
付録

Visual Studio での Git の使い方

VisualStudio の利用

ここでは Microsoft 社の Visual Studio を使って、GUI ベースで Git を操作する方法を紹介합니다。

まずは、無償で利用できる Visual Studio Community 版（Windows）のインストール手順について解説します。

ダウンロードとインストール

ダウンロード

下記 URL にアクセスすると、最新版をダウンロードできます。

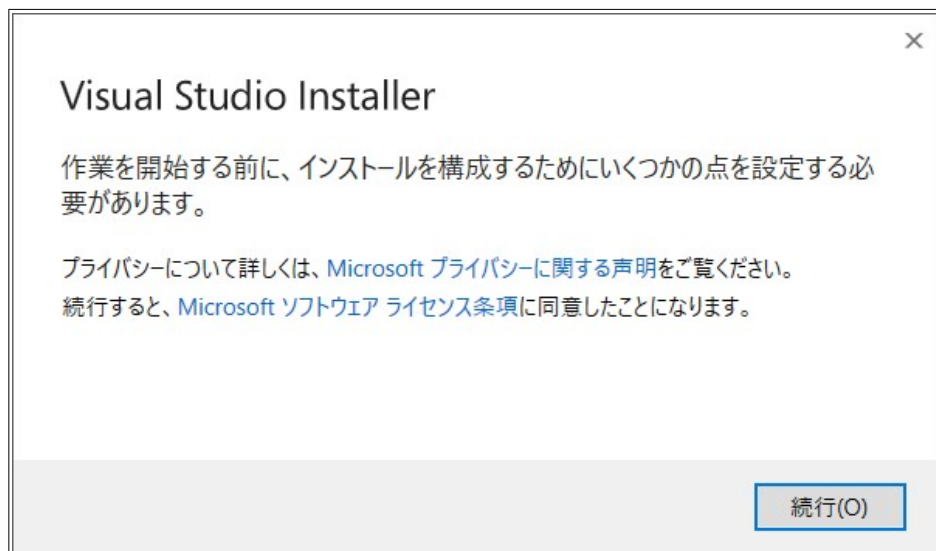
<https://visualstudio.microsoft.com/ja/vs/community/>



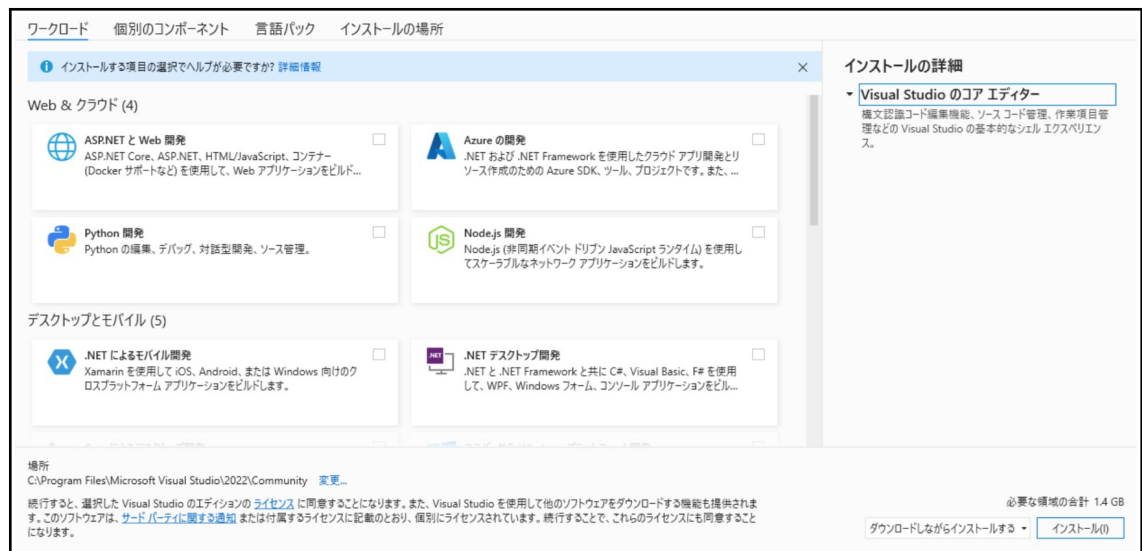
アクセスしたページの「Visual Studio のダウンロード」をクリックすると自動的にダウンロードが開始されます。任意のフォルダ内に保存するようにして下さい。

インストール

ダウンロードが完了したら、ダブルクリックしてインストーラを起動して下さい。続行を押します。



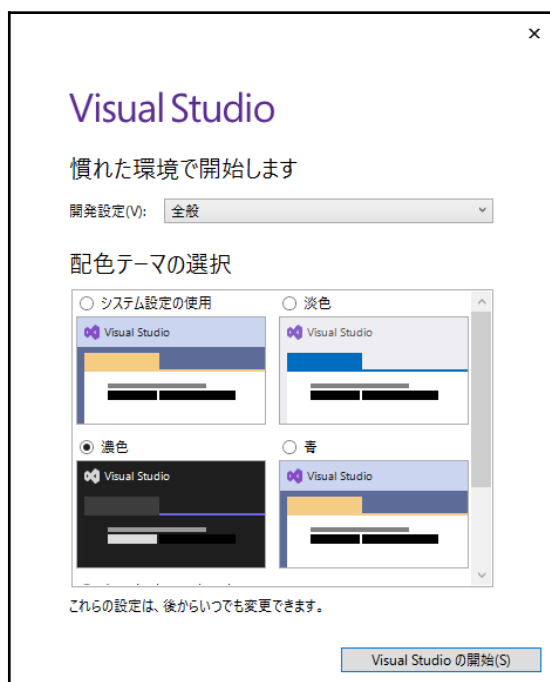
ワークロードは必要に応じて任意のものにチェックを入れて、右下の「インストール」ボタンをクリックして下さい（本書では何もチェックを入れません）。



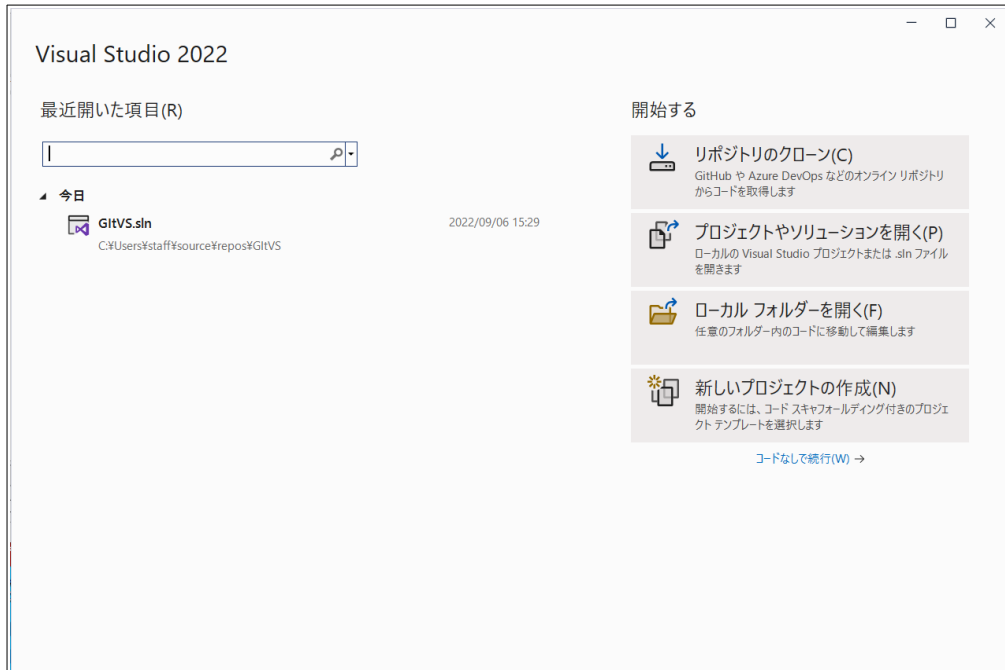
インストールが完了すると Visual Studio へのサインインを促されますが「後で行う」を選んで次へ進んで下さい。



「開発設定」や「配色テーマ」は自由に選んで構いません。右下の「Visual Studio の開始」をクリックします。



ここでは右下の「新しいプロジェクトの作成」を選びます。



ここでは「空のソリューション」を選んで右下の「次へ」をクリックします。



次の画面ではソリューションの名前を決めます。ここでは「GitVS」として右下の「作成」をクリックします。



新しいプロジェクトを構成します

空のソリューション その他

ソリューション名(N)

GitVS

場所(L)

C:\Users\staff\source\repos

...

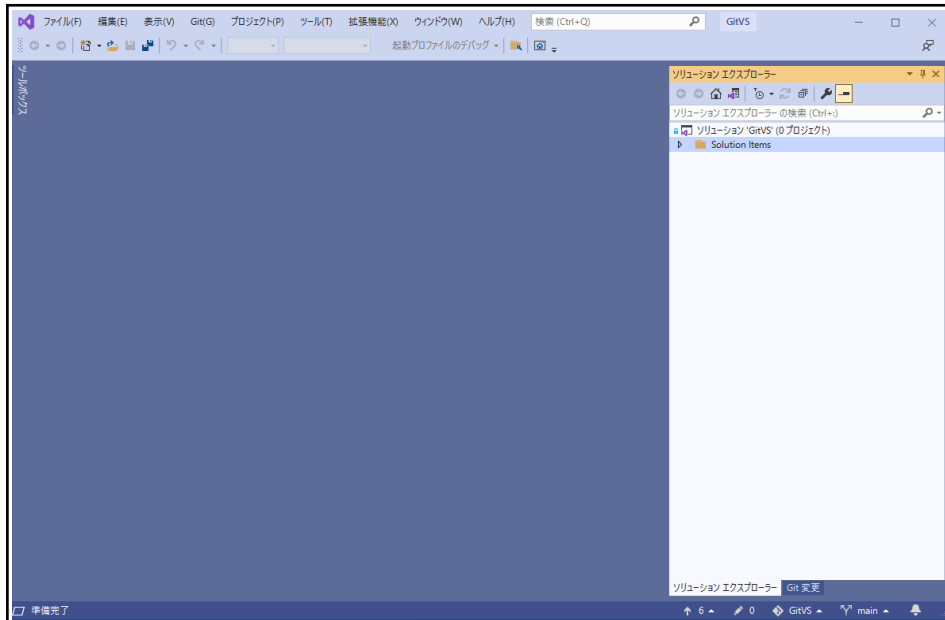
戻る(B) 作成(C)

これで使用準備が完了し、IDE が起動します。

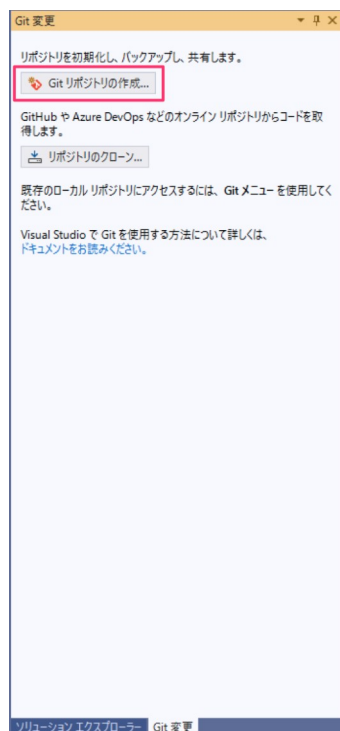
Visual Studio で Git の操作

Git リポジトリの作成

Visual Studio が起動すると、右側に「ソリューションエクスプローラ」があります。その下側のタブに「Git 変更」があるので、そこをクリックします。



タブが切り替わったら、「Git リポジトリの作成」ボタンをクリックします。

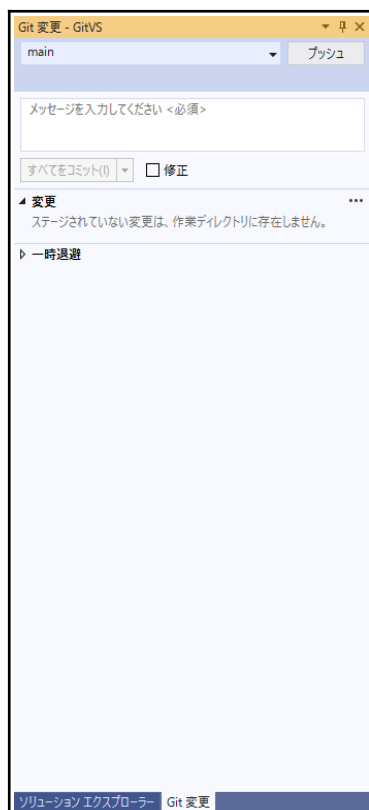


ローカルリポジトリの作成

今回はリモートリポジトリを使わないので「ローカルのみ」を選択します。ローカルパスも特に変更する必要はありません。下図の状態で「作成」ボタンをクリックして下さい。

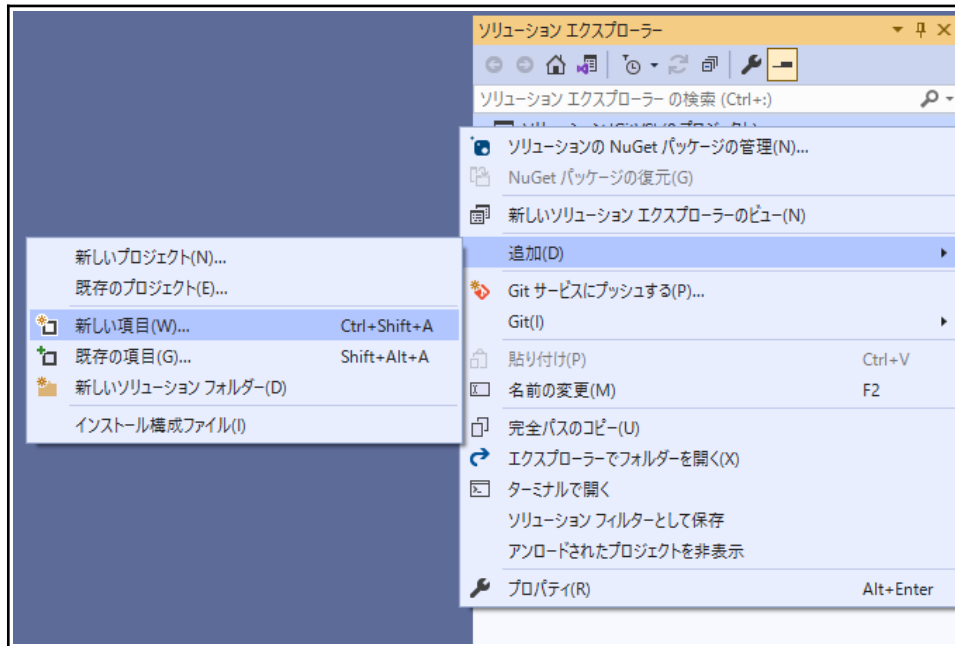


右側の「Git 変更」タブはこのようになります。main ブランチになっていることを確認して下さい。

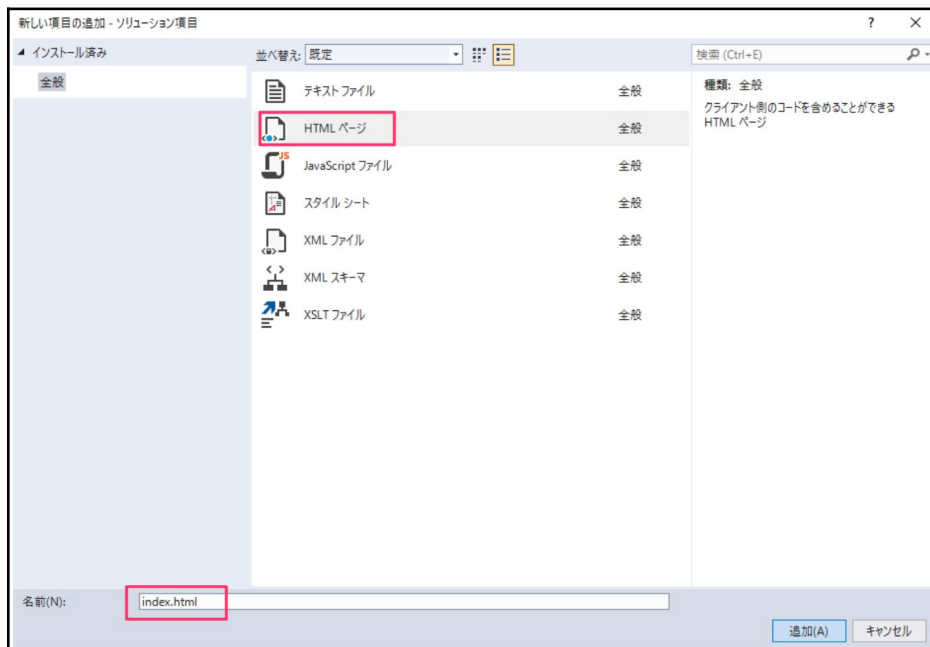


HTML ファイルを作成

ソリューションにファイルを追加します。「ソリューションエクスプローラ」タブに切り替えてソリューションを右クリックし、「追加」→「新しい項目」をクリックします。



「新しい項目の追加」ウィンドウで「HTML ページ」を選び、ファイル名は「index.html」とします。最後に「追加」ボタンをクリックします。



作成されたファイルを次のように編集し、保存して下さい。

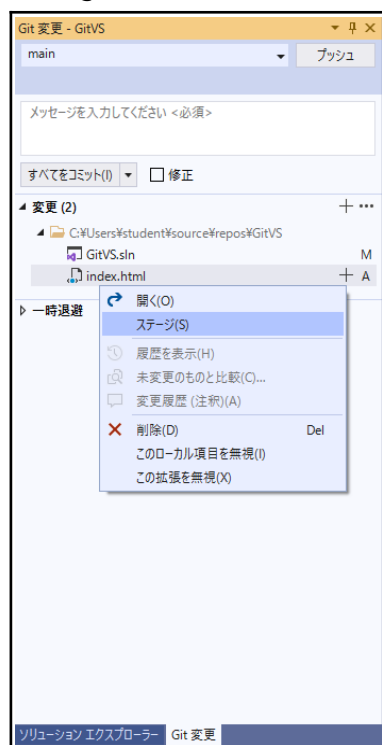
index.html

```
<!DOCTYPE html>

<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta charset="utf-8" />
    <title>Git の練習</title>
</head>
<body>
    <h1>Git の練習</h1>
    <p>main ブランチで段落を作成</p>
</body>
</html>
```

ステージングとコミット

「Git 変更」タブで index.html ファイルの名前を右クリックし、「ステージ」を選びます。これは「git add ファイル名」に該当する操作です。

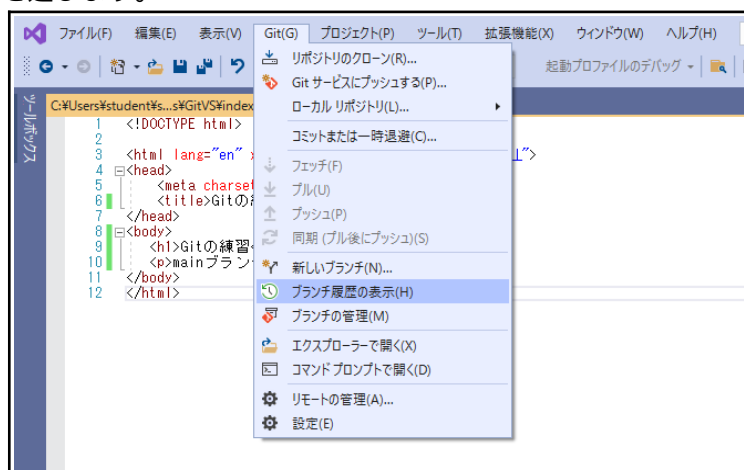


GitVS.slin ファイルも同様に右クリックからステージに追加して下さい。尚その際に保存を求められたら「上書き保存」を選んで下さい。今の操作によって「変更」欄にあった内容が「ステージ」されている「変更」欄へ移動します。

この状態でコミットをします。上部のテキスト入力エリアにコミットメッセージを「段落を作成」と入力し、「ステージング済みをコミット」ボタンをクリックするとコミットが完了します。



コミット履歴を確認してみましょう。画面上部の「Git」メニューから「ブランチ履歴の表示」を選びます。

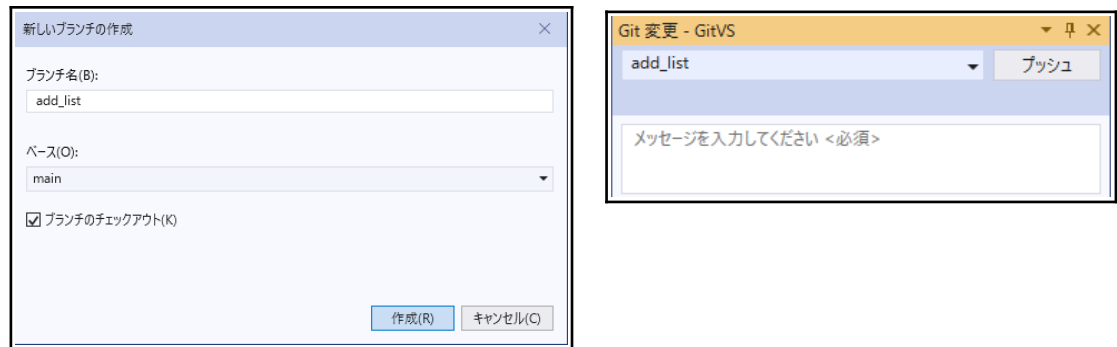


「段落を作成」のログが表示されれば成功です。尚、その下の2つのログは Visual Studio で最初にリポジトリを作成した際に自動的に出来たものです。

ブランチの作成とチェックアウト

「Git」メニューの「新しいブランチ」を選択し、ブランチ名を入力します。ここでは「add_list」という名前にします。

「ブランチのチェックアウト」はチェックしたままで「作成」をクリックします。



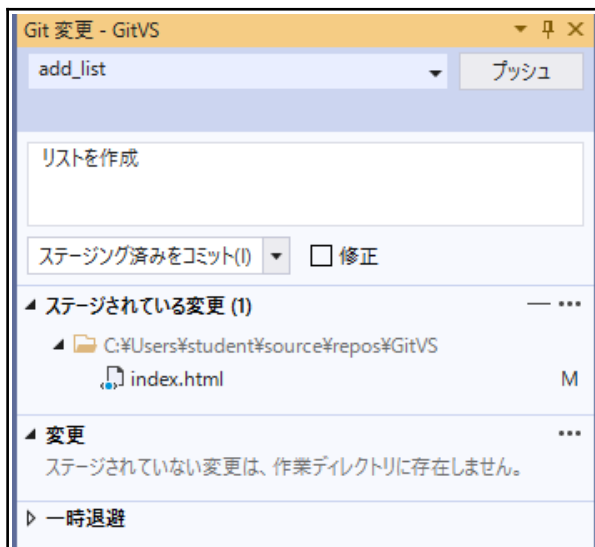
add_list ブランチに切り替わったことを確認してから、index.html を次のように編集して保存して下さい。

index.html

```
<!DOCTYPE html>

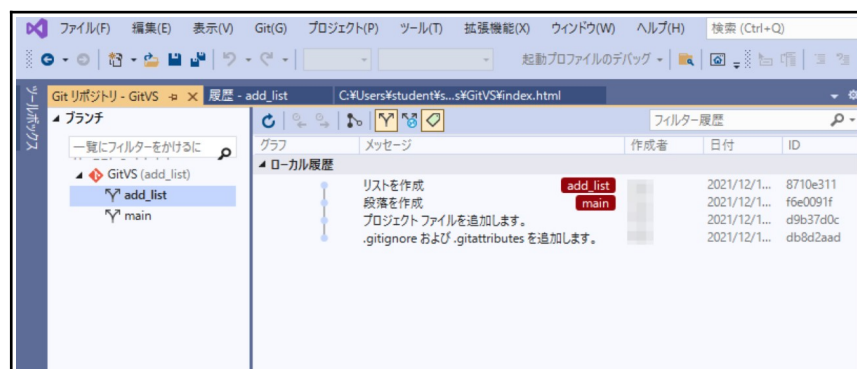
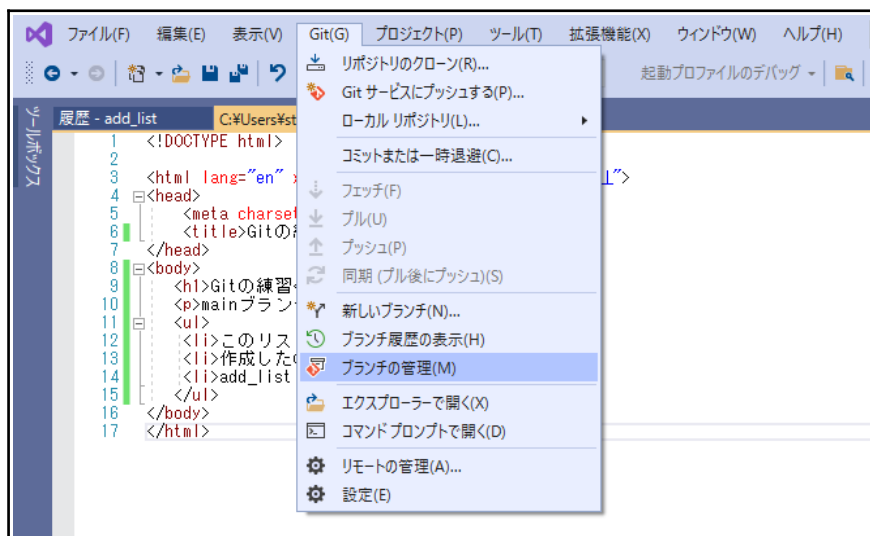
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta charset="utf-8" />
  <title>Git の練習</title>
</head>
<body>
  <h1>Git の練習</h1>
  <p>main ブランチで段落を作成</p>
  <ul>
    <li>このリストを</li>
    <li>作成したのは</li>
    <li>add_list ブランチです</li>
  </ul>
</body>
</html>
```

変更内容を Git に反映させるため、ステージングとコミットを行います。コミットメッセージは「リストの作成」として下さい。



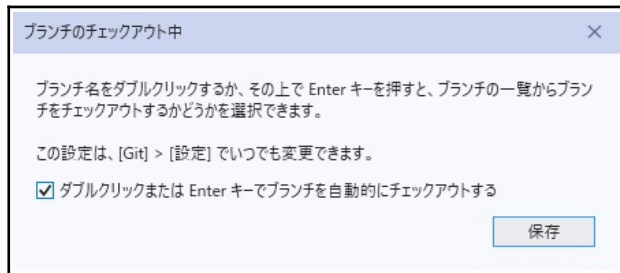
これで main ブランチと add_list ブランチの 2 つのコミット履歴ができました。内容を確認してみましょう。

「Git」メニューから「ブランチの管理」を選択します。

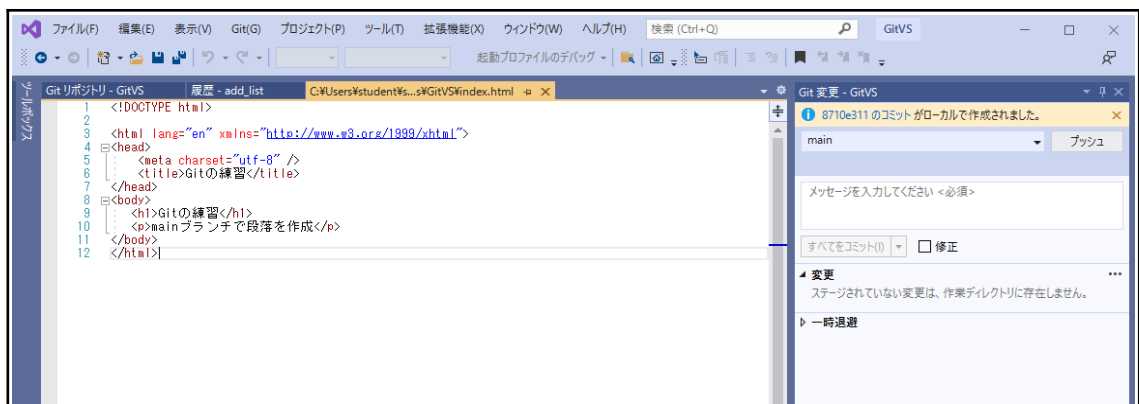


ここから main ブランチにチェックアウトします。

左側のブランチツリーにある main ブランチをダブルクリックすると、次のようなウィンドウが表示されます。チェックボックスにチェックをいれて「保存」をしておけば、次回以降はこのウィンドウが表示されることなく、ダブルクリックするたびにブランチを切り替えられるようになります。



main ブランチでは、先ほどの add_list ブランチで行われた変更は反映されていないことを確認して下さい。index.html にタブを切り替えると下記のように元の内容に戻っています。



今度は新しく「add_definition」ブランチを作成します。

「Git」メニューから「新しいブランチ」を選び、先ほどと同じようにブランチを作成し、チェックアウトまで行います。ベースとなるブランチは「main」のままにして下さい。

add_definition ブランチ上で、index.html ファイルを次のように編集します。

index.html

```
<!DOCTYPE html>

<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta charset="utf-8" />
  <title>Git の練習</title>
</head>
<body>
  <h1>Git の練習</h1>
  <p>main ブランチで段落を作成</p>
  <dl>
    <dt>作成者</dt>
    <dd>add_definition ブランチ</dd>
    <dt>作成内容</dt>
```

```
<dd>定義リスト</dd>
</dl>
</body>
</html>
```

編集し終わったら保存を忘れずに行い、ステージング、コミットまで手順を進めます。コミットメッセージは「定義リストの作成」として下さい。

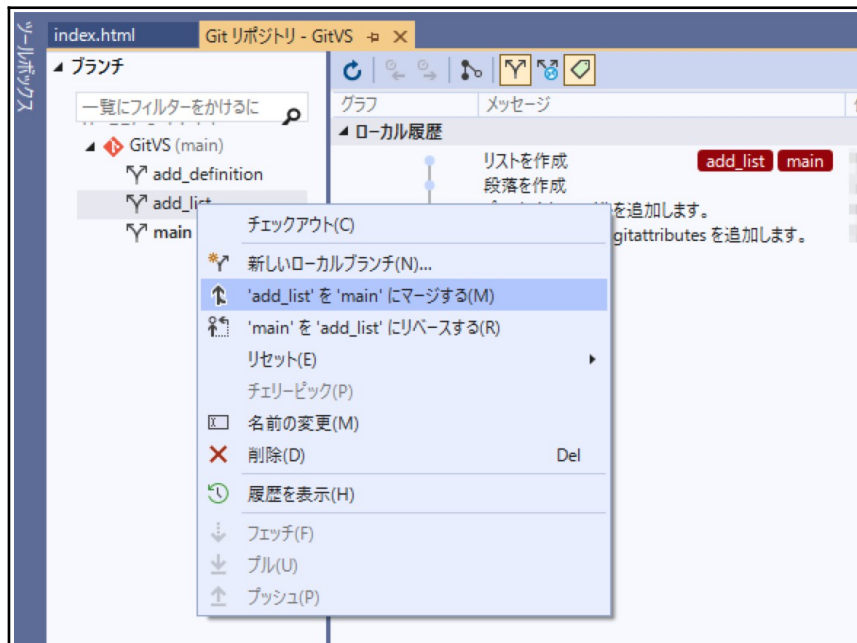
ここまでで、各ブランチのコミットログは次のような状態になっています。

- main ブランチ・・・段落を作成
- add_list ブランチ・・・リストを作成
- add_definition・・・定義リストを作成

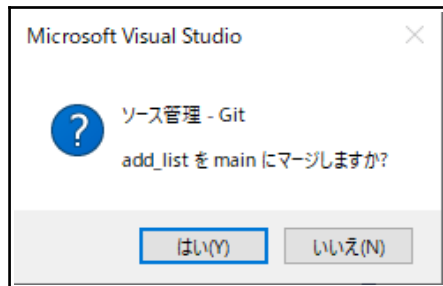
マージ

add_list と add_definition のブランチを main にマージします。まずは add_list からです。「ブランチの管理」を開いて main ブランチに切り替えて下さい。

次に add_list ブランチを右クリックし、「'add_list'を'main'にマージする」を選びます。



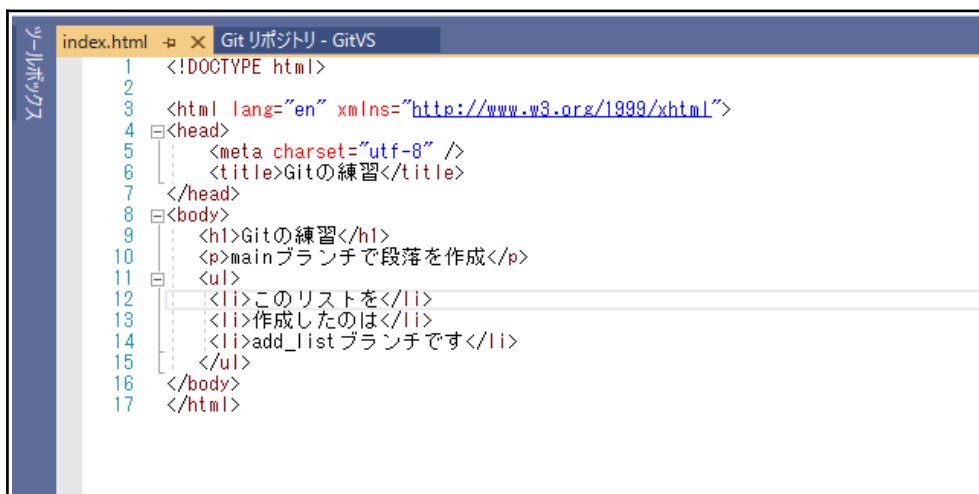
ダイアログでは「はい」をクリックします。



マージが完了しました。「ブランチの管理」を開いて main ブランチのコミットログを確認します。



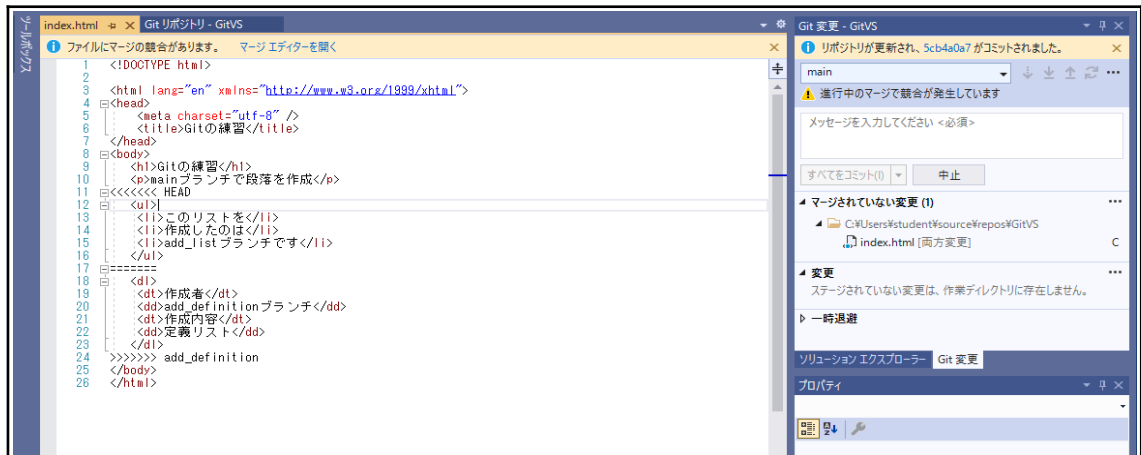
main ブランチの html ファイル上でも内容が反映されていることを確認します。



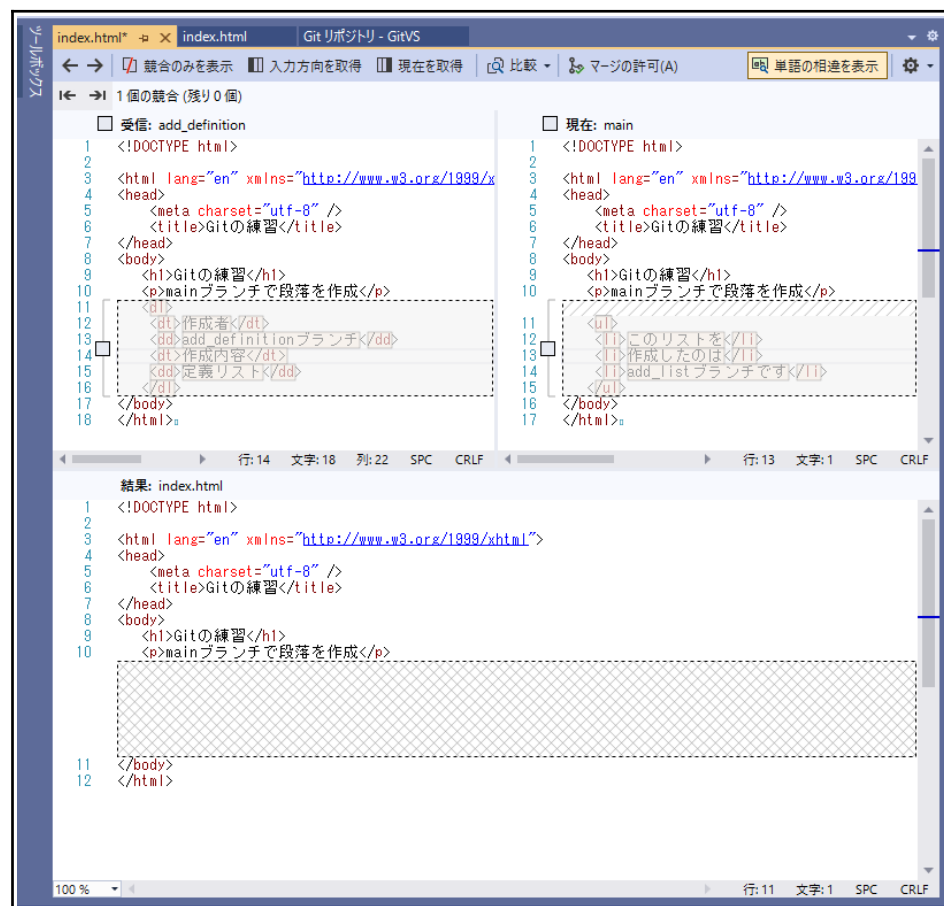
コンフリクト（競合）の解消

今度はadd_definition ブランチをマージします。「ブランチの管理」でadd_definition を右クリックし「'add_definition'を'main'にマージする」を選んで下さい。

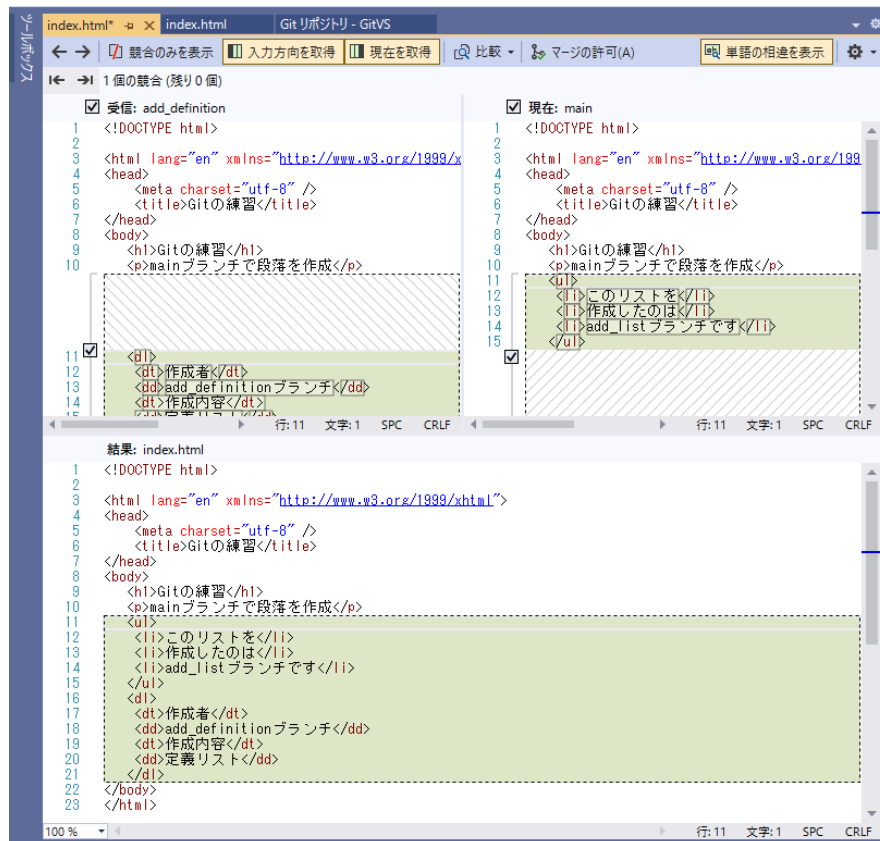
マージを実行すると、先ほどと違って今回はコンフリクトが発生します。



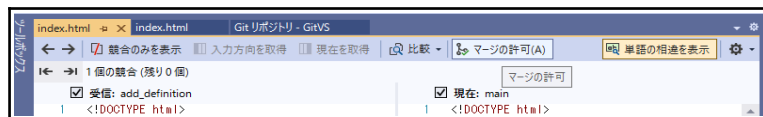
Visual Studio にはコンフリクトの解消をサポートしてくれる「マージエディター」が搭載されています。エディタ上部にある「マージエディターを開く」をクリックして下さい。



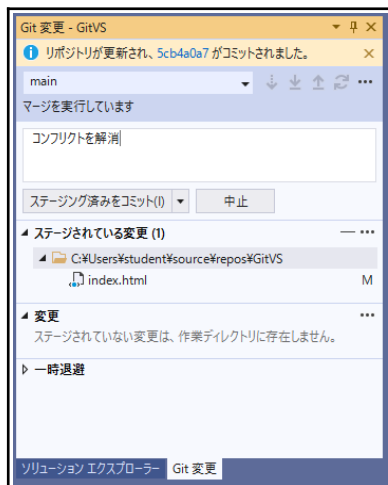
今回は「受信側：add_definition」と「現在：main」の両方の内容を使用したいので、双方のチェックボックスにチェックを入れます。



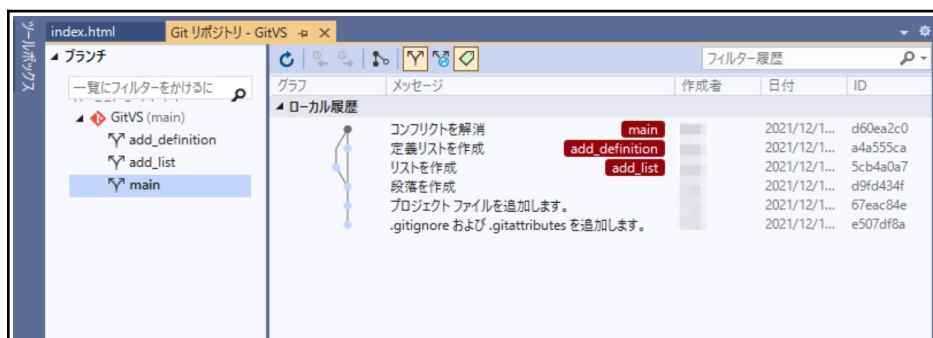
この状態でファイルを保存し、エディタ上部にある「マージの許可」をクリックします。



続いて、マージした内容をコミットします。コミットメッセージは「コンフリクトを解消」として下さい。



「ブランチの管理」で main ブランチのコミット履歴を確認します。



このように一旦枝分かれしたコミットが合流している様子がわかります。

リモートにプッシュ

リモートリポジトリを登録して、そこへプッシュします。本書では GitLab を使用して解説を進めていきます。

GitLab にリモートリポジトリを作成

GitLab にログインし、「新規プロジェクト」→「Create blank project」をクリックします。新規プロジェクト作成画面では、以下の内容でプロジェクトを作成して下さい。

- ◆ プロジェクト名 : gitvs
- ◆ 可視レベル : プライベート
- ◆ リポジトリを初期化した README ファイルを生成する : チェックを外す

作成するとプロジェクトページが表示されます。「クローン」ボタンをクリックすると、リモートリポジトリの URL が表示されるので、コピーしておいて下さい。これを Visual Studio の設定画面で使します。

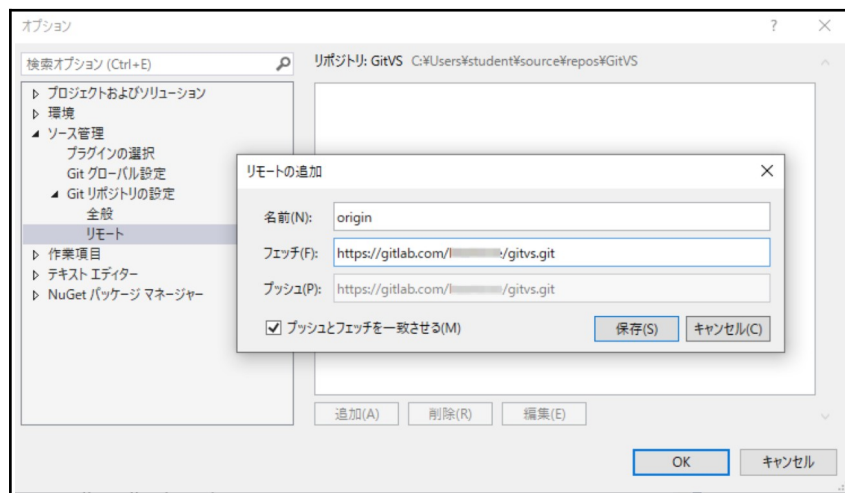


リモートの登録

Visual Studioに戻り、「Git」メニューから「リモートの管理」を選びます。

「オプション」ダイアログで「ソース管理」→「Git リポジトリの設定」→「リモート」が選択された状態で、「追加」ボタンを押します。

「リモートの追加」ダイアログが表示されたら、以下の内容で入力して「保存」をクリックします。

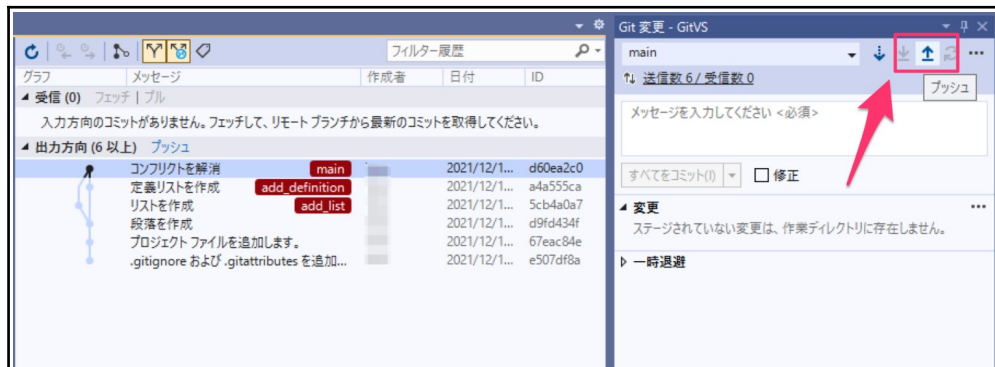


- ◆ 名前・・・origin
- ◆ フェッチ・・・GitLab でコピーしておいたリモートの URL
- ◆ プッシュ・・・「プッシュとフェッチを一致させる」にチェック

保存したら最後に「OK」を押して登録を終了します。

リモートにプッシュ

リモートの登録ができると、「Git 変更」タブの上部が変化します。上矢印のボタンからリモートにプッシュできるようになっています。



GitLab の方にもこのように反映されていることを確認してください。



クローン

今度はリモートリポジトリをクローンして、Visual Studio 上でソースを扱えるようにします。

Visual Studio を再起動するか、または「ファイルメニュー」から「ソリューションを閉じる」を選択し、スタートウィンドウで「リポジトリのクローン」を選びます。



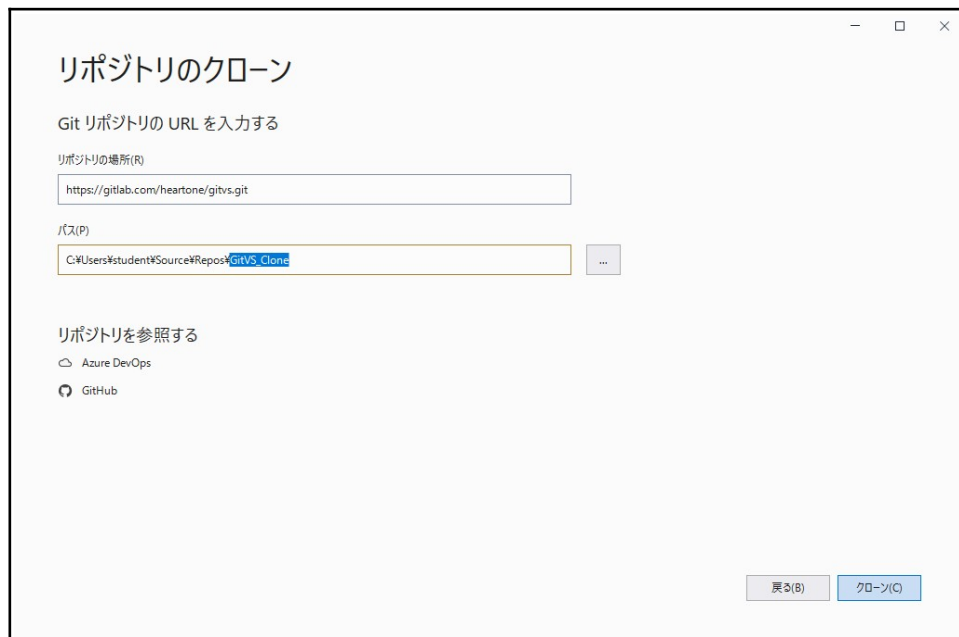
「リポジトリのクローン」ウィンドウでは、「リポジトリの場所」の欄に先ほどコピーした GitLab 上のリポジトリの URL を貼り付けます。

`https://gitlab.com/[GitLab ユーザー名]/gitvs.git`

「パス」の欄には、参照ボタンから新しいフォルダ「GitVS_Clone」を作成し、そこをターゲットとします。または入力欄に直接記入しても構いません。

例

`C:\Users¥[Windows ユーザー名]¥Source¥Repos¥GitVS_Clone`



「クローン」をクリックすればクローンが開始されます。

