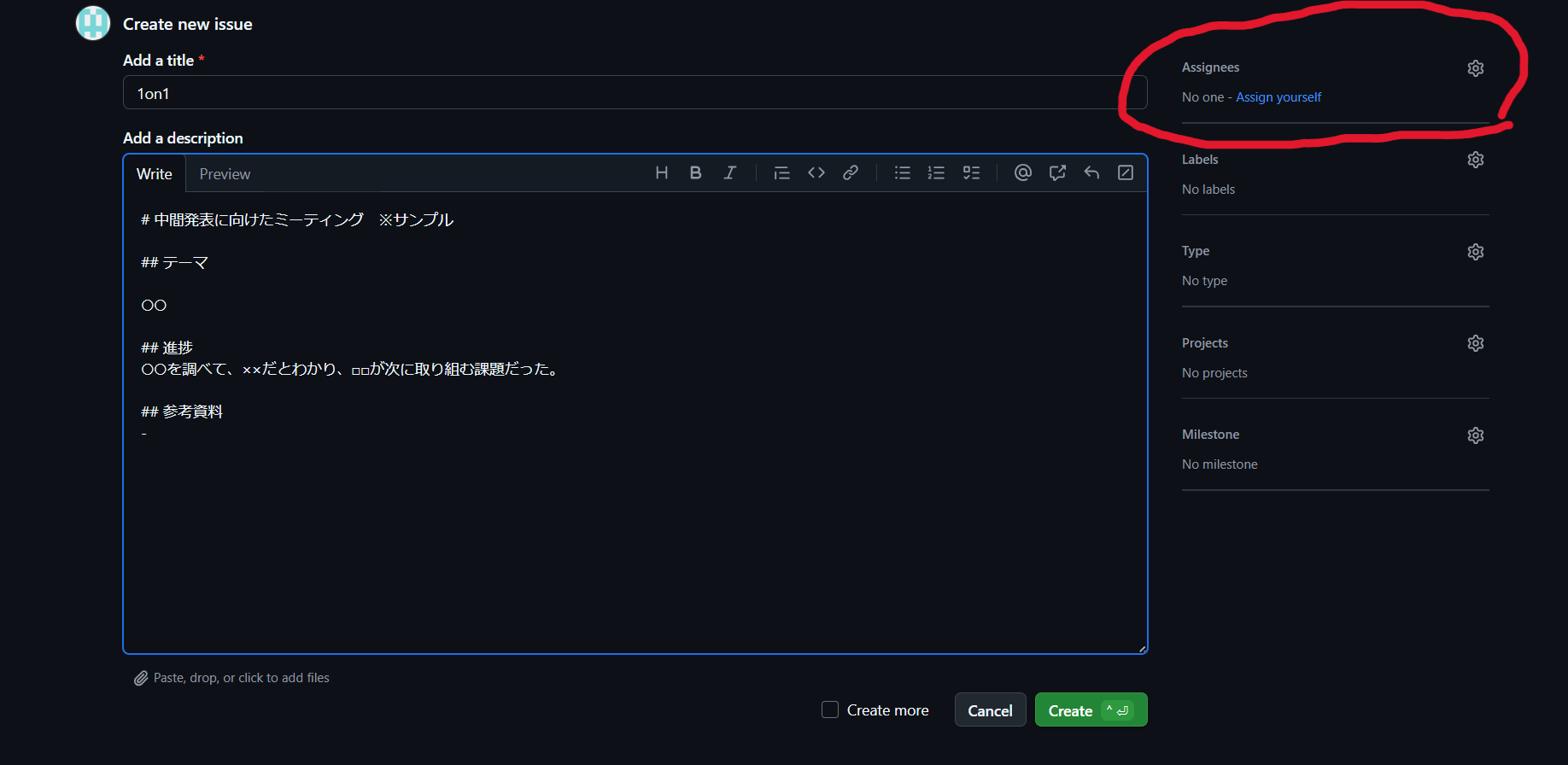
### 以下の項目を入力します：

### **Title（件名）**：例「1on1ミーティング」

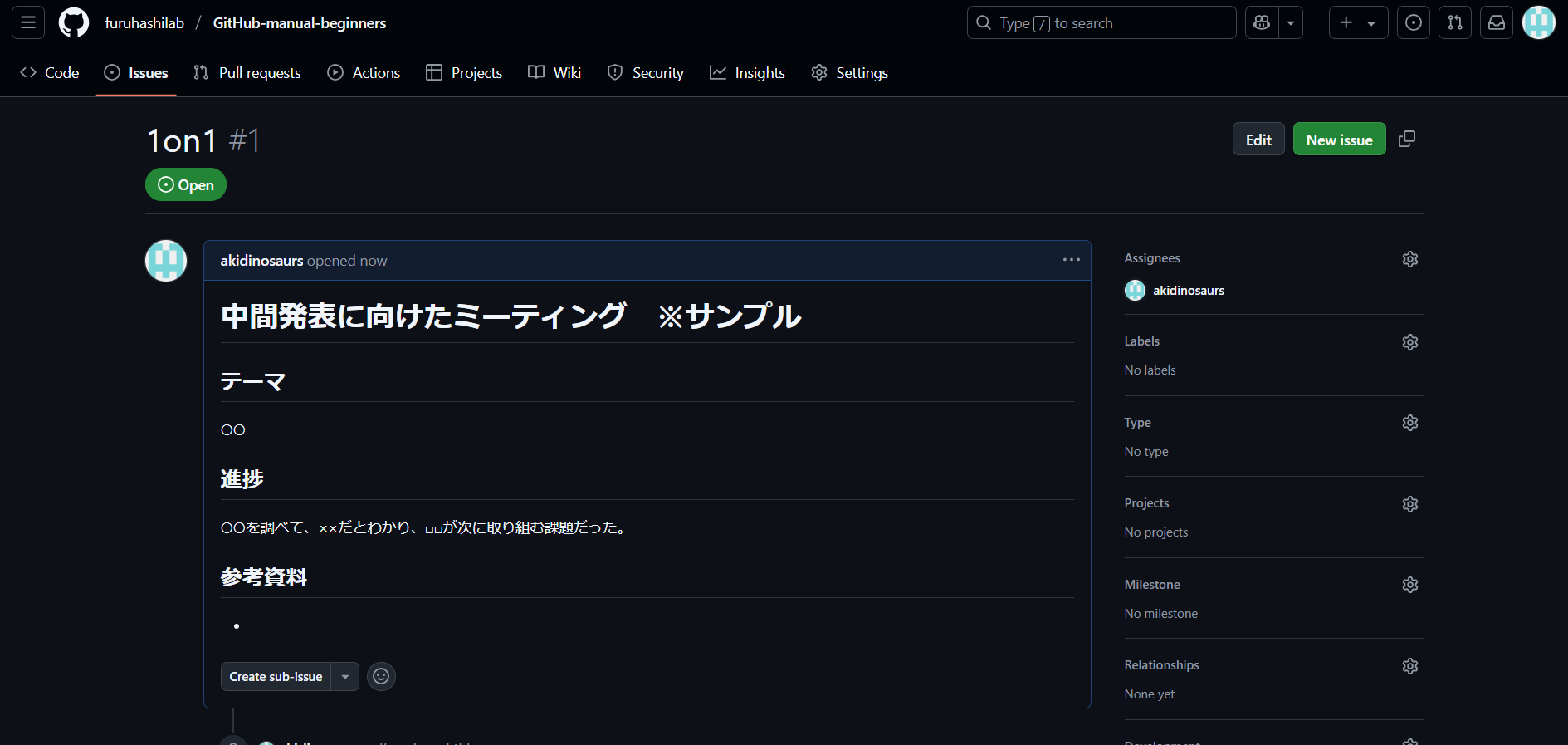
### **Description（説明）**：例「ゼミの中間発表に向けたミーティング」。内容も記載すると良いです。

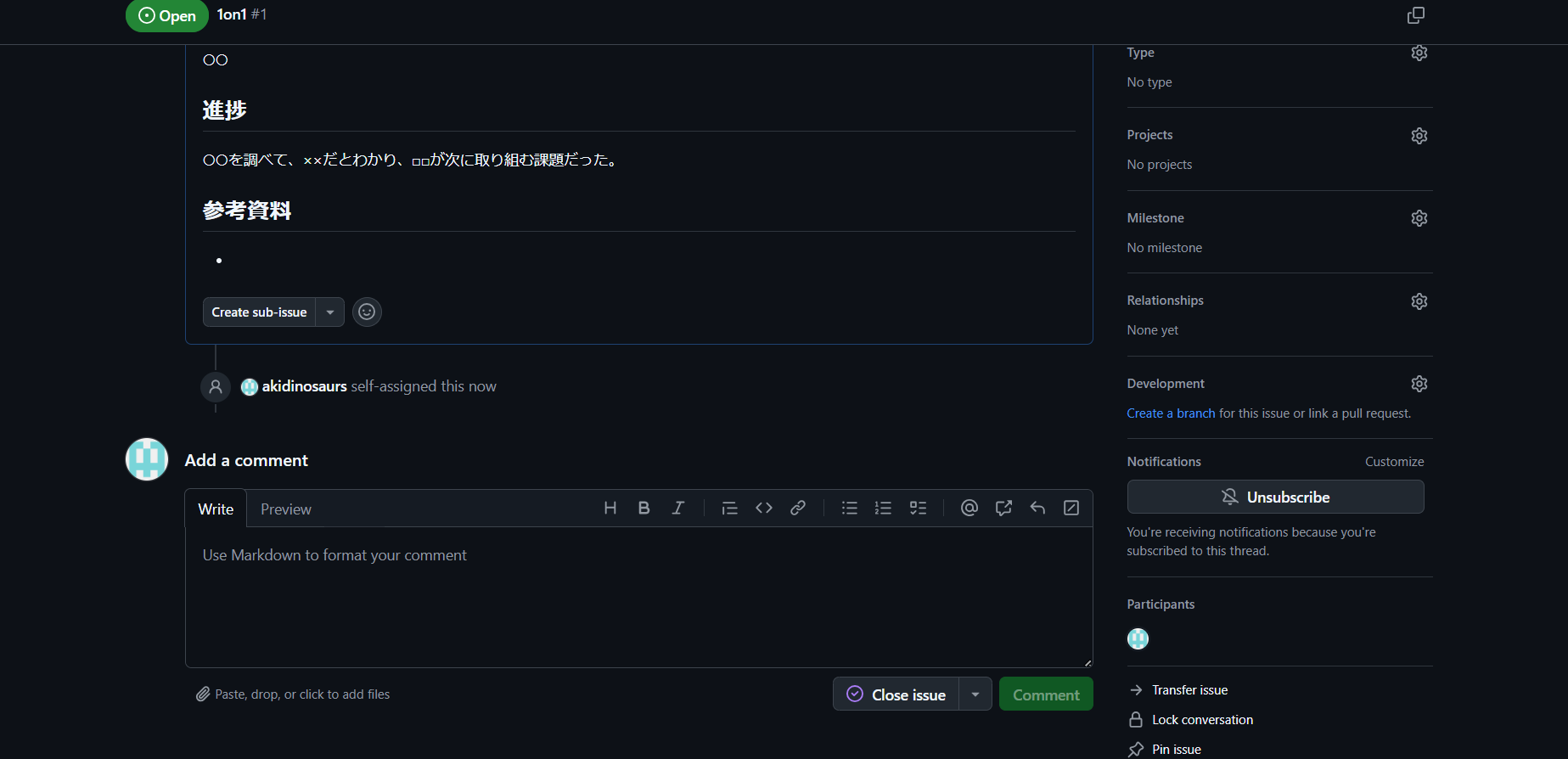


### **「Assign」**で、issueを更新予定の人を選択します。複数人での共同作業で、使用する機能です。**自分ならAssignyouself**、複数人から指名するときは**設定マークからメンバーの選択＆検索が可能**です。



### **「create」ボタンをクリック**して登録します。これでissueの完成です。isuueは、タイムラインのように、複数の投稿を作成できます。





### 

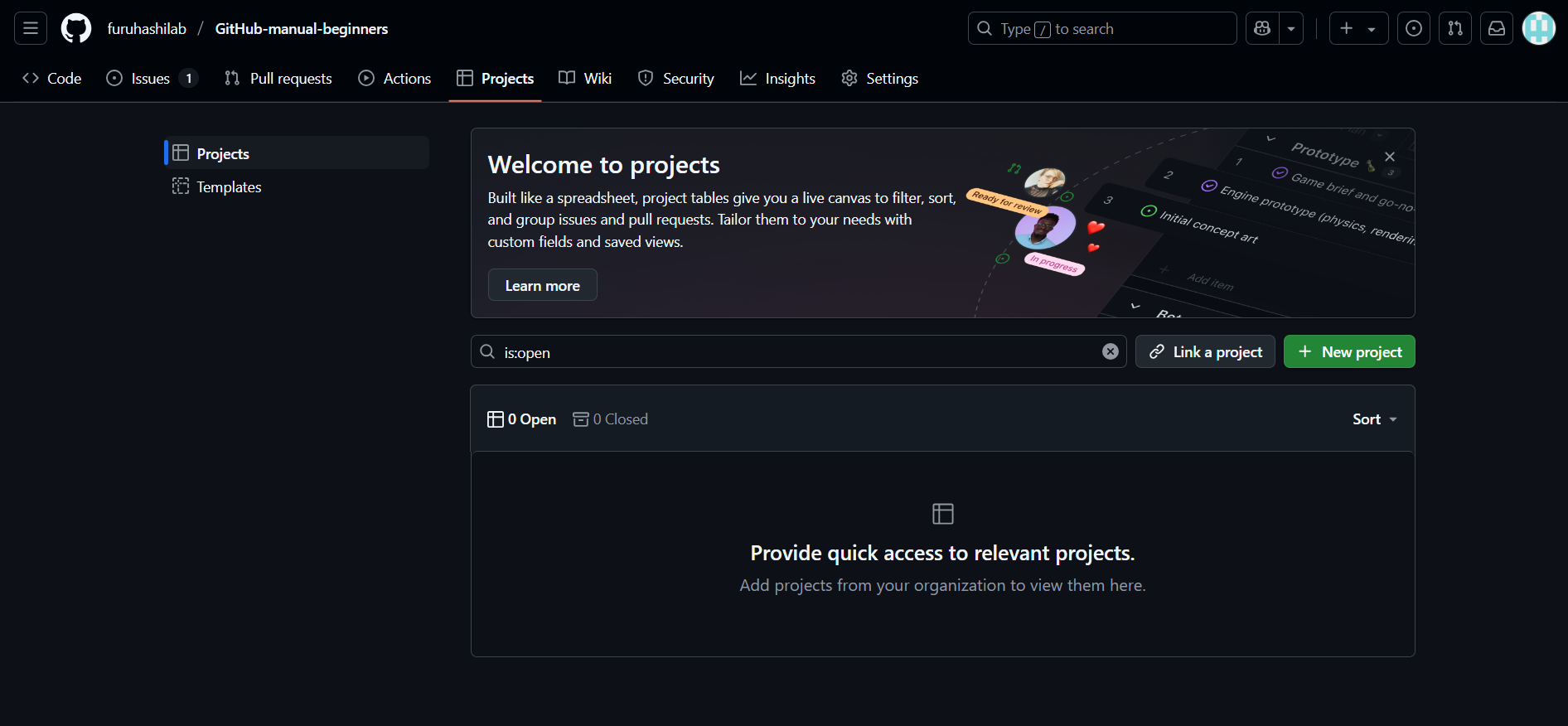
## 8. Projects（プロジェクト管理）を使う

### 8.1 Projectsとは？

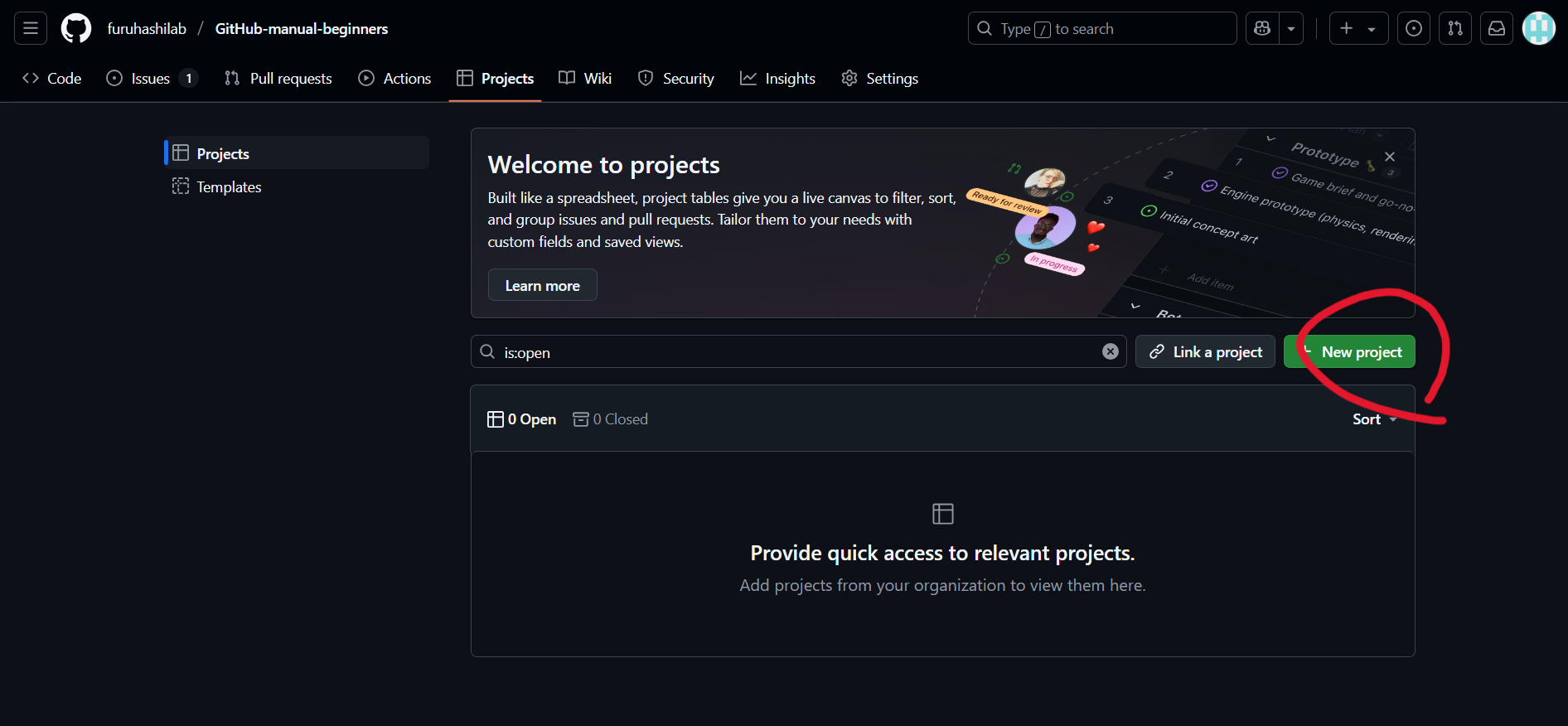
Projectsは、**Issueなどをカード形式で管理できる「かんばんボード」機能**です。「To do」「In progress」「Done」といった列に分けて、視覚的に進捗を管理できます。

### 8.2 Projectを作成する手順

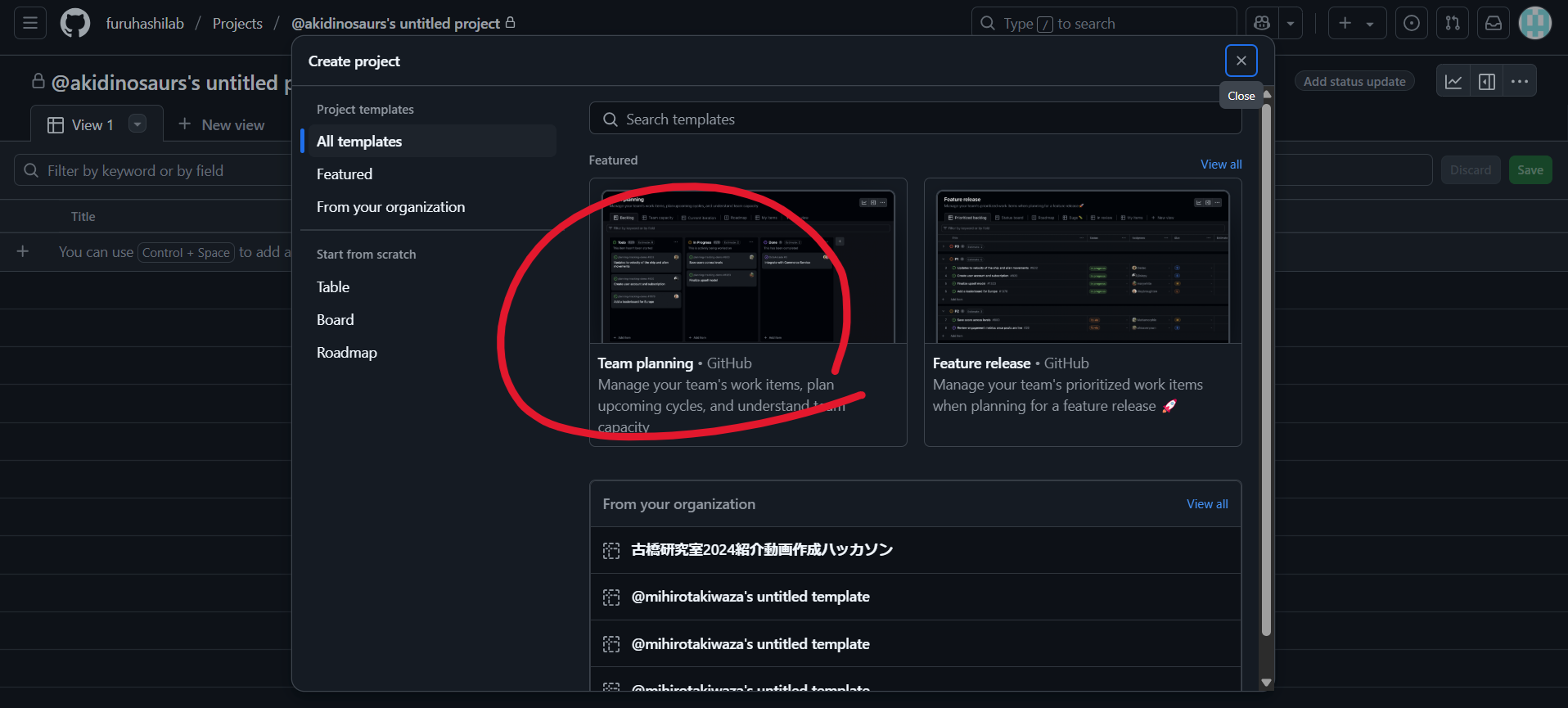
1. リポジトリ内の「Projects」タブをクリックします。



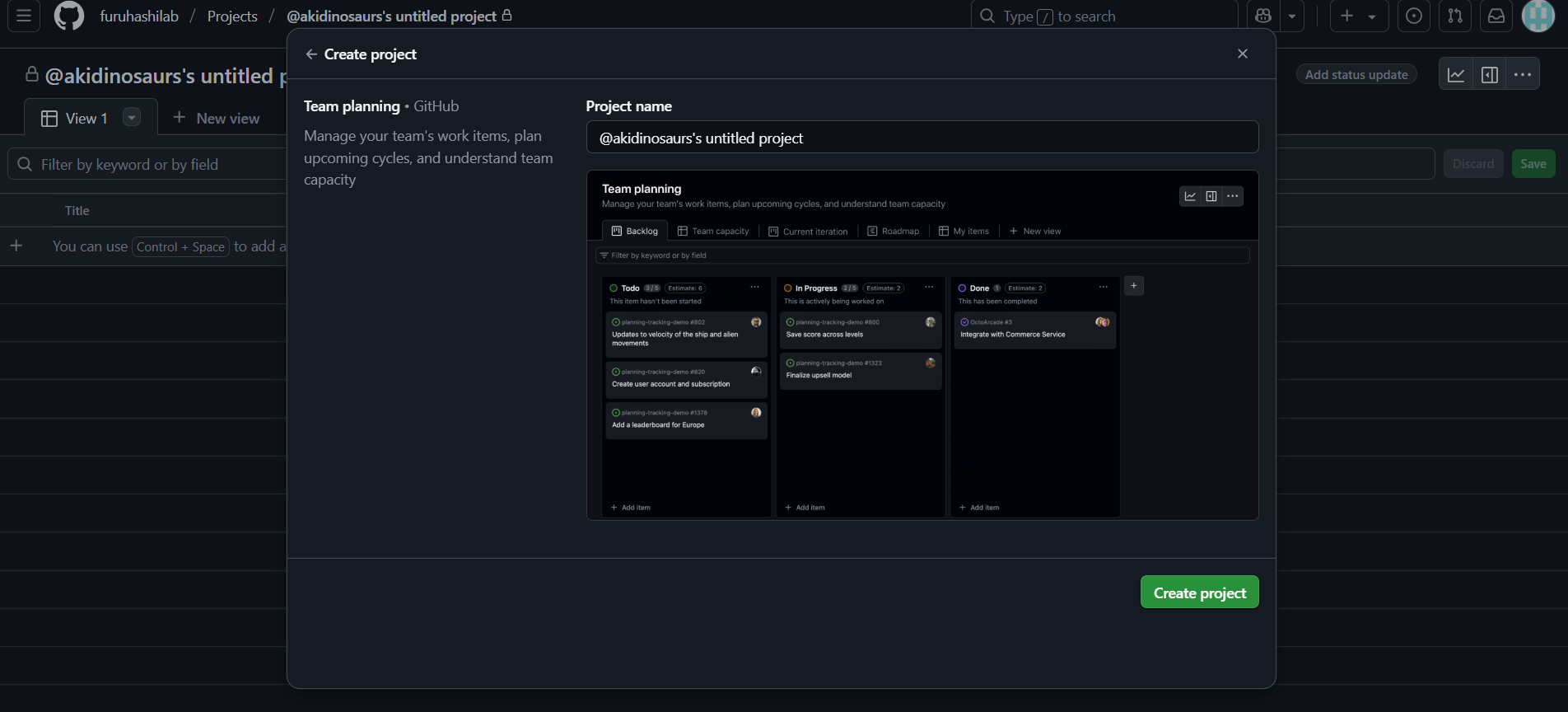
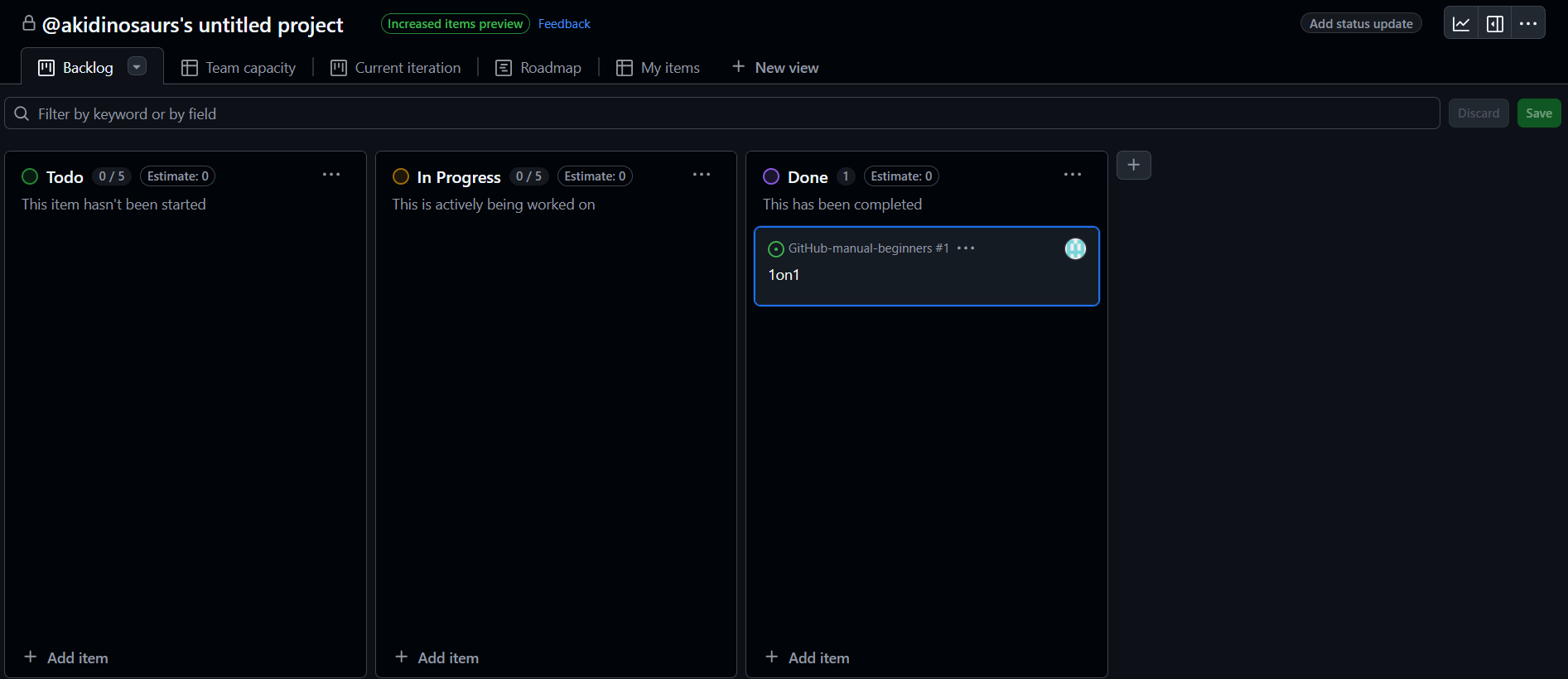
1. 「New project」→「Classic project」を選択します。



1. ボード名を入力し、テンプレートは「Team planning」を選びます。共同作業でタスクの進捗管理におすすめです。



1. 「Create project」ボタンをクリックします。

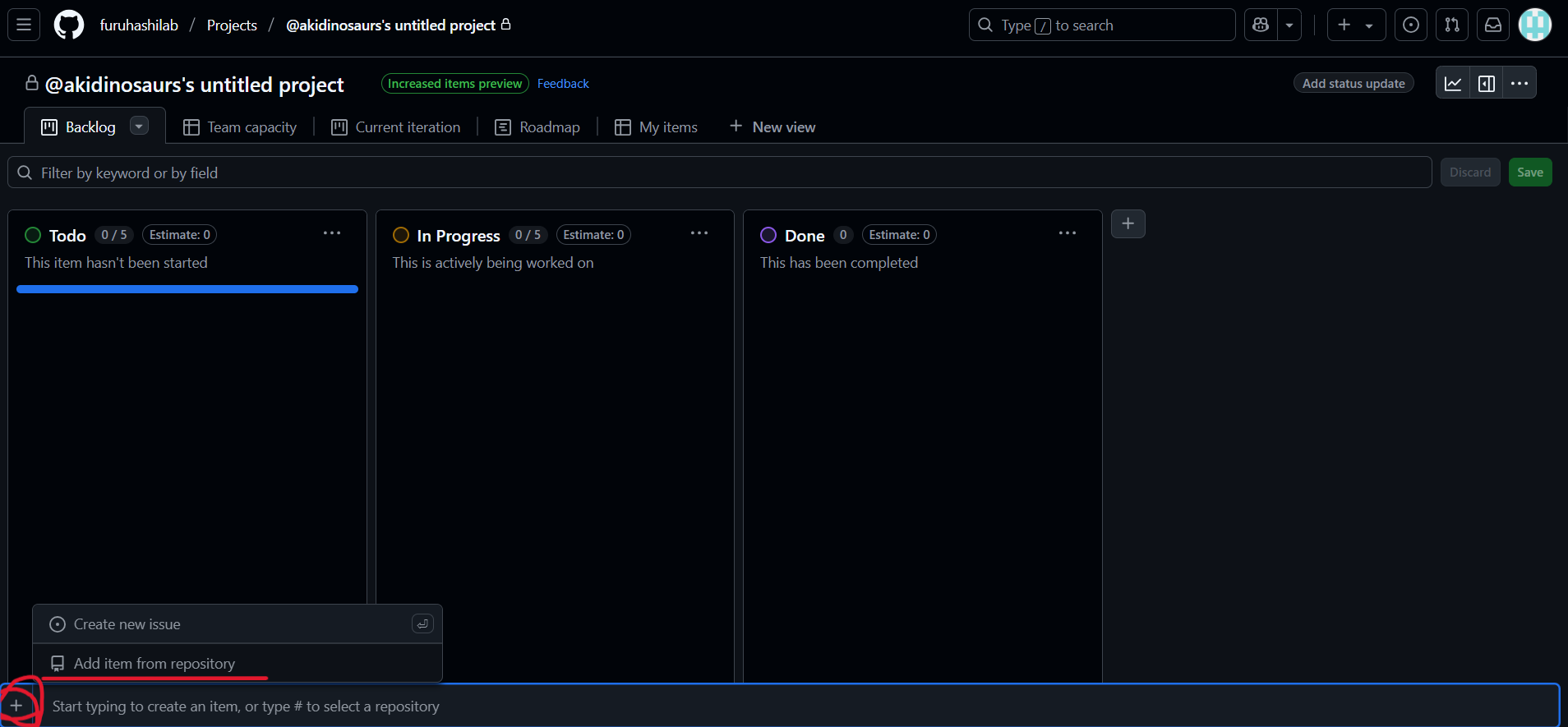
  


### 8.3 IssueをProjectに追加する方法

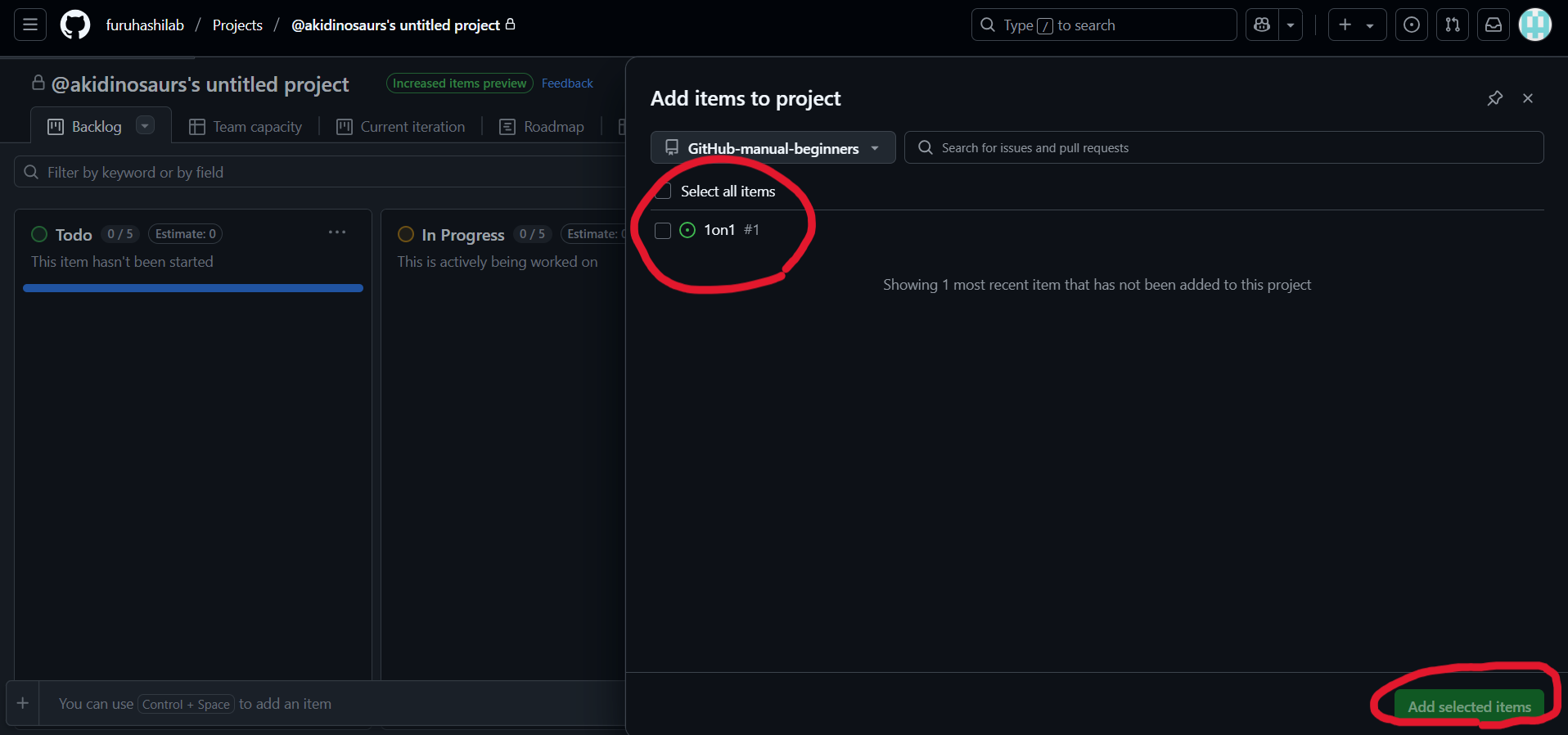
* 作成済みのIssueをドラッグしてボード上に追加できます。
* 「＋」マークから新しいIssueを作成して追加することも可能です。

※以下に、「∔」マークから既存のissueとタスクをリンクさせる方法を載せます。

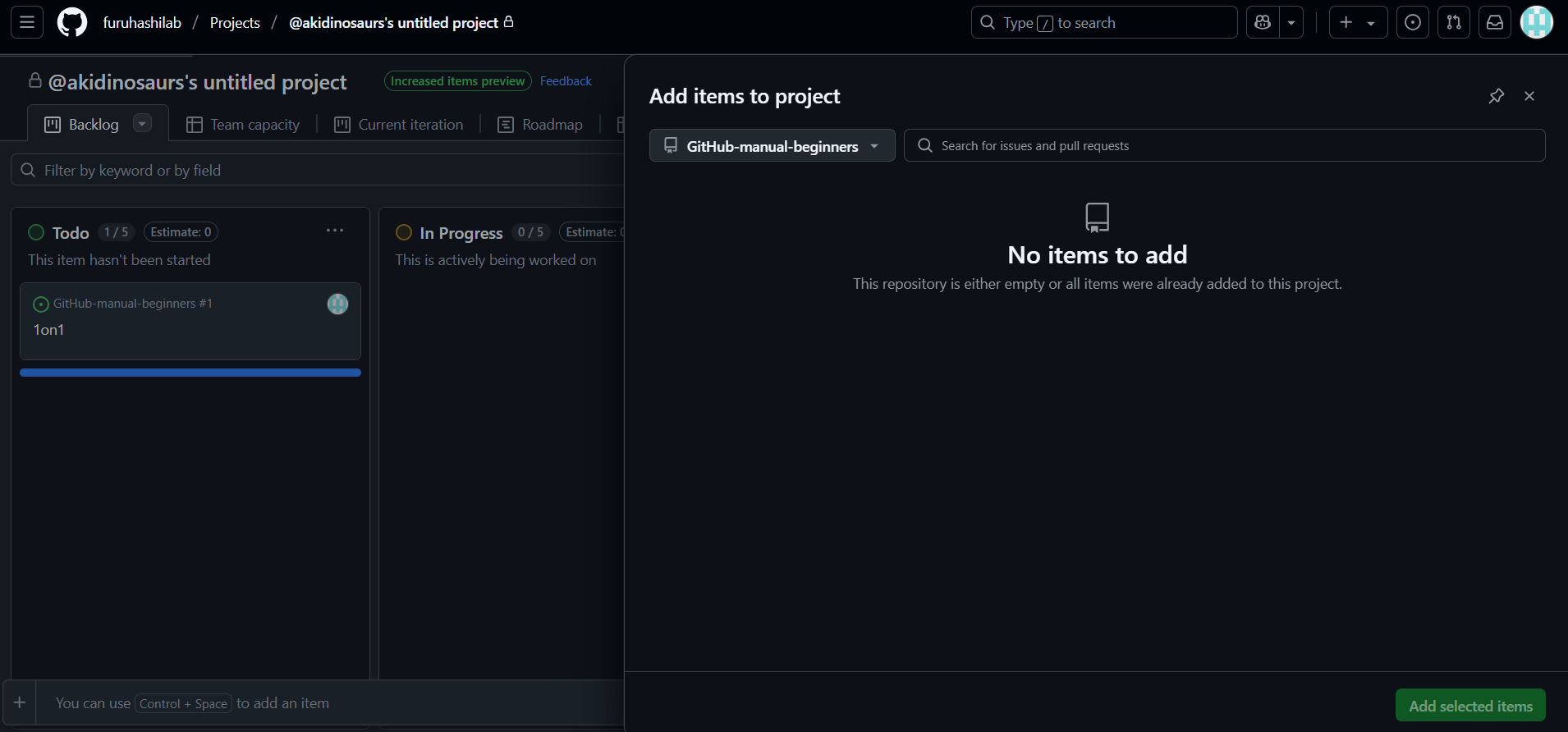
↓「+」を押して、「Add issue from repository」を選択します。



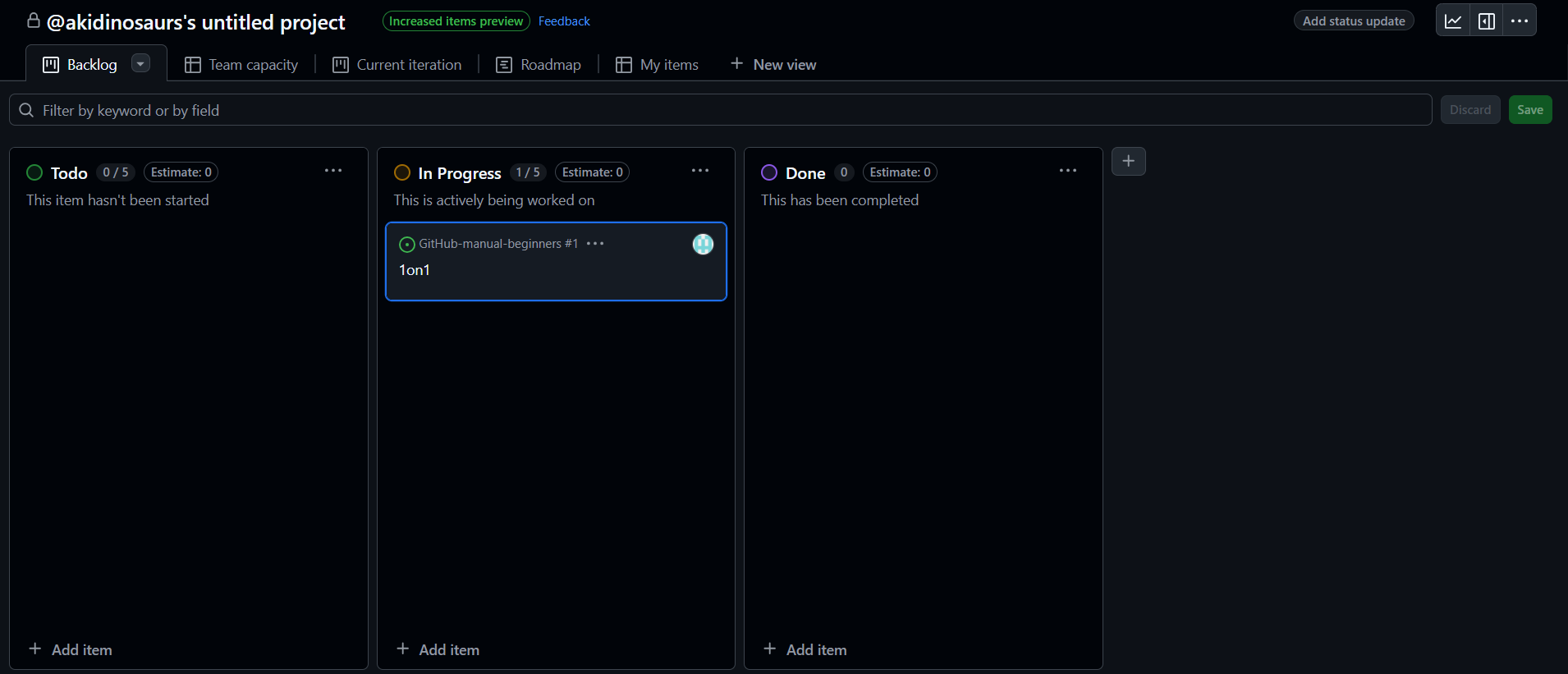
既存のissueを選択し、「Add selected items」。

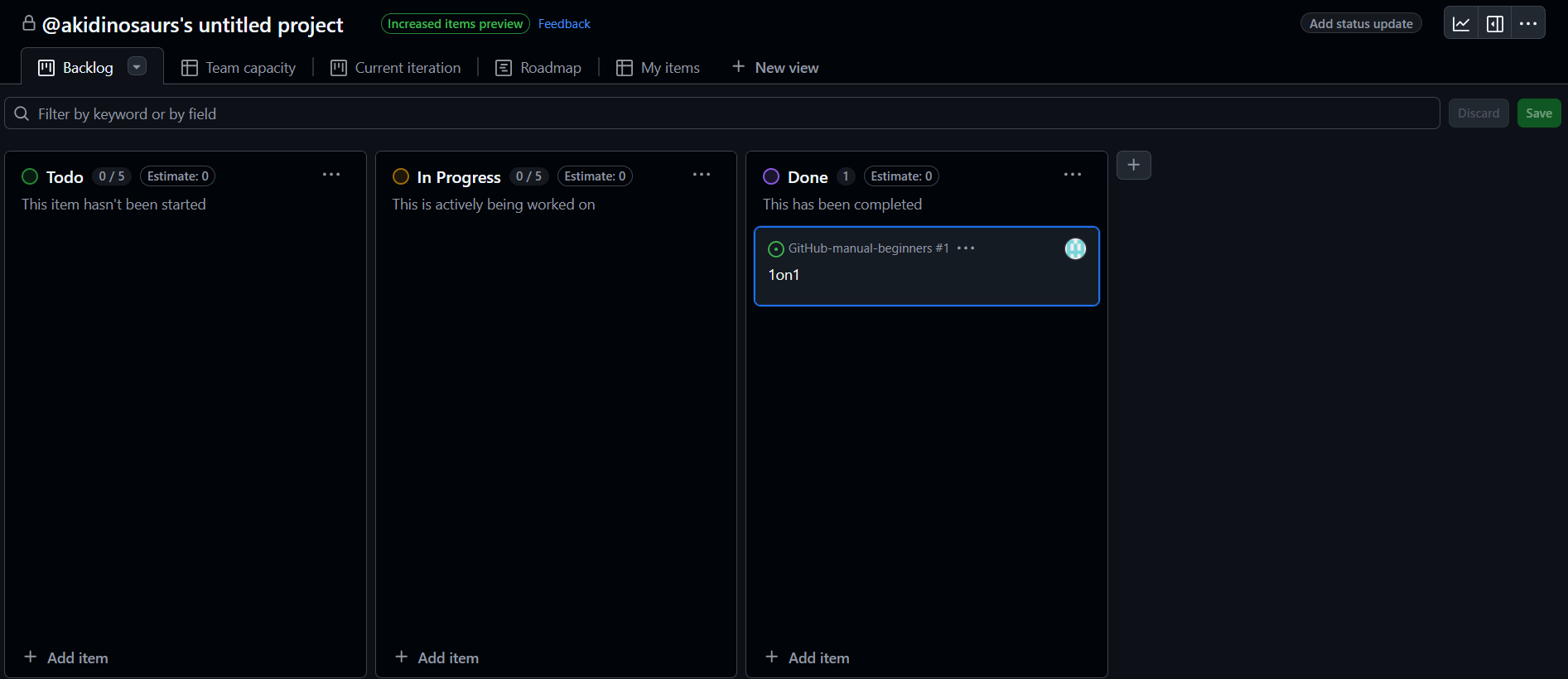


↓これで、issueがtodoタスクに追加されました。



タスクは、進捗に応じて、ドラッグ＆ドロップで「todo」「In Progress」「Done」に分けることができます。





## 9. 通知と共有（おまけ）

* **通知の設定**：右上のプロフィールアイコン →「Settings」→「Notifications」から、メール通知のON/OFFが設定できます。
* **他人との共有**：PublicリポジトリならURLを送るだけでOKです。
* **共同作業者の追加（Privateの場合）**：リポジトリ内の「Settings」→「Collaborators」から追加します。

## まとめ

以上が、GitHubの基本的な操作方法の説明になります！他にも、GitHubにはさまざまな機能がありますが、基本操作以外は必要になったら覚えるくらいで大丈夫です。

## 追記：データを削除しないように注意してください！

リポジトリやファイルのデータを誤って削除してしまう場合もあるので、操作が不安であれば、「Delete」ボタンは押さない方が安全です！