

Multivariable Calculus Self-Learning Module

User Manual

Contents

1	Introduction	2
2	Installation / setup	2
2.1	Git	2
2.2	GitHub	2
3	Usage	3
3.1	Starting a new Git repository	3

1 Introduction

The website is currently hosted for free by *GitHub Pages*. In order to deploy a website using GitHub Pages you need a *GitHub* account.

What is GitHub? GitHub is a very popular, free, cloud-based platform that allows users to create, store and share code. Think of it as a storage space just like Google Drive for anything you might want to share with others. Each project on GitHub is stored in its own directory, which is called a *repository*. GitHub uses *Git* to keep track of any changes that happen in a repository.

What is Git? Git is an open-source version control software, which is widely used to keep track of any changes that happen in a file. Think of it as an alternative to making multiple files such as: “exam-v1”, “exam-v2”, “exam-v1-fixed”, ..., “exam-final”. Git keeps track of all the changes made, and you can revert back to any previous version of your file if you mess up, or you can test new stuff by branching out to a new path without ever losing your main path. Git will store the entire version tree and will let you jump on any branch at any moment. GitHub uses Git, but Git is independent of GitHub and works with its own local repository.

How does GitHub Pages work? You simply deploy your website directly from your GitHub repository on GitHub pages. After that point, your website is always running, and whenever you make a change to the repository, it will almost immediately be shown on the website.

What is the point of all this? The idea is the following:

1. You keep your own version of the website on your computer, along with a local installation of Git.
2. Say you make some changes that you are happy with, e.g. you add an exercise or fix a mistake. You *commit* the changes to your local Git repository.
3. You then *push* the local Git repository to the online GitHub repository.
4. GitHub pages will automatically detect the new changes from your GitHub repository and deploy a new version of your website.

The good part is that once setup this process is totally free, standardised, and should be familiar to anyone who has ever collaborated on a coding project.

2 Installation / setup

2.1 Git

The latest version of Git for Windows can be downloaded from:

<https://git-scm.com/download/win>

In the vast majority of cases the 64-bit version (first link) is required. The default settings are fine. After installation is done, check if everything is alright by opening a terminal (can be done by Windows key + X > Terminal (Admin)) and typing the command:

```
git -v
```

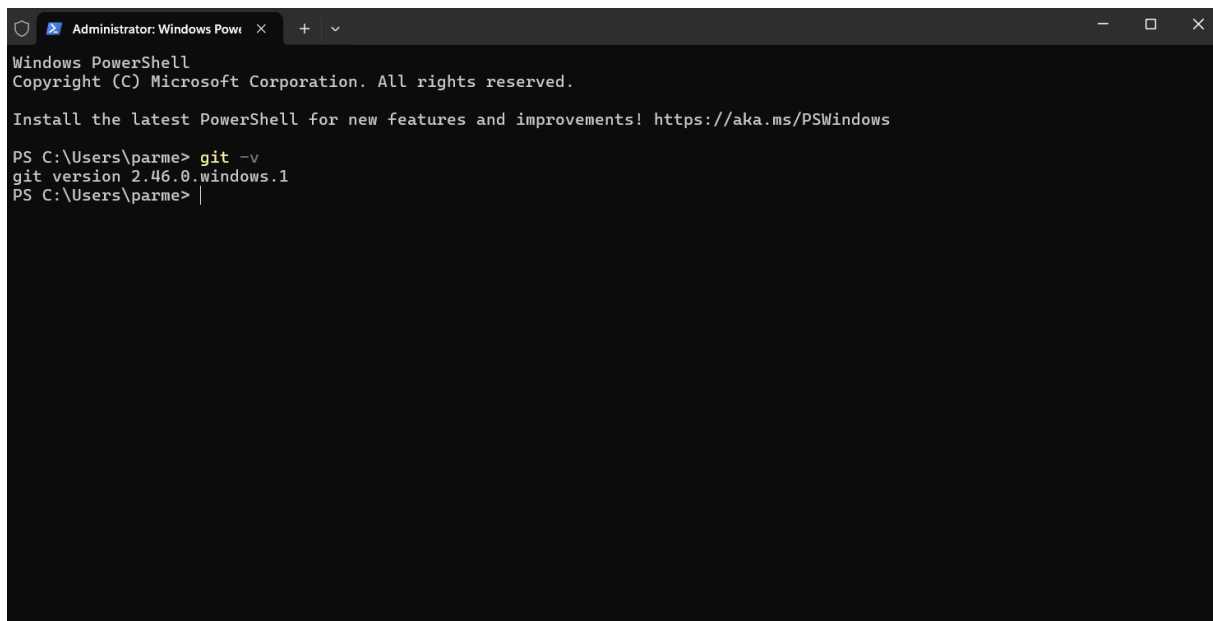
The response should be the latest version of git (Figure 1).

If not, try restarting the computer so that Git is added to the Windows PATH variable, which lets Windows know what the *git* command means. If this also doesn't work, Git must be added to the PATH manually. This is annoying and should not really happen but let me know if it does so that I can add instructions.

2.2 GitHub

Opening a GitHub account should be as straightforward as opening any account in any other service if you select “Sign Up” from

<https://github.com/>

A screenshot of a Windows PowerShell terminal window. The title bar reads 'Administrator: Windows PowerShell'. The window content shows the following text: 'Windows PowerShell', 'Copyright (C) Microsoft Corporation. All rights reserved.', 'Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows', 'PS C:\Users\parme> git -v', 'git version 2.46.0.windows.1', and 'PS C:\Users\parme> |'.

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\parme> git -v
git version 2.46.0.windows.1
PS C:\Users\parme> |
```

Figure 1: git -v response example.

3 Usage

3.1 Starting a new Git repository

There are two ways to initiate a new Git repository.

1. **Initiate a new Git repository for an existing local folder:** Suppose you have the folder *mvc-self-learning-module* somewhere on your computer and it is not already associated with a git repository. In order to initiate one, you can open the folder in a terminal (can be done by navigating within the folder, and then Right Click > Open in Terminal) and typing the command:

```
git init
```

You can visually inspect the Git repository if you have the option of seeing the hidden files on your computer (can be done within the File Explorer itself from View > Show > Hidden Items (Figure 2)). In that case, a folder named *.git* should appear in the current directory, which shows that the directory is now a git repository (first folder in Figure 2).

2. **Clone an already existing Git repository from GitHub:** This can be done as follows:
 - (a) Find the GitHub repository using your browser.
 - (b) Copy the link to the repository, e.g. `https://github.com/StefanMaubach/mvc-self-learning-module`.
 - (c) Navigate to the location on your computer where you would like to store the repository, and type the command:

```
git clone https://github.com/StefanMaubach/mvc-self-learning-module
```

The repository should then appear in a new folder named *mvc-self-learning-module* in the desired location.

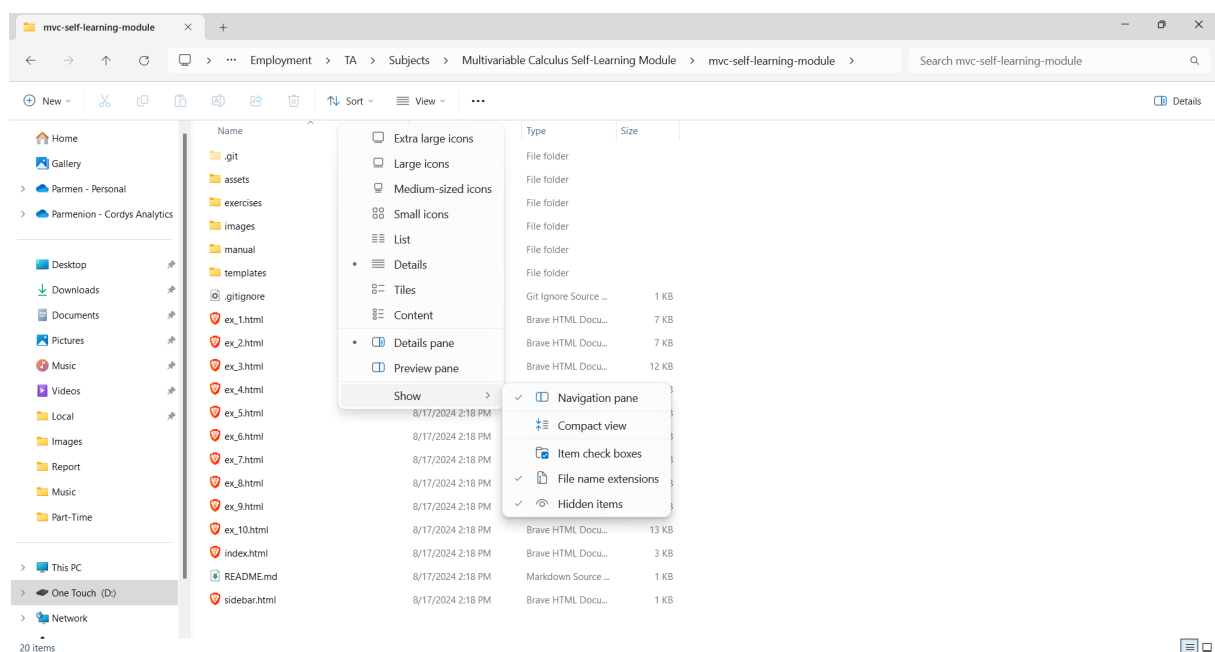


Figure 2: Enable hidden items.