



Getting started with Git and GitHub for the reluctant IT Pro

Thomas Vochten

Agenda

1. Introduction to git
2. The tools you need to get started
3. Getting basic stuff done with git
4. Where to go next?



Thomas Vochten

@thomasvochten



<https://thomasvochten.com>



mail@thomasvochten.com



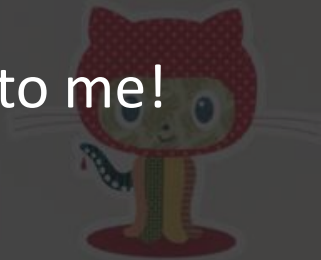
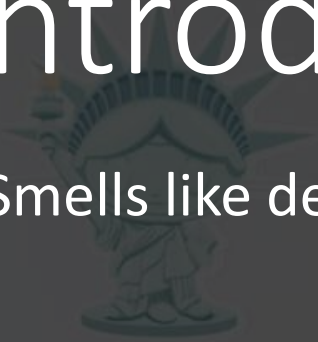
BIVUG



☁ Technology Evangelist ☕ 💻
#Microsoft365 #Azure #CommunityRocks

1 | Introduction to git









Smells like developer stuff to me!



A photograph of two puppets designed to look like Albert Einstein and an older man, sitting at a table. The Einstein puppet is on the left, with his characteristic wild white hair and a small white mustache. He has a weary or skeptical expression, with his eyes partially closed. The second puppet, on the right, is an older man with a large, bulbous nose, thick eyebrows, and a wide, somewhat mischievous or questioning smile. Both puppets are wearing dark, textured clothing. They are seated in ornate chairs with red upholstered seats and backs. The background is dark and out of focus, suggesting an indoor setting. The overall lighting is dim, creating a moody atmosphere.

Why exactly would I care?

Rings a bell?

-  Script_todo_Stuff.ps1
-  Script_todo_Stuff_v2.ps1
-  Script_todo_Stuff_v2-CustomerA.ps1
-  Script_todo_Stuff_v3.ps1
-  Script_todo_Stuff_v4.ps1
-  Script_todo_Stuff_v4-CustomerB.ps1
-  Script_todo_Stuff_v5.psm1
-  Script_todo_Stuff_v5_prod.psm1



Source control to the rescue

- Version control for code or scripts
- Keep different versions of your files
- Ability to go back if you messed up
- Work together with other people
- Work on new stuff while keeping the existing files intact
- Make sure your new stuff stays up to date when you change the existing files
- ...



Git is a free and open-source
distributed version control system

<https://thvo.me/git>

Key concepts

- Branching & merging
- Distributed
- Staging area
- Small & fast
- Robustness





What about GitHub then?

- Hosted git
- Acquired by Microsoft in 2018
- The de facto standard

Some alternatives



Bitbucket



Gitlab



Azure Repos

THIS IS GIT. IT TRACKS COLLABORATIVE WORK
ON PROJECTS THROUGH A BEAUTIFUL
DISTRIBUTED GRAPH THEORY TREE MODEL.

COOL. HOW DO WE USE IT?

NO IDEA. JUST MEMORIZE THESE SHELL
COMMANDS AND TYPE THEM TO SYNC UP.
IF YOU GET ERRORS, SAVE YOUR WORK
ELSEWHERE, DELETE THE PROJECT,
AND DOWNLOAD A FRESH COPY.



<https://ohshitgit.com/>

Oh shit, git!

Git is hard: screwing up is easy, and figuring out how to fix your mistakes is fucking impossible. Git documentation has this chicken and egg problem where you can't search for how to get yourself out of a mess, unless you *already know the name of the thing you need to know about* in order to fix your problem.

So here are some bad situations I've gotten myself into, and how I eventually got myself out of them *in plain english*.*

Oh shit, I did something terribly wrong, please tell me git has a magic time machine!?!

```
git reflog
# you will see a list of every thing you've done in git, across all br
# each one has an index HEAD@{index}
# find the one before you broke everything
git reset HEAD@{index}
# magic time machine
```

You can use this to get back stuff you accidentally deleted, or just to remove some stuff you tried that broke the repo, or to recover after a bad merge, or just to go back to a time when things actually worked. I use `reflog` A LOT. Mega hat tip to the many many many many many people who suggested adding it!

Oh shit, I committed and immediately realized I need to make one small change!

Dangit, git!

Git is hard: screwing up is easy, and figuring out how to fix your mistakes is nigh on impossible. Git documentation has this chicken and egg problem where you can't search for how to get yourself out of a mess, unless you *already know the name of the thing you need to know about* in order to fix your problem.

So here are some bad situations I've gotten myself into, and how I eventually got myself out of them *in plain english*.*

Dangit, I did something terribly wrong, please tell me git has a magic time machine!?!

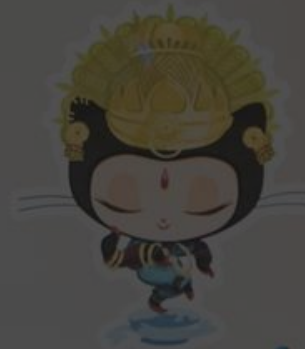
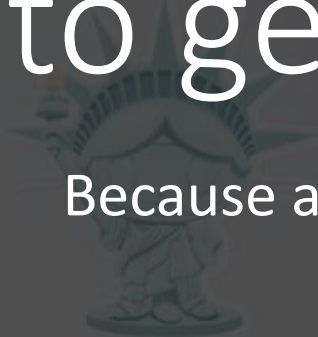
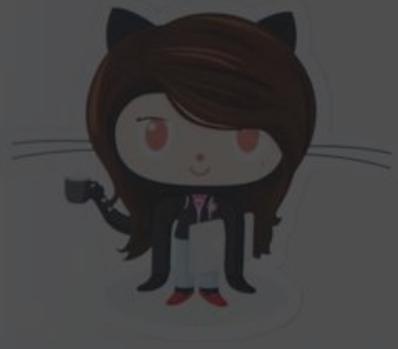
```
git reflog
# you will see a list of every thing you've done in git, across all br
# each one has an index HEAD@{index}
# find the one before you broke everything
git reset HEAD@{index}
# magic time machine
```

You can use this to get back stuff you accidentally deleted, or just to remove some stuff you tried that broke the repo, or to recover after a bad merge, or just to go back to a time when things actually worked. I use `reflog` A LOT. Mega hat tip to the many many many many many people who suggested adding it!

Dangit, I committed and immediately realized I need to make one small change!

2 | The setup you need to get started

Because all geeks love tools

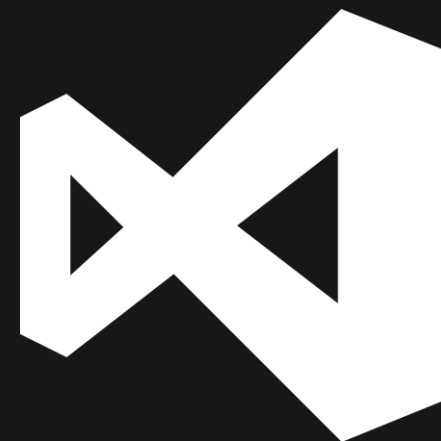




git



Windows Terminal



Visual Studio Code

GitHub Desktop

<https://thvo.me/ghdesktop>



GitHub Desktop interface showing the Logitech Camera Settings 2.12.8.0 commit (#4700) by Thomas Vochten.

Current repository: winget-pkgs
Current branch: master
Fetch origin: Last fetched just now

Changes: Select branch to compare...

History: Added Jitsi Meet version 2.4.2. (#4672) by Easton Pillay • 2d, Submitting HttpMaster Professional 4... by joxi, HttpMaster • 2d, VLC media player nightly version 4.0.0-... by ItzLevvie • 2d, Add EduMIPS64 v1.2.9 (#4695) by Andrea Spadaccini • 2d, add Stretchly 1.3.0 (#4697) by Asurada • 2d, Update conan to 1.30.2 (#4473) by Edgar • 2d, Update SoundSwitch to v5.5.4 (#4683) by Antoine Affalo • 2d, Fiddler Everywhere Insiders version 1.2.... by ItzLevvie • 2d, Fiddler Everywhere Internal version 1.2.... by ItzLevvie • 2d, Fiddler Everywhere version 1.2.1 (#4708) by ItzLevvie • 2d, FileZilla Client Nightly version 3.51.0-2... by ItzLevvie • 2d, Node.js Nightly version 16.0.0-2020110... by ItzLevvie • 2d, Zoom version 5.4.2 (#4703) by ItzLevvie • 2d, Create 3.7.9.yaml for Python (#4710) by ItzLevvie • 2d.

Logitech Camera Settings 2.12.8.0 (#4700)
Thomas Vochten • 22486255 • 1 changed file • Hide Whitespace

manifests\Logitech\Ca...2.12.8.0.yaml

```
@@ -0,0 +1,16 @@
1 +Id: Logitech.CameraSettings
2 +Publisher: Logitech
3 +Name: Logitech Camera Settings
4 +Version: 2.12.8.0
5 +AppMoniker: camerasettings
6 +Description: Provides easy access to camera controls
7 +Homepage: https://support.logi.com/hc/en-us/articles/360049055854
8 +License: (c) 2019 Logitech
9 +InstallerType: nullsoft
10 +Switches:
11 +   Silent: /S
12 +Installers:
13 +   - Arch: Neutral
14 +     Url: https://download01.logi.com/web/ftp/pub/techsupport/cameras/webcams/LogiCameraSettings_2.12.8.exe
15 +     Sha256: 4A18682E139A48F665F78F348C2887E52C656C1C483797817A28496688088B97
16 +# ManifestVersion: 0.1.0
```

Upgrade your command prompt

```
C:\Dev\MyRepository > ps main &
```

```
C:\Dev\MyRepository > ps main & +0 ~1 -0 !
```

Upgrade your command prompt

- Install Windows Terminal
- Install a Powerline-capable font, e.g. from NerdFonts:

<https://thvo.me/nerdfonts>

- Setup git & Powerline integration in PowerShell

```
Install-module posh-git  
Install-Module oh-my-posh  
Set-Theme Paradox
```

- Modify your Windows Terminal profile settings to use the font

"fontFace": "FiraCode Nerd Font"

Integration with Visual Studio Code



- Supports the most common git commands by default
- Use the command line for more advanced scenarios
- Useful extensions:



GitLens

<https://thvo.me/gitlens>



Git History

<https://thvo.me/githistory>



GitHub Pull Requests & Issues

<https://thvo.me/vscodegithub>

GitHub CLI

```
$ gh issue list
```

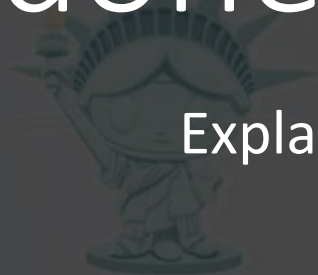
```
Showing 4 of 4 issues in cli/cli
```

```
#16 Improving interactions with protected branches  
#14 PR commands on a detached head  
#13 Support for GitHub Enterprise (enhancement)  
#8 Add an easier upgrade command (bug)
```

<http://thvo.me/githubcli>

3 | Getting basic stuff done with git

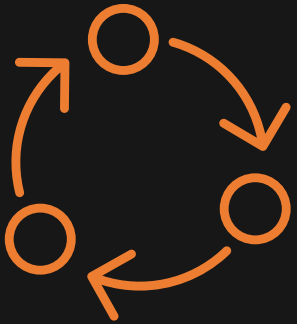
Explain it like I'm 5



Basic git workflow

You clone or create a repo

You work in your local copy of the repo



- You commit changes to your local repo
- You push to the remote repo (e.g. GitHub) when you're ready
- You pull from the remote repo regularly to stay up to date

Essential Lingo

- Downloading an existing repo to your pc (**cloning**)
- Checking in your changes (**committing**)
- Move your changes to a remote repository (**pushing**)
- Getting changes from the remote repository (**pulling**)
- Integrating changes (**merging**)
- Creating a separate area to work on stuff (**branching**)
- Creating your own version of existing code (**forking**)
- Asking someone else to integrate your work (**pull request**)

① Many practices are just conventions

Demo - Common Operations

git clone, git status, git commit, git add, git push, git pull

	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

Initializing a (remote) repository

Quick setup — if you've done this kind of thing before



Set up in Desktop

or

HTTPS

SSH

<https://github.com/thomasvochten/TechTrain.git>



Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# TechTrain" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/thomasvochten/TechTrain.git
git push -u origin main
```



...or push an existing repository from the command line

```
git remote add origin https://github.com/thomasvochten/TechTrain.git
git branch -M main
git push -u origin main
```



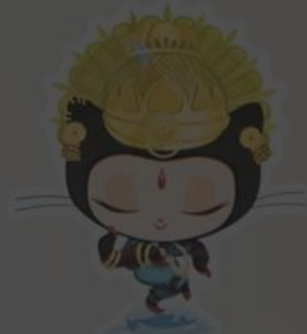
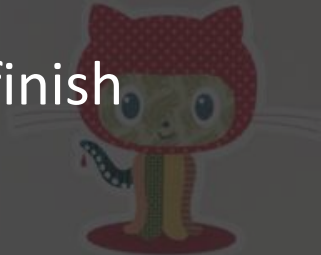
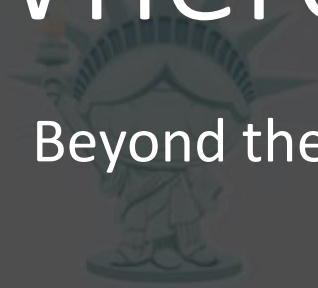
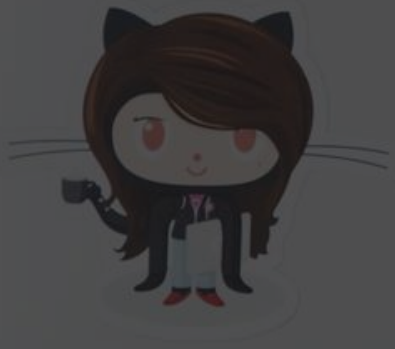
...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code

4 | Where to go next?

Beyond the next - next - finish



What to learn next

- Get friends with clone, commit, push & pull
- Learn how to work with (remote) repositories
- Create your first branch and merge it back in
- Fork and existing repository
- Issue your first pull request

In case of fire



1. git commit

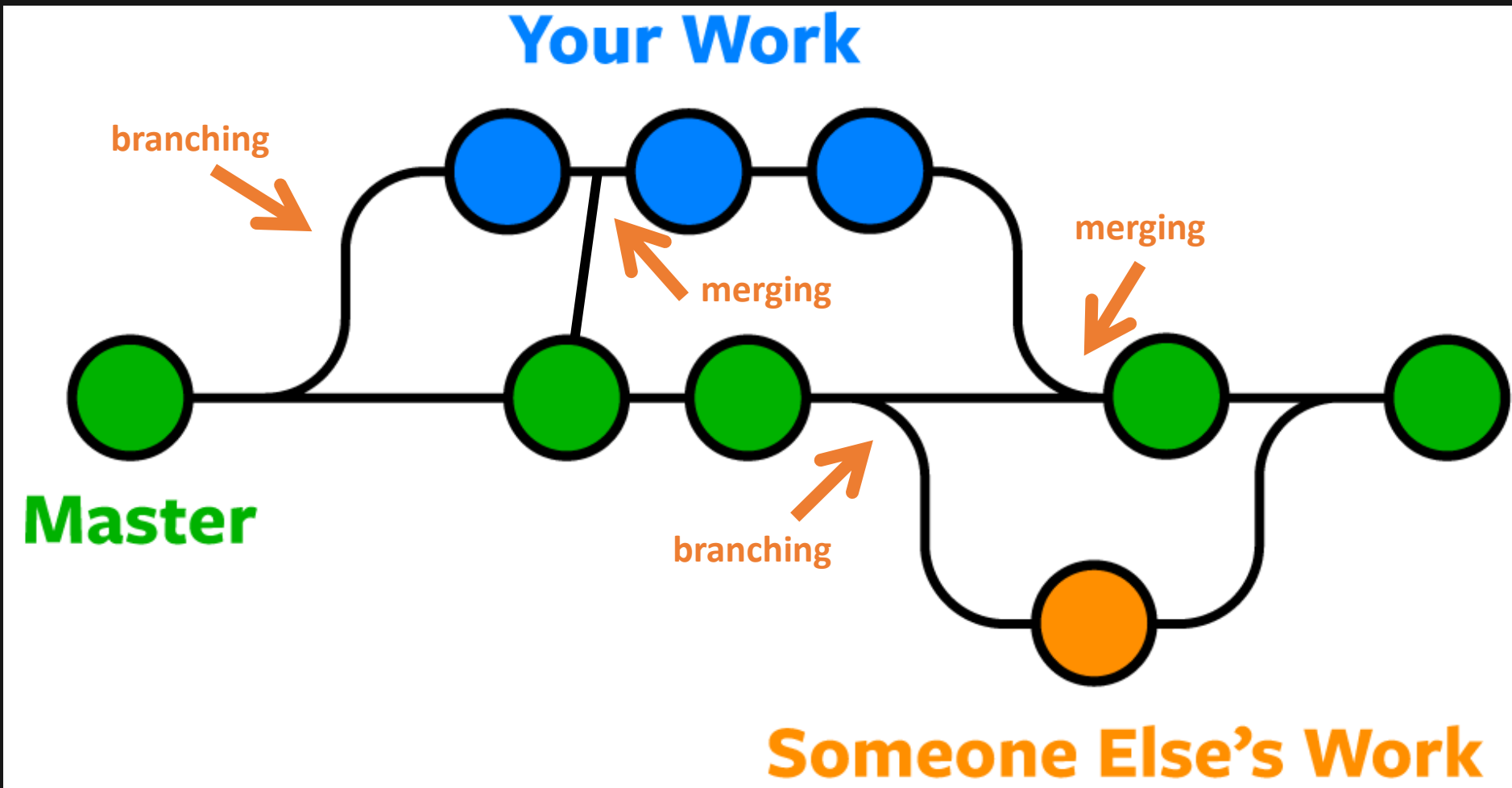


2. git push



3. leave building

Branching & Merging



Sharing is caring!

Everyone can contribute.

Share the things you're proud of:


- PowerShell scripts & modules
- Samples and tutorials
- Documentation
- ...

Contributing to docs.microsoft.com

Set up development environment

Bookmark Feedback **Edit** Share

Set up your SharePoint Framework development environment

07/31/2020 • 5 minutes to read •  +19

You can use Visual Studio or your own custom development environment to build SharePoint Framework solutions. You can use a Mac, PC, or Linux environment as well.


Note

Before following the steps in this article, be sure to [Set up your Microsoft 365 tenant](#).

You can also follow these steps by watching this video on the SharePoint PnP YouTube Channel:



Is this page helpful?

 Yes  No

In this article

[Install Node.js](#)

[Install a code editor](#)

[Install development toolchain prerequisites](#)

[Install a modern web browser](#)

[Trusting the self-signed developer certificate](#)


[Optional tools](#)

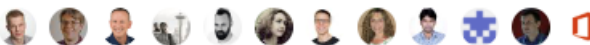
[SPFx & SharePoint Server 2016](#)

[Next steps](#)

[Troubleshooting](#)




master [sp-dev-docs / docs / spfx / set-up-your-development-environment.md](#) [Go to file](#) [...](#)

 heythisispaul Merge branch 'master' into fix/trusting-self-signed ✓ Latest commit bb13e2d 12 days ago [History](#)

[25 contributors](#)  +13

149 lines (94 sloc) | 8.91 KB

[Edit the file in your fork of this project](#)


[Raw](#) [Blame](#)   


	title	description	ms.date	ms.prod	localization_priority	ms.custom
--	-------	-------------	---------	---------	-----------------------	-----------

You're making changes in a project you don't have write access to. Submitting a change will write it to a new branch in your fork thomasvochten/sp-dev-docs, so you can send a pull request.


[sp-dev-docs / docs / spfx /](#) [set-up-your-developmei](#) [Cancel](#)

<> Edit file

 Preview changes

Spaces 


2 

Soft wrap 

```
1 ---
2 title: Set up your SharePoint Framework development environment
3 description: Use Visual Studio or your own custom development environment to build SharePoint Framework solutions. You can use a Mac, PC, or Linux.
```


Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).

 base repository: SharePoint/sp-dev-docs ▾

base: master ▾

←

head repository: thomasvochten/sp-dev-docs ▾

compare: patch-2 ▾

✓ **Able to merge.** These branches can be automatically merged.

Discuss and review the changes in this comparison with others. [Learn about pull requests](#)

Create pull request

🔗 1 commit

± 1 file changed

💬 0 comments

👤 1 contributor



 Commits on Aug 12, 2020



  Typo correction

Unverified 65ecef1

± Showing 1 changed file with 1 addition and 1 deletion.

Unified Split

▼ 2  docs/spfx/set-up-your-development-environment.md 

  ...

Typical open-source contribution workflow

- Fork the repo you want to contribute to
- Create a branch for your change
- Make & test your changes
- Create a pull request
- Wait for the pull request to be accepted
- Remove your fork

 Learn about pull requests etiquette...

Key takeaways

- Getting started with git is relatively simple
- The learning curve is quite dramatic once you go beyond the basics
- Don't be afraid of the command line, but also:
- Take advantage of tools like VSCode or GitHub Desktop to help you
- Commit, push & pull as often as you can!
- Don't stop here!



Additional resources

Video series by Scott Hanselman

<https://thvo.me/scotthagit>

Git documentation

<https://thvo.me/gitdocs>

Set up Powerline in Windows Terminal

<https://thvo.me/wintermpowerline>

@thomasvochten

THANK
YOU



Download session materials