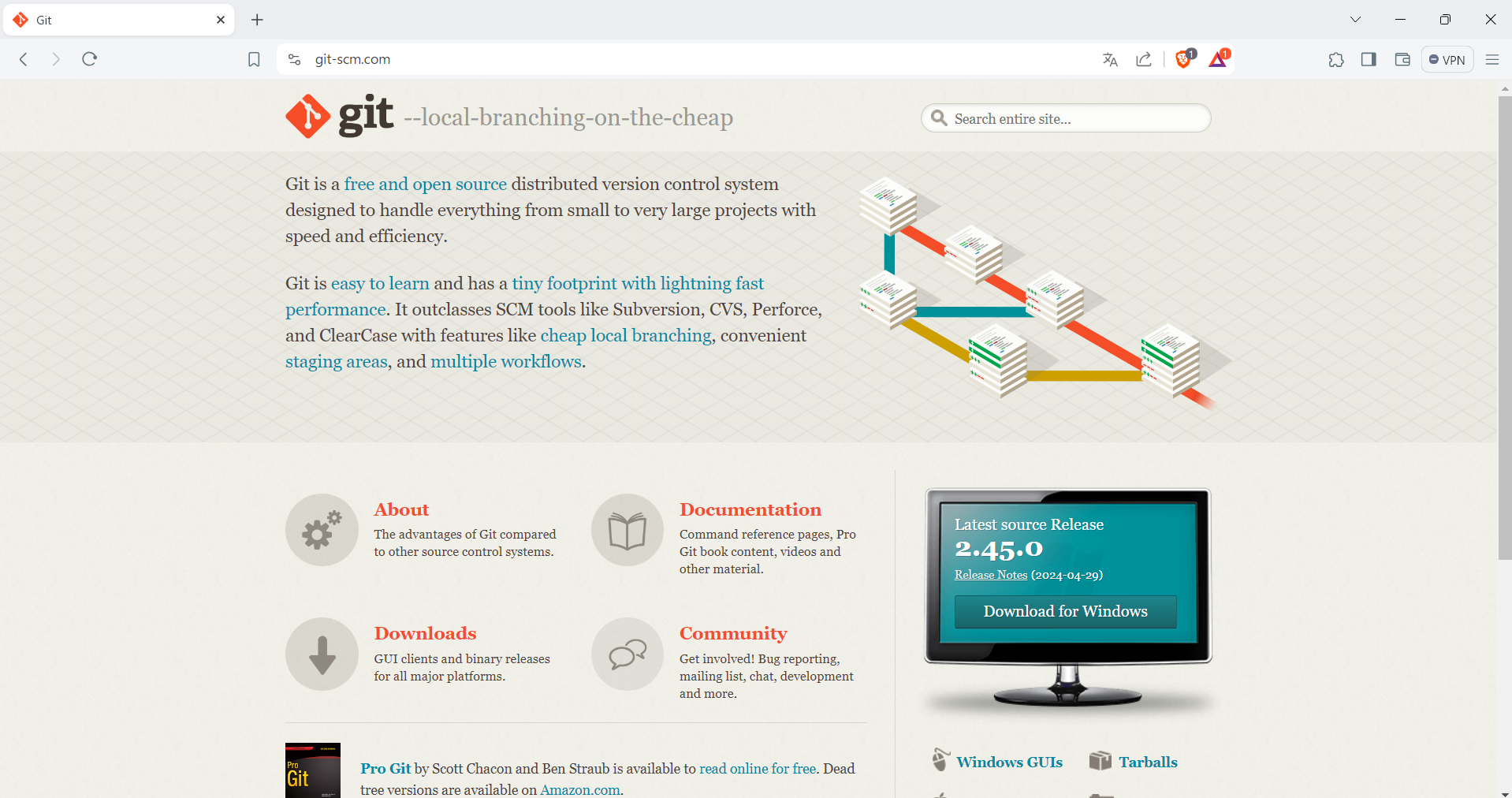
Git BashとGitHubを使用して基本的なGit操作（クローン、マージ、プル、プッシュ）を学ぶためには、通常、完全な初心者であれば数日から1週間程度かかることが一般的です。これは毎日少しずつ学び、実際に手を動かして練習することを想定しています。

以下に基本的な手順を書いていきます。

0. Git BashとGitHubのセットアップ(済)

1. \*\*Gitのインストール：\*\* 公式サイト（https://git-scm.com/）からGitをダウンロードしてインストールします。インストール中に「Git Bash」を選択してください。



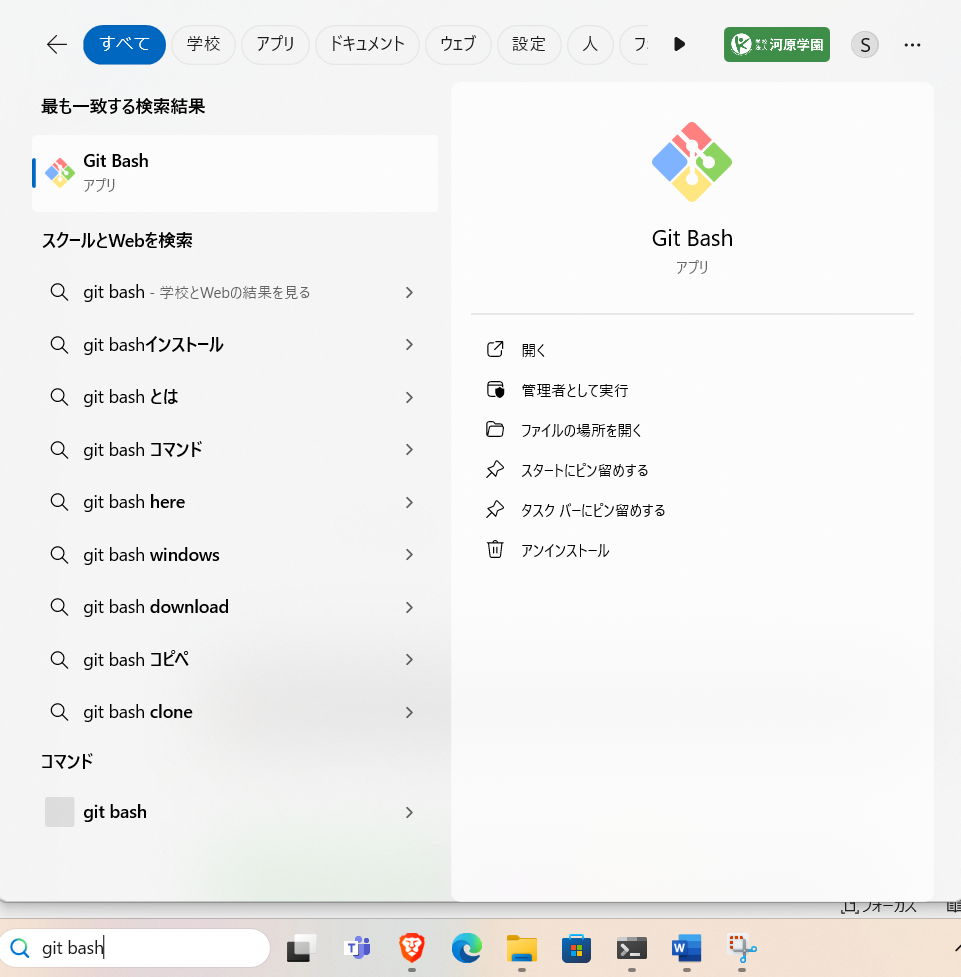
2. \*\*GitHubアカウントの作成：\*\* GitHub（https://github.com/）にアクセスし、アカウントを作成します。



1. デスクトップディレクトリへの移動

まずは、Git Bashを開きます。画面左下の虫眼鏡のマークをクリックして、

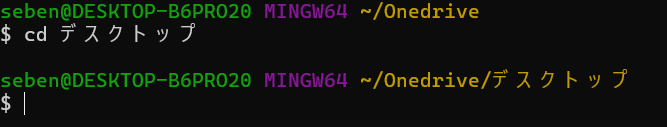
「Git Bash」というアプリを選びましょう。



git bash の画面を表示出来たら、以下のコマンドを入力してください。



入力ができたら、このような画面になるはずです。

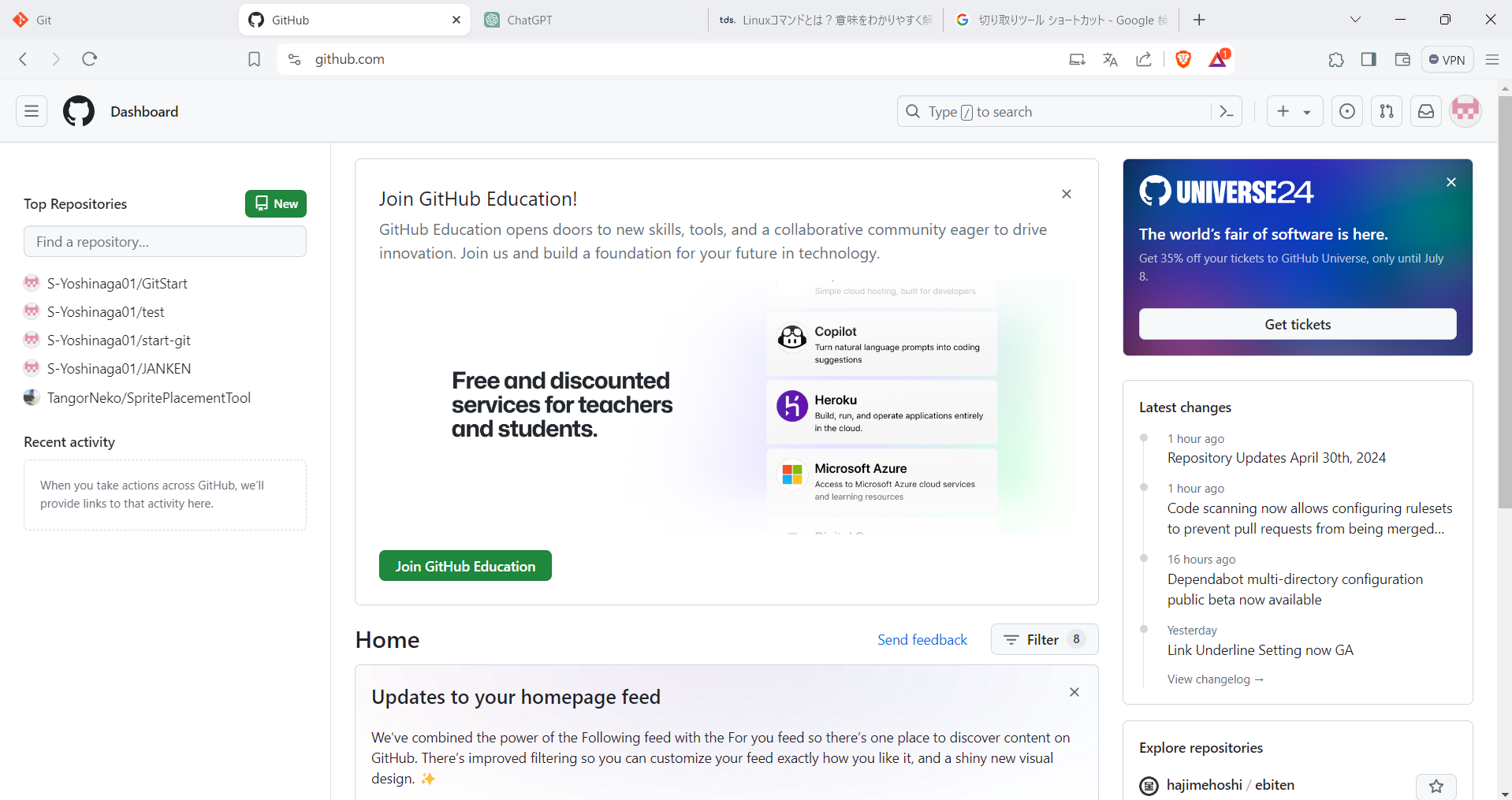


(私の環境では、デスクトップがカタカナでしたが、皆さんはDesktopだと思います)

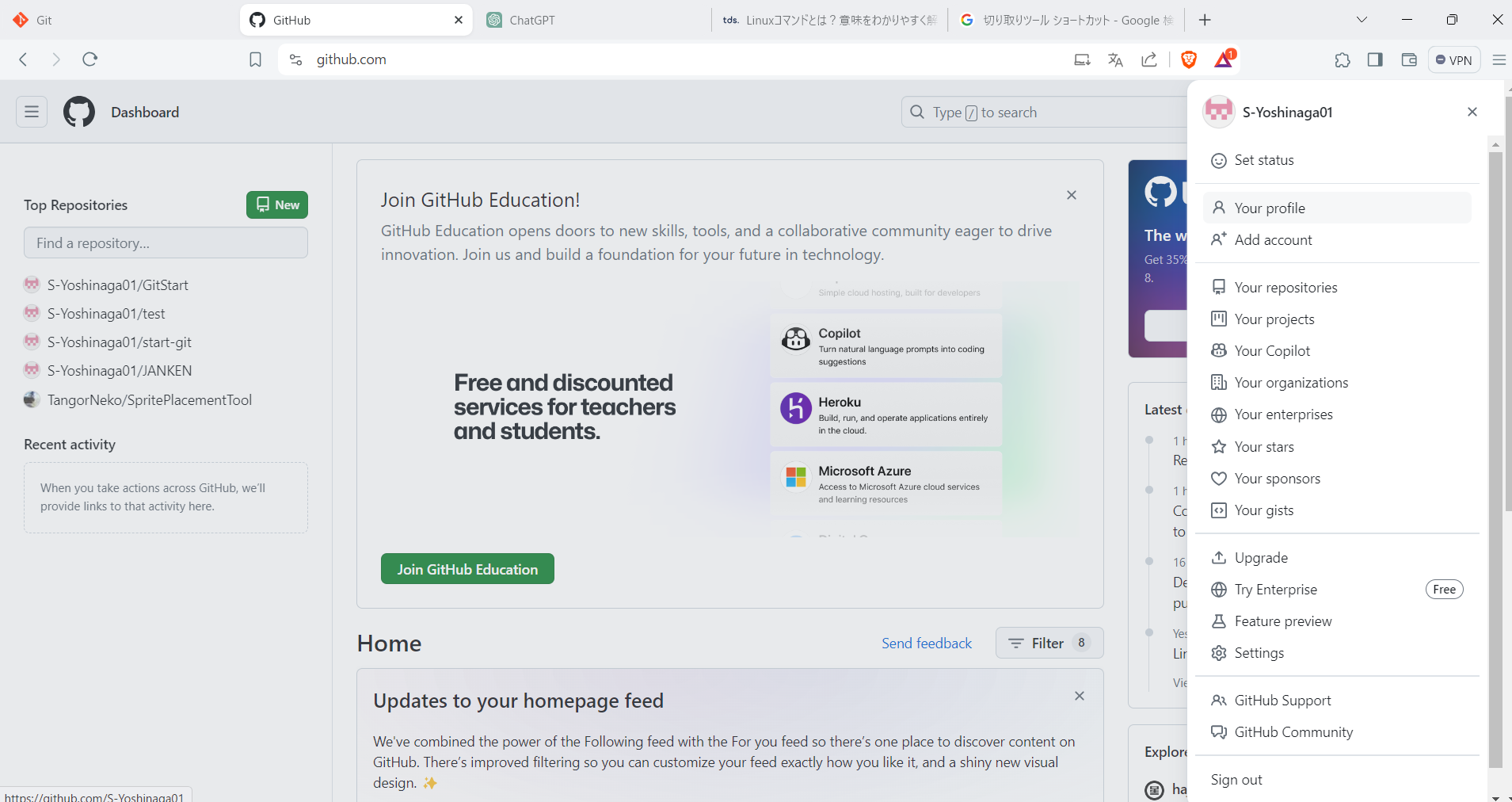
黄色でdesktop　と表示されたら成功です。

1. リポジトリのクローン

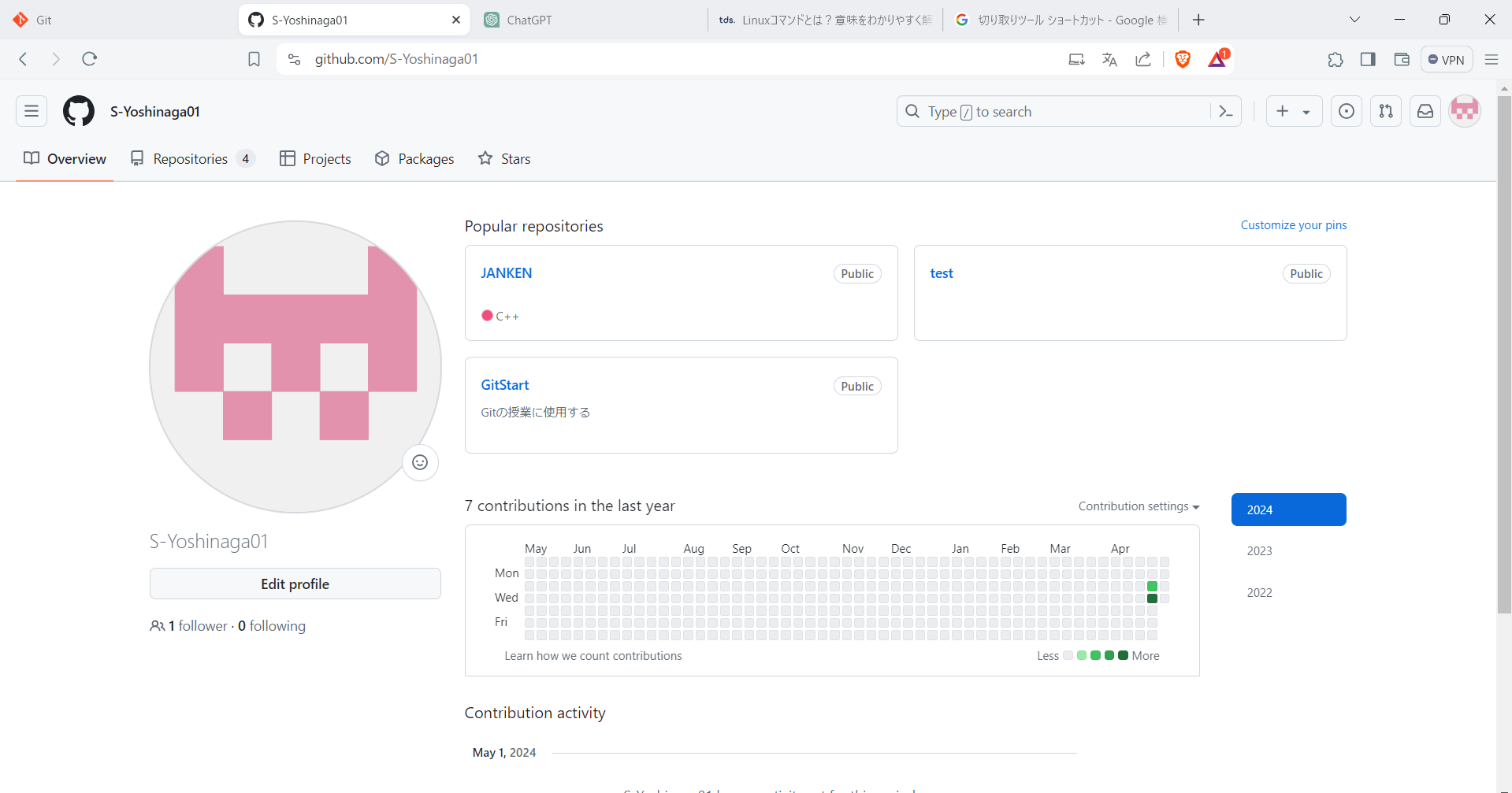
まず、Githubを開きましょう



このような画面が表示されたら、右上のドット絵アイコンをクリックします。

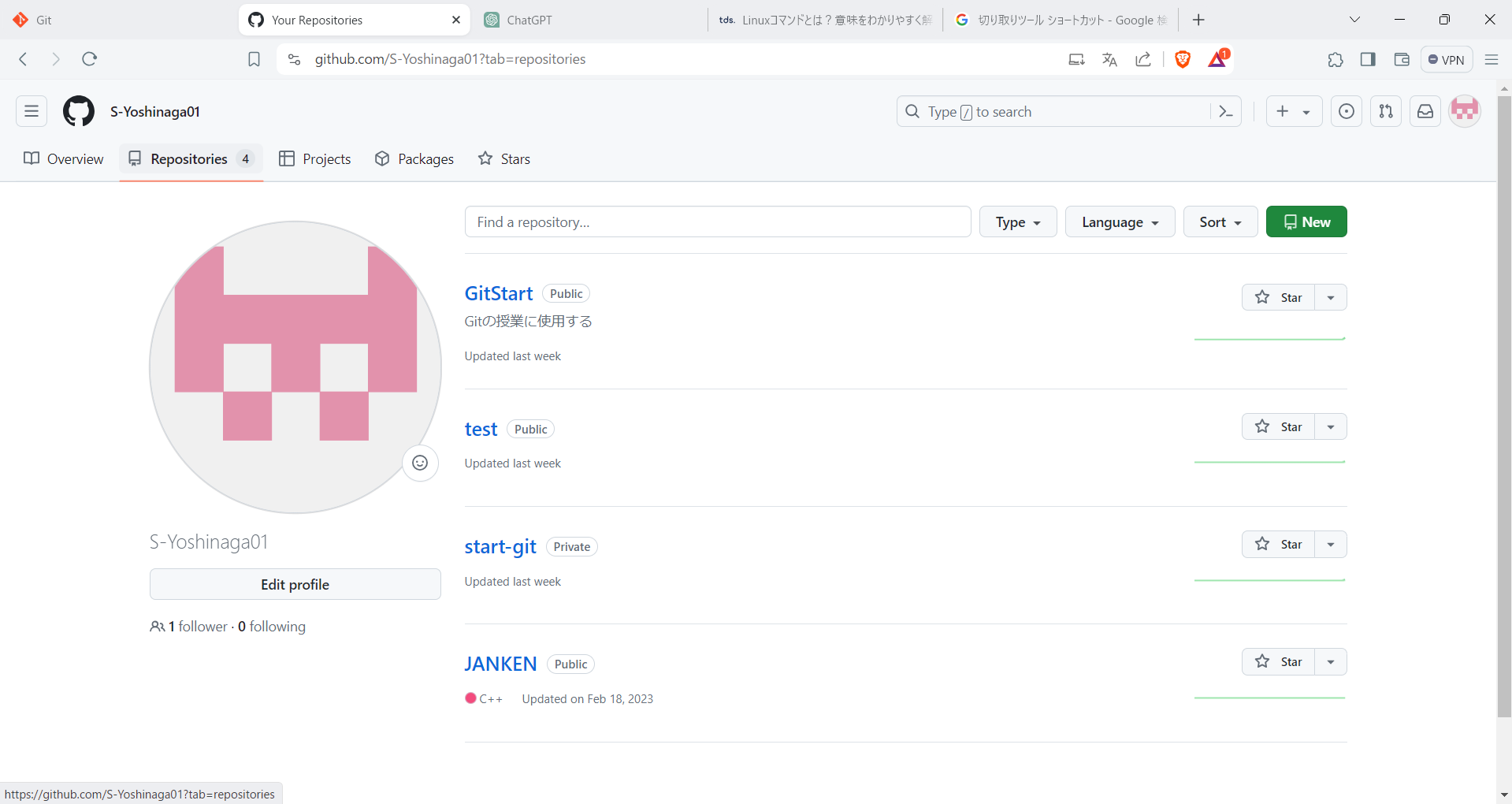


クリックしたら、Your profile をクリック。



このような画面になると思います。

この画面から、Repositoriesをクリックしてみましょう。

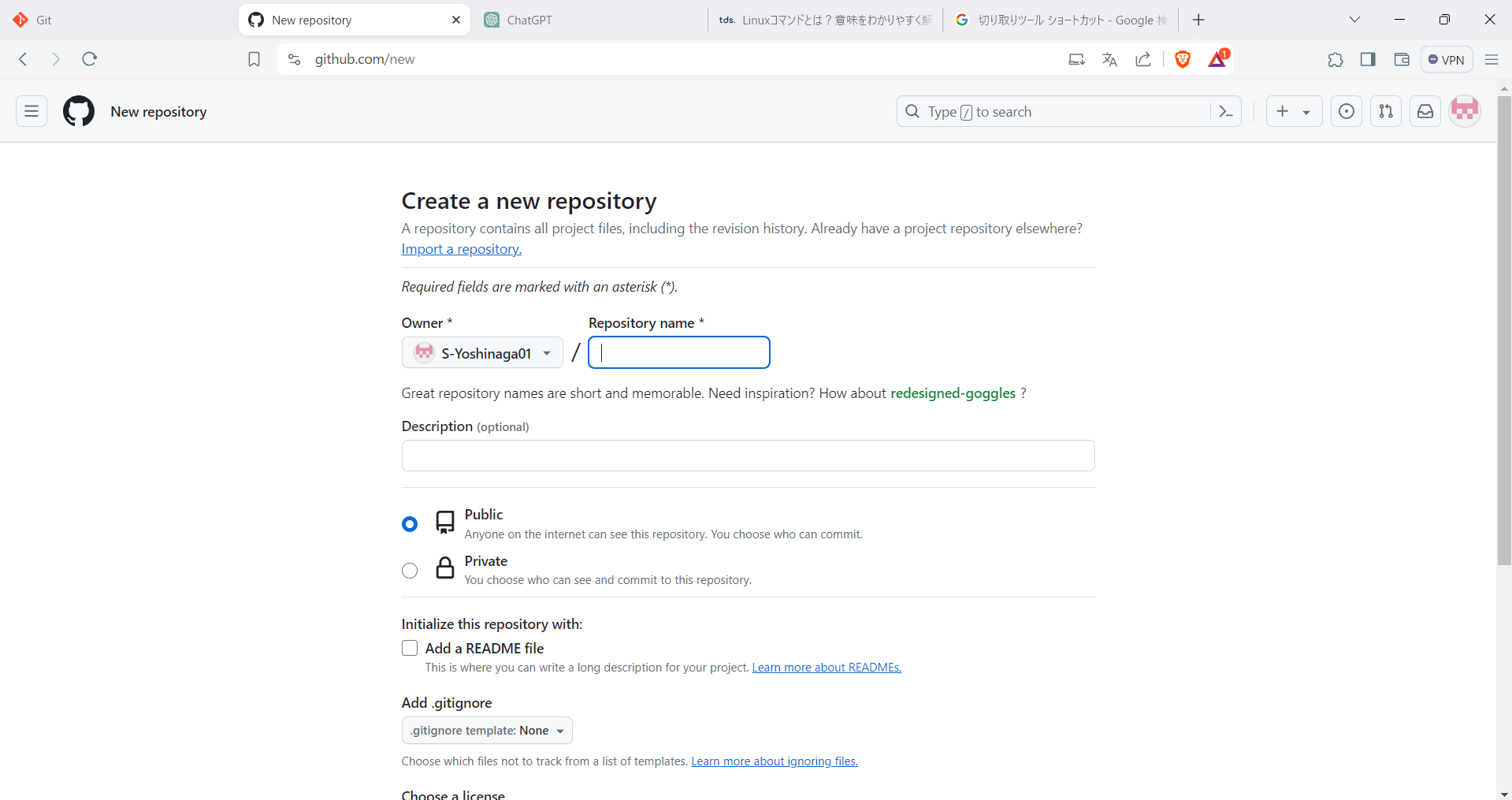


ここをクリックすると、今まで作ってきたリポジトリが確認できるようになります。

リポジトリとは、データやコードなどの資料を保存しておくための場所です。

ゲームやITなどのソフトウェア開発では、プロジェクトのソースコードを管理するために使われることが多いです。

これが確認できた人は、右上の緑のボタン、 New　をクリックしましょう。



すると、このような画面になります。

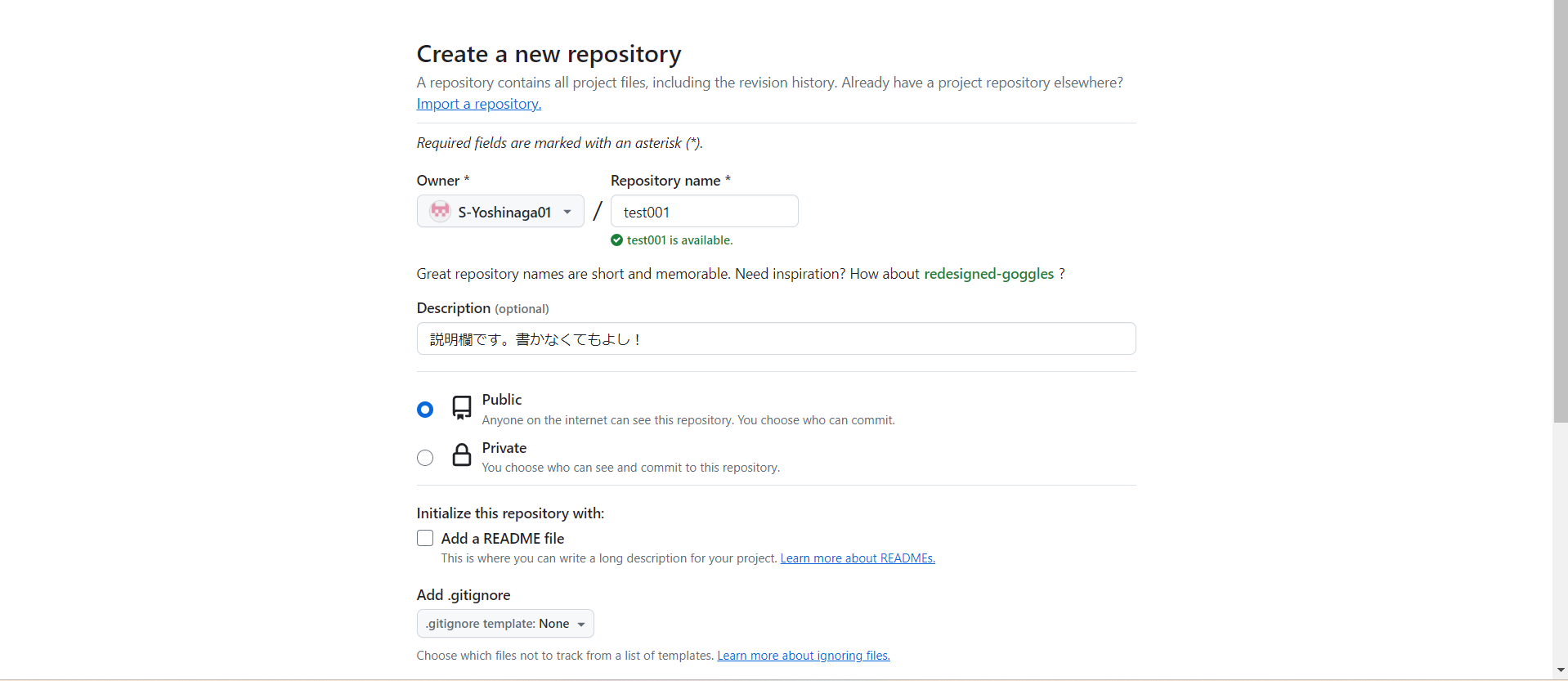
ここでは、新しくリポジトリを作る設定を選択します。

まずは、このリポジトリの名前を設定してやりましょう。

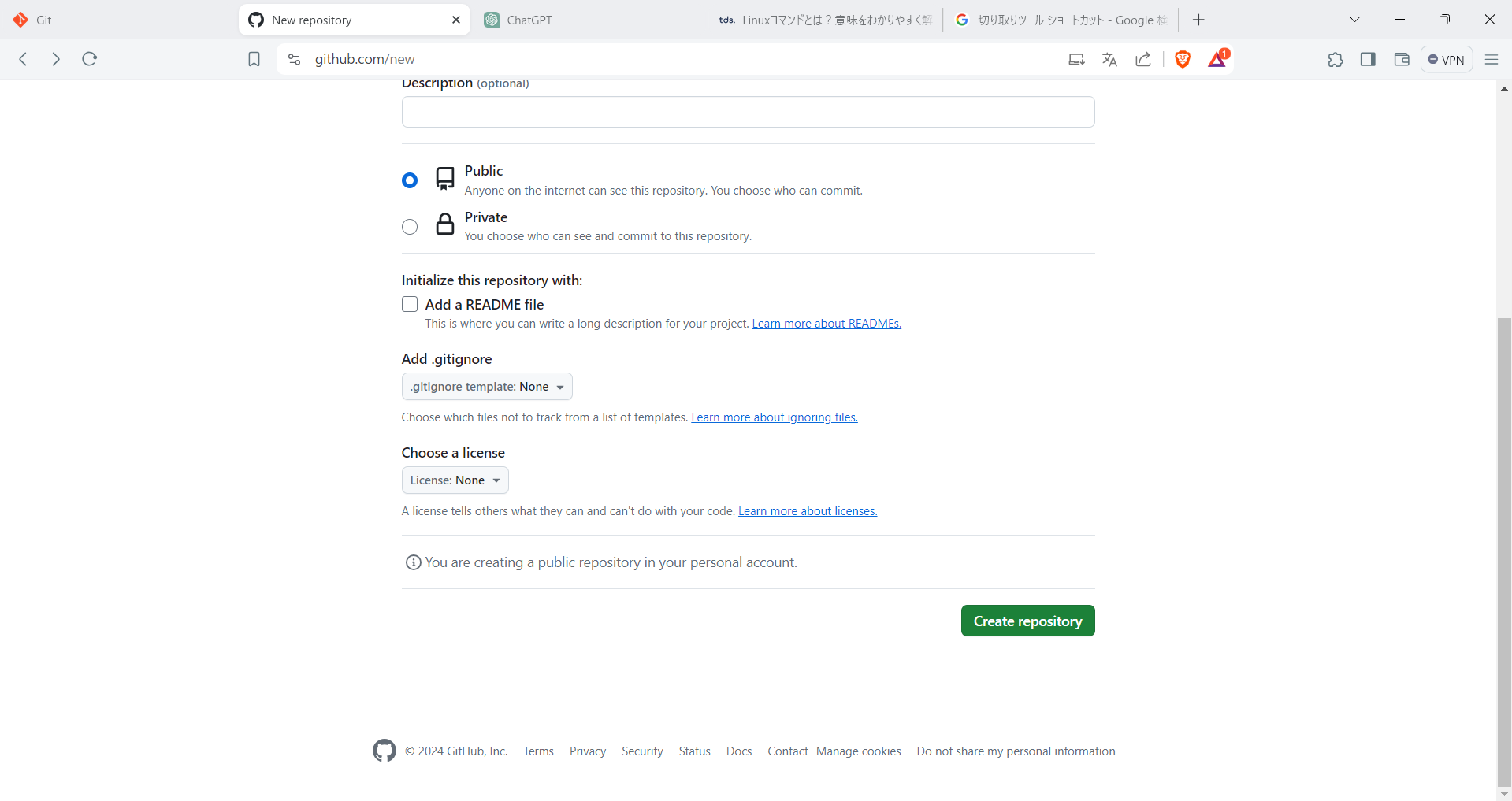
Repository name と書かれている場所に、test001と入力してみてください。

(Repository nameはどんな名前でもいいですが、全世界に公開されますので

公開してもいい名前にしておいてください。ネットリテラシー大事。)



ここまで書けたら、下にスクロールして、Create repositoryをクリックします。



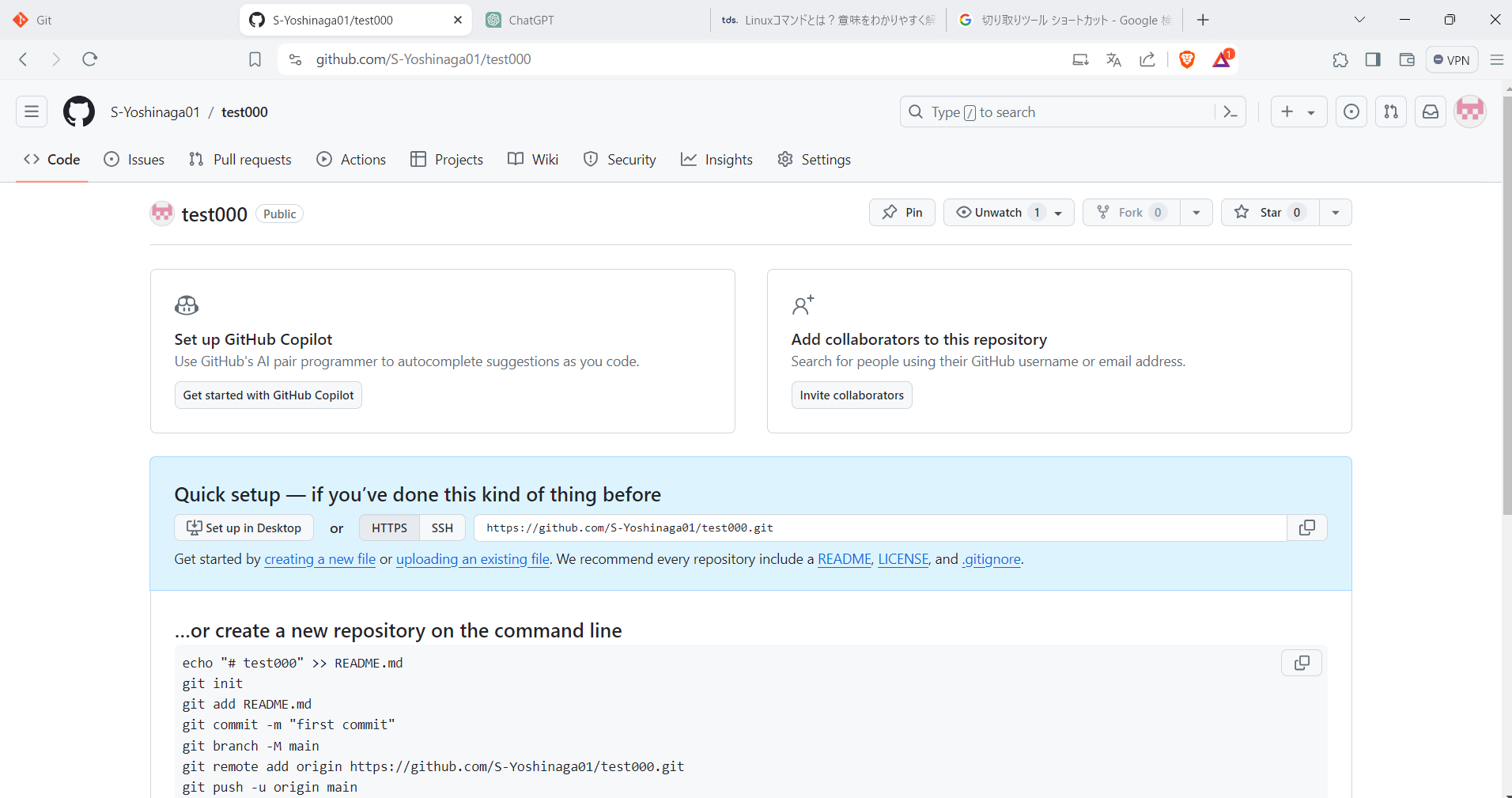
このような画面ができたら、リポジトリが完成します。

先ほど、リポジトリとはデータやコードなどの資料を保存しておくための場所　と

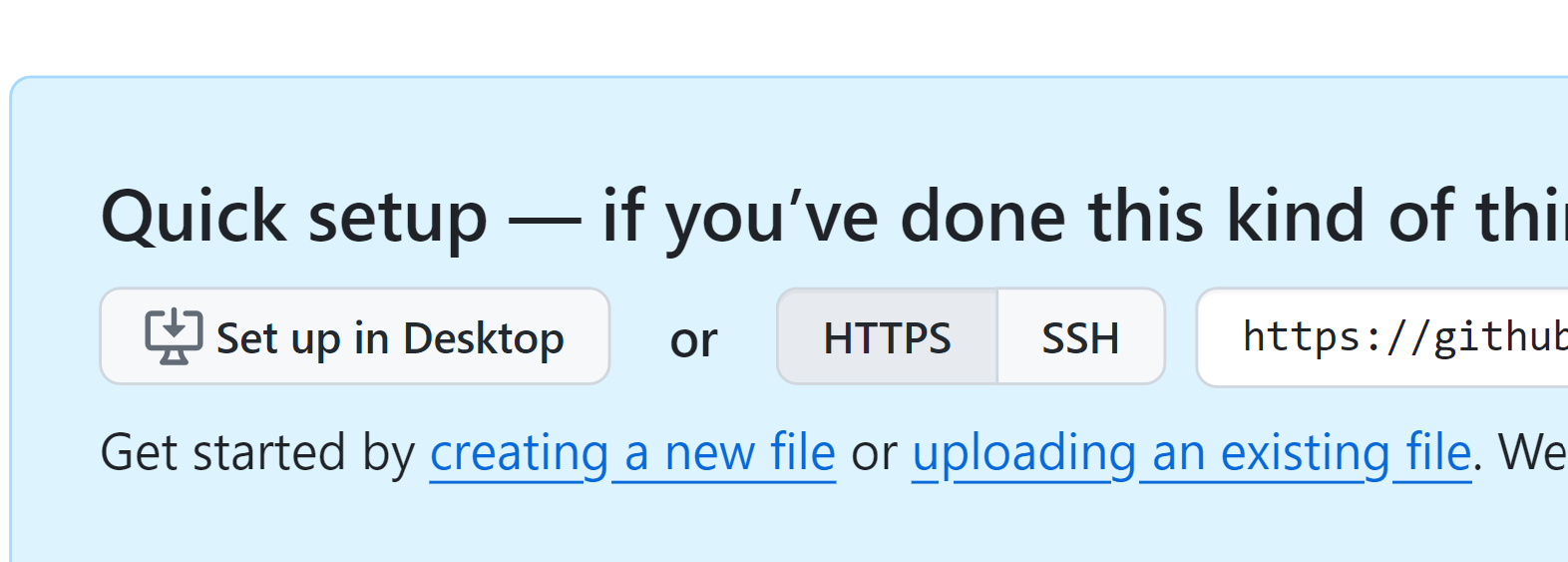
説明しましたが、この状態は、何のコードも入れていないまっさらな状態です。

試しに、このリポジトリに新しいファイルを作ってみましょう。

わかりづらいですが、画面左中央の creating a new file をクリックしましょう。



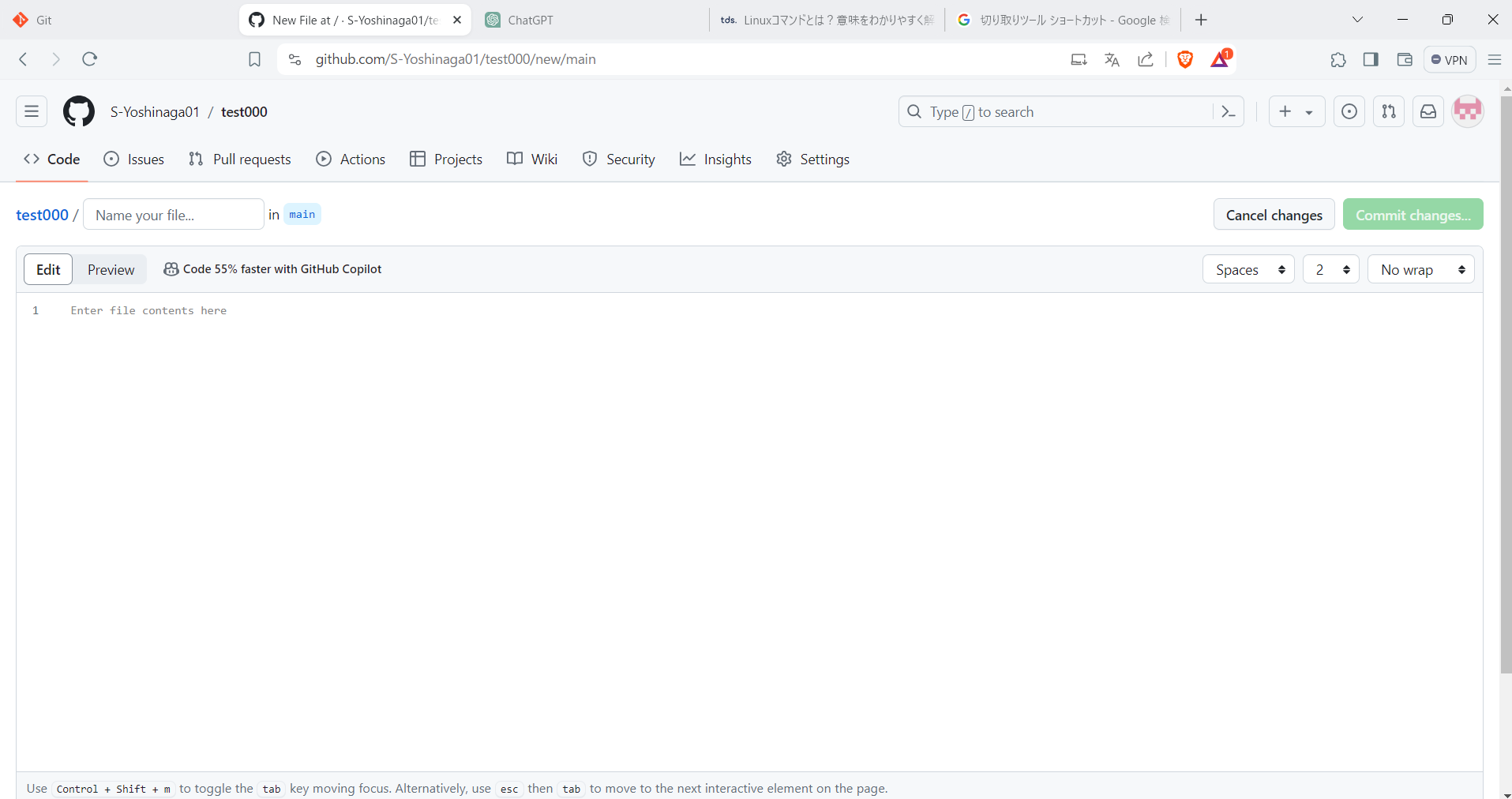
拡大画面です。　一応　　HTTPS になっていることを確認してください。



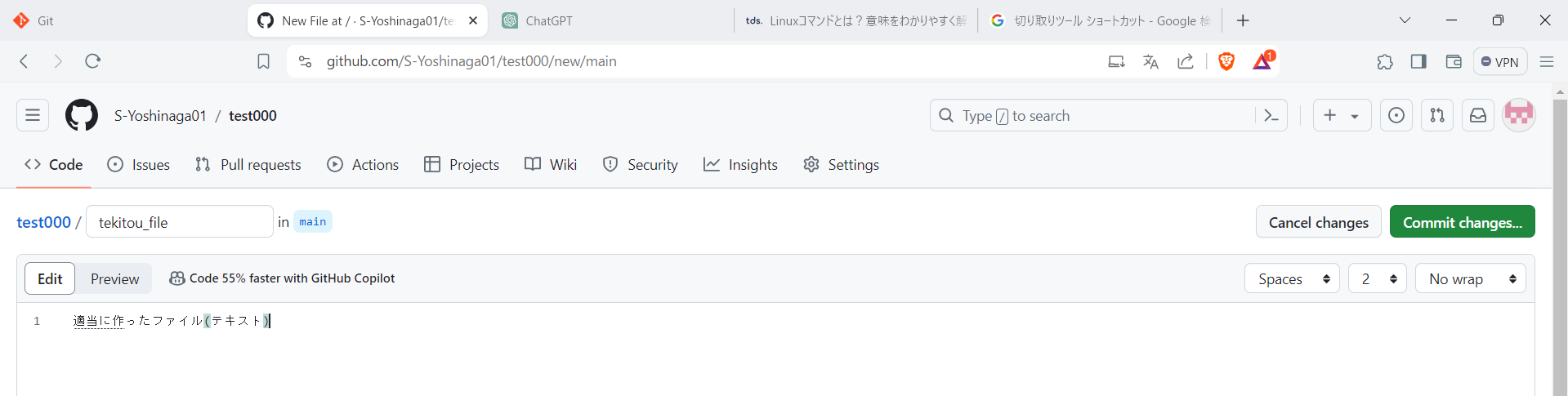
クリックすると、Github上での新規ファイルの作成画面に入ります。

新規ファイルを作るには、ファイル名とファイル内容が必要になります。

早速書いていきましょう



【参考例】↓こんな感じです。↓



ファイル名と、何かしらファイルの内容を書けたら、

右上にあるcommit changes を押しましょう。

Commit とは「確定」という意味で、commit changesは

「ファイルの変更を確定する」という意味のボタンになります。

Commit massage　を入力する画面がありますが、

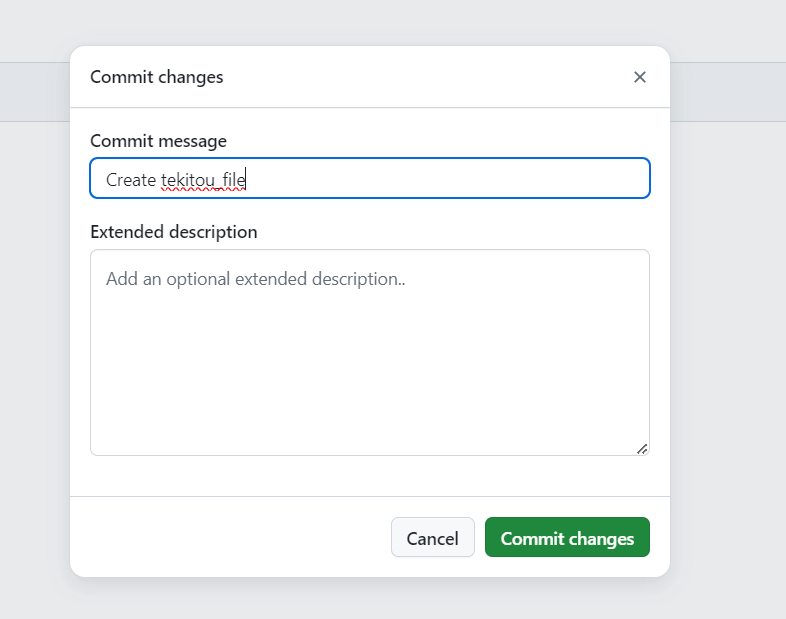
ここは変更内容を記載するときに書くメッセージになります。

「初めてのコミット」とか、「new file create」とか意味がある文章を

書くのがいいです。(今回、参考例ではスルーしています。)

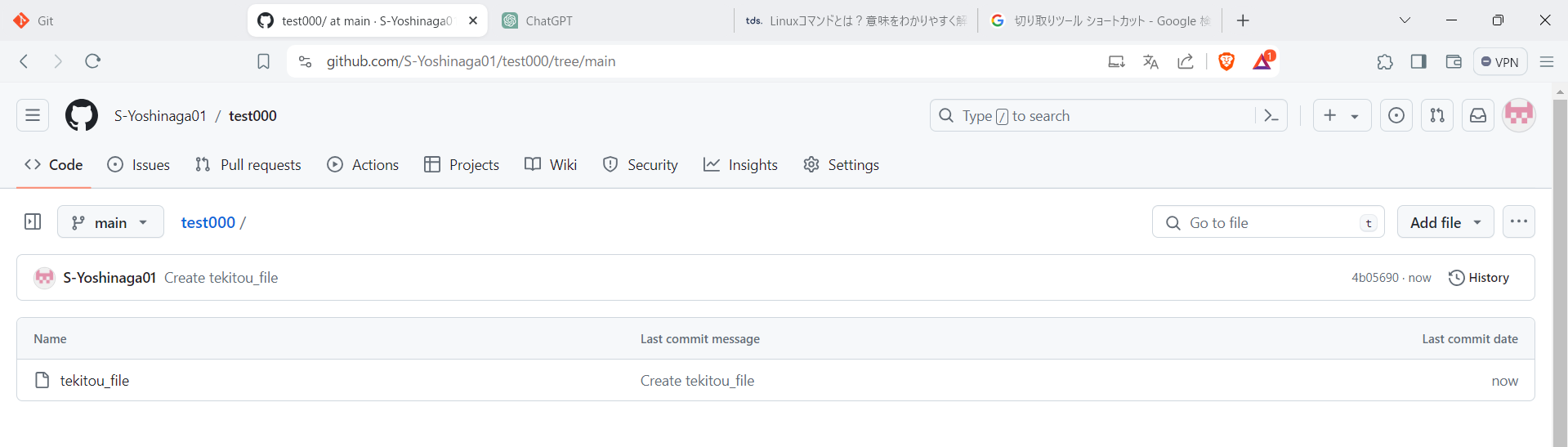
このメッセージを書くことで、他の人に変更内容を簡潔に見せることができます。

文章が書けたら、右下のCommit changesを押しましょう。



これを押すと、先ほど作ったリポジトリの画面に

これまた先ほど作ったfileが入力されているのが確認できたでしょうか。



この画面になったら一度、 F5を押してwebページを再表示してみましょう。

右上にCode の文字が見えるようになったと思います。ここをクリックします。

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション, メール

自動的に生成された説明

このボタンは、現在Githubに上がっているリポジトリ(コードの集合)を実際に

ダウンロードできるボタンになっています。

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション, メール

自動的に生成された説明

押すとこんな感じの画面になります。

この画面になったら、右にあるコピーのマークを押してください。

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション, メール

自動的に生成された説明

こんな感じで、チェックマークが出たら成功です。

Github側の作業は以上になります。続いて、GitBash側の作業に行きましょう。

Git Bashを開き、以下のコマンドを実行します。

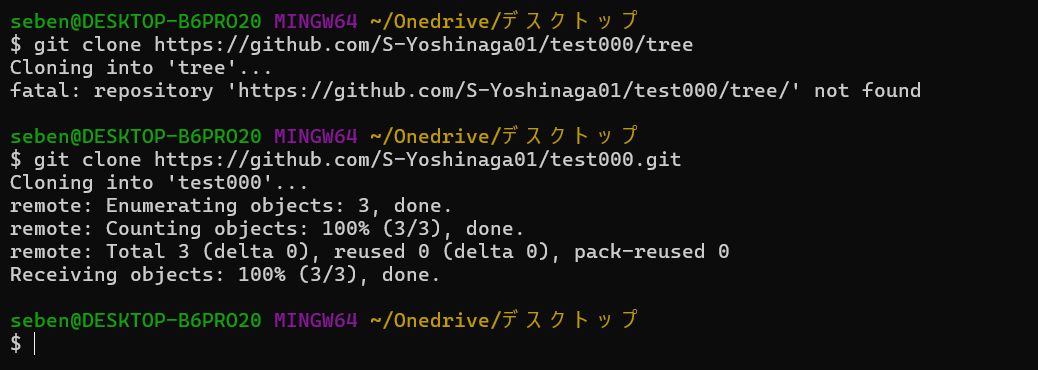


ここでは、自分がgithubで作成したリポジトリにURLを、例のように入れてみましょう。

ここでは　git clone 　というコマンドの後に

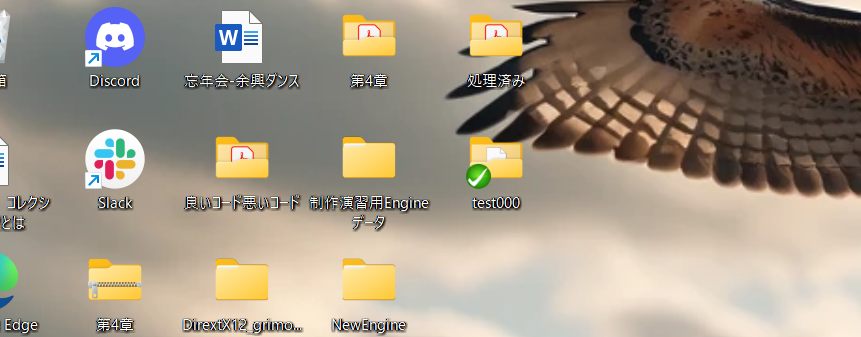
https://github.com/S-Yoshinaga01/test000.gitを入力しています。

正常な挙動が起きると、水色の枠のような画面が出てきます。



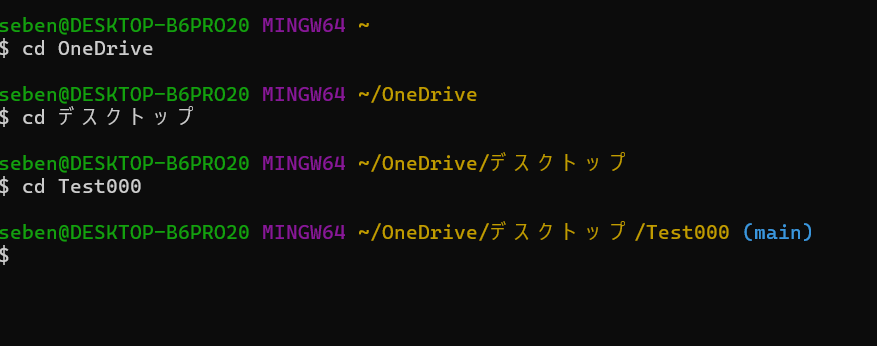
ここで、一度Gitbashを最小化して、デスクトップを見てみましょう。

先ほど作ったファイル「test000」が作られていることが確認できたら大成功です。



ここまで非常に長い手順でしたが、ここからがようやくGitの話になります。

もう一度Gitbashを拡大して、cd test000に入ってみましょう。 (main)が出てたらokです。



3. プル（Pull）

リモートリポジトリの変更をローカルリポジトリに取り込みます。

1. Git Bashでクローンしたリポジトリのディレクトリに移動します。

2. 以下のコマンドを実行します。



main

1. プッシュ（Push）

これを行う前に、まずデスクトップに追加されたファイルに入ります。

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション

自動的に生成された説明

そして、白いところを右クリック

→新規作成→テキストドキュメントで適当な.txtファイルを作ってください。

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション, Word

自動的に生成された説明

この操作が終わった後に、下記のGit Bashのコードを入力してください。

ローカルの変更をリモートリポジトリに送信します。

1. 変更をステージングエリアに追加します。



1. 変更にコミットします。



3. リモートリポジトリにプッシュします。



これが打ち込めたら、Git hubにアクセスしてみて下さい。

先ほど追加したファイルが、push(押し出す)され、Git hub上に登録されます。

5. マージ（Merge）(ここは今日はやりません。)

異なるブランチの変更を現在のブランチに統合します。

1. マージしたいブランチに切り替えます。



2. マージするブランチを指定します。



### 練習のポイント

毎日少しずつ実践することが重要です。実際に手を動かしてみることで、理解が深まります。

エラーメッセージに慣れることも重要です。エラーが発生した場合は、その内容を読んで理解し、解決策を探してみてください。