

# Version control

(using git)

---

**Danny Awty-Carroll**

May 15, 2018

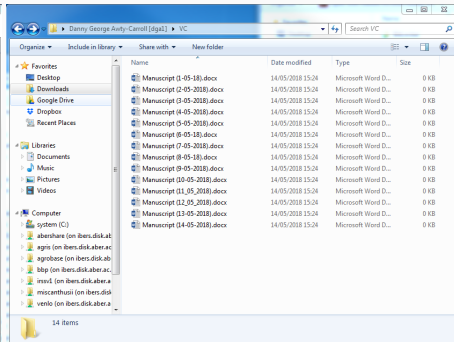
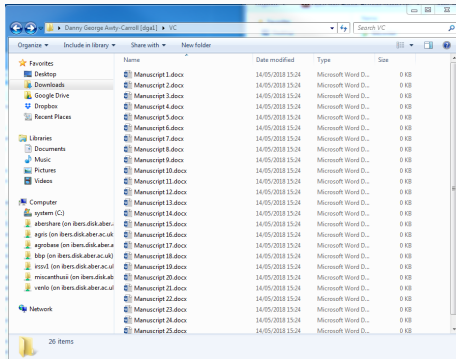
This will focus on using git in windows with a UI

# Why version control?

---

# What is version control

This is just the management of versions of a document.  
One document through time.



All of us use some version control

# Where things get complicated

Numbering or dating documents works OK but can fall down when

- There are multiple documents that need to work together (i.e. a script and data)
- There are multiple people working on the documents (are they on the latest version and merging changes)
- There are updates to the project that may break things
- Line changes need to be reviewed

Some of this is solved in programs like word with track changes (so you see who altered what when)

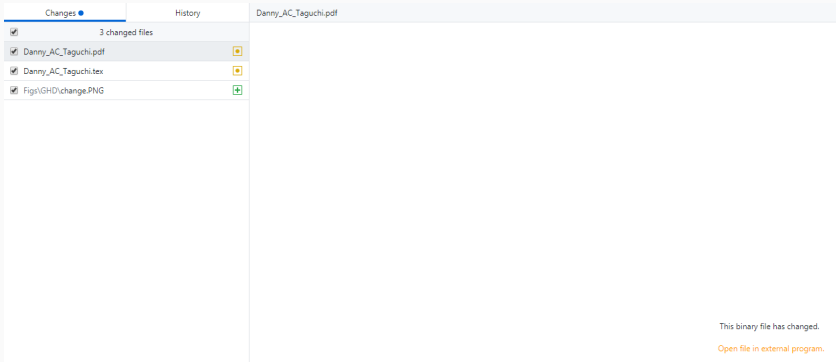
# What version control systems can do

- Give line by line user by user line changes
- Make a stucher to vershons so there can be side branches
- Minimise problems of multiple users working on the same document at the same time

Changes ●	History	Danny_AC_Taguchi.tex	
✓ 2 changed files		72	72
✓ Danny_AC_Taguchi.pdf		73	73
✓ Danny_AC_Taguchi.tex		74	74
		75	- \item There are multiple documents that need to work together (i.e. a script and data)
		76	- \item There are multiple people working on the documents (are they on the latist version and merging changes)
		77	- \item There are updates to the project that may brake things
		78	- \item Line changes need to be reviewed
		75	+ \item Give line by line user by user line changes
		76	+ \item Make a stucher to vershons so there can be side branches
		77	+ \item Minimise problems of multiple users working on the same document at the same time
		79	\end{itemize}
		80	\end{frame}
		81	
		82	-\begin{frame}[fragile]{what version control systems can do}
		81	+
		82	+\begin{frame}[fragile]{what version control systems can do}
		83	\begin{itemize}
		84	\item There are multiple documents that need to work together (i.e. a script and data)
		85	\item There are multiple people working on the documents (are they on the latist version and merging changes)

# What version control systems can't do

- Add much extra control to binary files (not plan text)
- Back up in real time



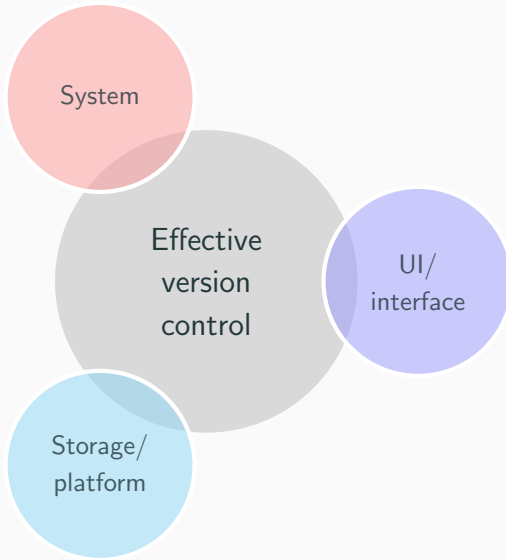
# Version control system

---



# What version control system?

The first thing about version control systems is which system (we will cover git)



# What version control system?

- There are two popular version control systems (git SVN)
- We will cover git as it is the most popular, the one I know and easiest to use
- Git was developed in 2005 by Linus Torvalds

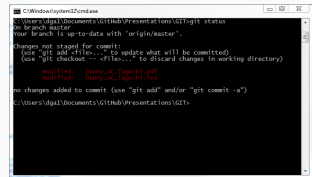


(Principal developer of Linux)

# What is git?



- Git is just system of version control

A screenshot of a Windows Command Prompt window titled "GitWindows\system32\cmd.exe". The window shows the output of the "git status" command. The text is as follows:

```
C:\Users\dgal\Documents\Github\Presentations\GIT>git status
On branch master
Your branch is up-to-date with 'origin/master'.

changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   Dumpy-X-1[epoch].pdf
        modified:   Dumpy-X-1[epoch].tex

no changes added to commit (use "git add" and/or "git commit -a")
C:\Users\dgal\Documents\Github\Presentations\GIT>
```

- By default it is used through a terminal
- Git has projects could reposetrys (or repos)
- It then has a tree struchure to manige the versions

# Tree structure?



[master] 6c6faa5 My first commit - John Doe

[develop] 3e89ec8 Develop a feature - part 1 - John Doe

[develop] e188fa9 Develop a feature - part 2 - John Doe

[master] 665003d Fast bugfix - John Fixer

[myfeature] eaf618c New cool feature - John Feature

[master] 8f1e0e7 Merge branch 'develop' into 'master' - John Doe

[master] 6a3dacc Merge branch 'myfeature' into 'master' - John Doe

0.1

[master] abcdef0 Release of version 0.1 - John Releaser



The master branch is in gray with colour branches coming of and then being merged back

# Key comards?

In the terminal interface there are some useful commards:



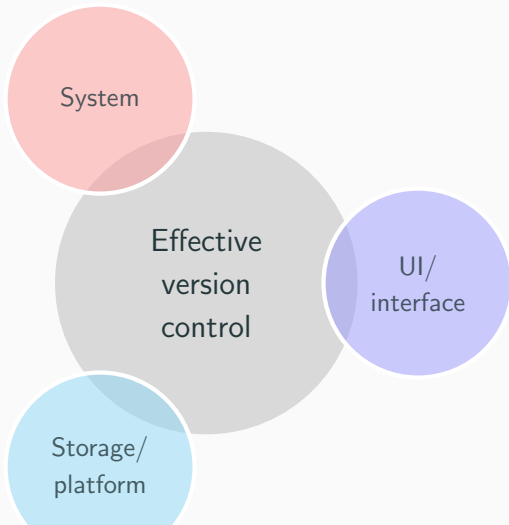
- *init* - inisholies a repo
- *status* - tells you if you have flies out of sync with the current verntion
- *commit* - adds commits to the current verntion - a change with a comment and an ID
- *add* - adds new files and current commit
- *remove* - removes files and current commit
- *push* - pushes comits to an online repo (only if using a git platform)
- *pull* - pulls commits from an online platform (only if using a git platform)

# Platform

---

# Which platform?

You can just use git with the terminal on your PC or you can store work on an online repository managing site, the most popular of these are GitHub and Bitbucket



# Why not just store the repostry on dropbox?

You can and it would be baked up and sherable, but there are problems:

- You can run into Dropbox syncing and git version control clashes
- If you shere the files with a colaborator they are't identified separately to you
- You or a colaborator can completely mess up the git system (mostly by deleating the records in the '.git' folder)
- If you do you can't just re download





# Which platform?

## Bitbucket

- Allows unlimited public or private repositories
- Charges per collaborator over 5 on the repositories
- 1GB storage for large files all repositories, 5GB if using a .ac email



## GitHub

- Has unlimited public repositories
- With unlimited collaborators
- Pay for private repositories
- However as a student or academic you can sign up for free private repositories
- Github is about  $\times 4$  more popular than Bitbucket

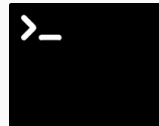
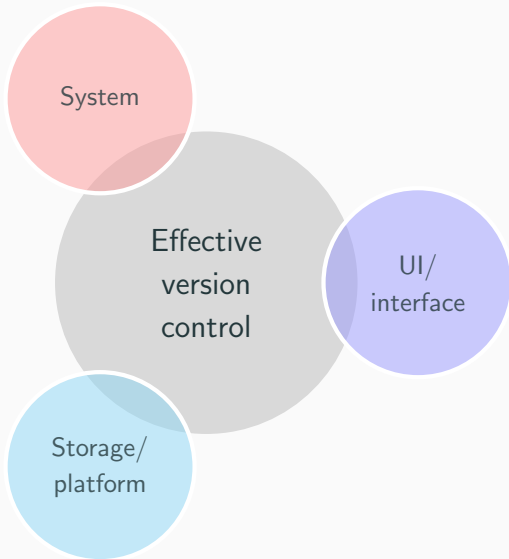


UI



# Which UI?

You can just use git with the terminal but there is a friendlier interface



# Which UI?

## The Terminal

You can just use the terminal with the raw git commands as seen before... but:

- If you are not using it anyway or are happy using it a graphical UI is nice
- You don't need to remember commands or repository names



```
C:\Windows\system32\cmd.exe

C:\Users\dgal\My Documents\GitHub\Presentations\GIT>git status
On branch master
Your branch is up-to-date with 'origin/master'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   Figs/git/Dropbox.png

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   Danny_AC_Taguchi.pdf
    modified:   Danny_AC_Taguchi.tex

Untracked files:
  (use "git add <file>..." to include in what will be committed)

    Figs/git/gitdesktop.png
    Figs/git/terminal.png

C:\Users\dgal\My Documents\GitHub\Presentations\GIT>git commit -m "added image f
or dropbox"
```

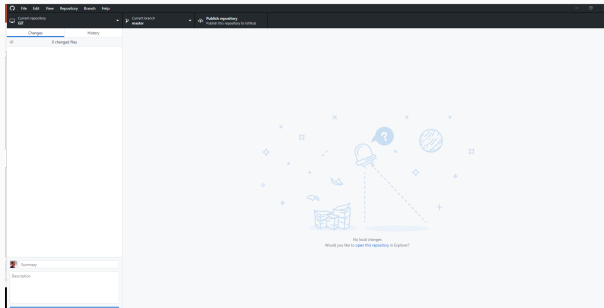
## Which UI?

## Git Desktop

The easiest to use and most popular UI is git desktop:

- This is made by github and integrates very well with github or local (on pc) repositories
- It will work with other hosting platforms but less easily
- Is the easiest to use
- Will install git for windows

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



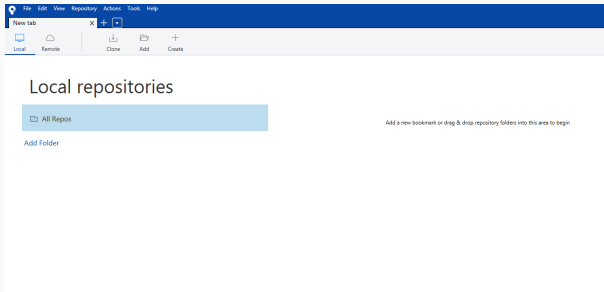
# Which UI?

## Sourcetree

The hosting platform independent:

- Sourcetree is used by atlassian who owns Bitbucket
- It will work with other hosting platforms easily
- It has more complex controls
- Will install git for windows

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



## Using git

---

# How to use git

**This will assume use of githithub and gitdesktop**

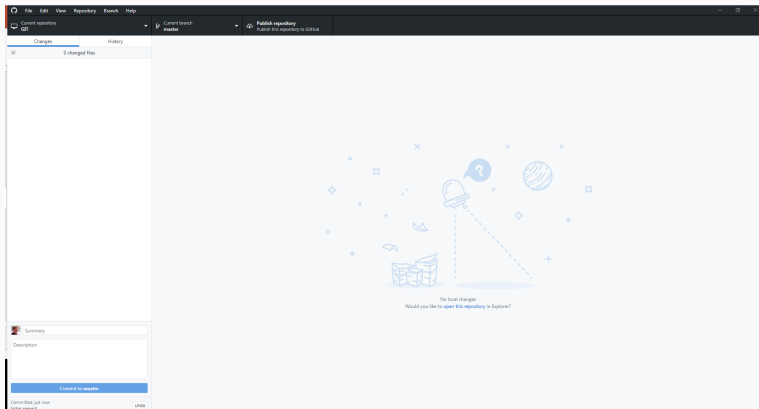
`https://desktop.github.com`

Following is a bref dmonstration of the interface

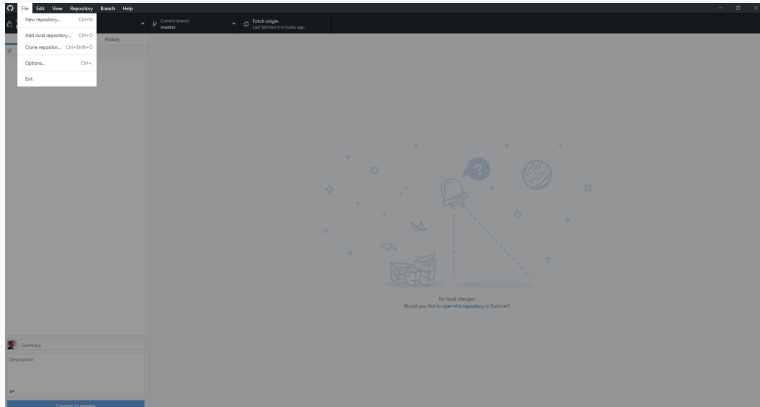




# Adding a new repository

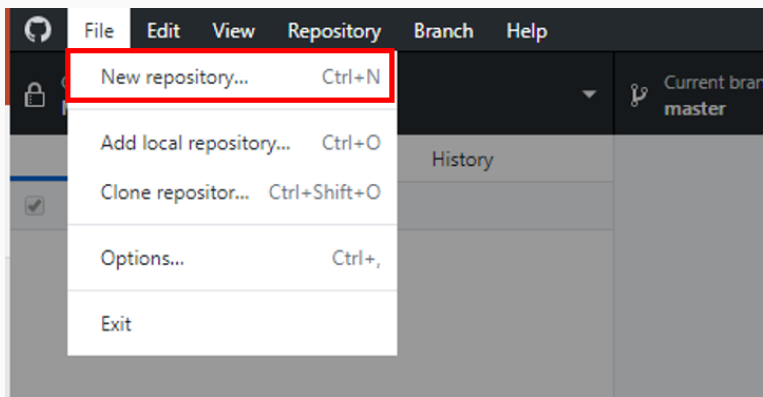


# Adding a new repository



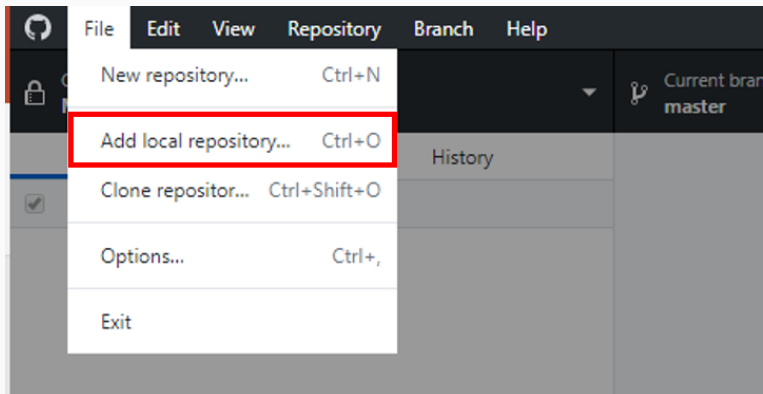
## Adding a new repository

This makes a totally new repository (with no '.git' folder)



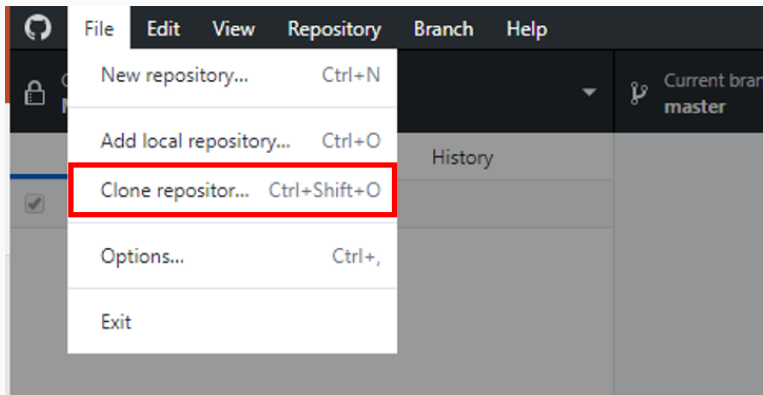
## Adding a new repository

adds an existing repository (with a '.git' folder)



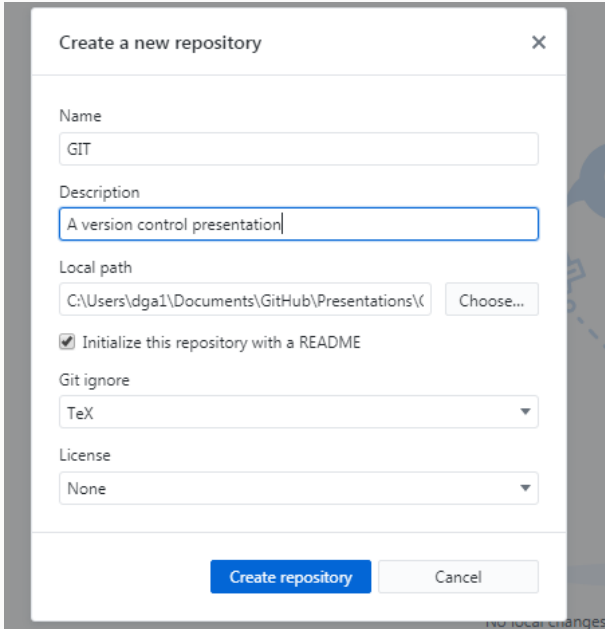
## Adding a new repository

Copys an  
online  
reposi-  
tory to  
the PC



# Adding a new repository

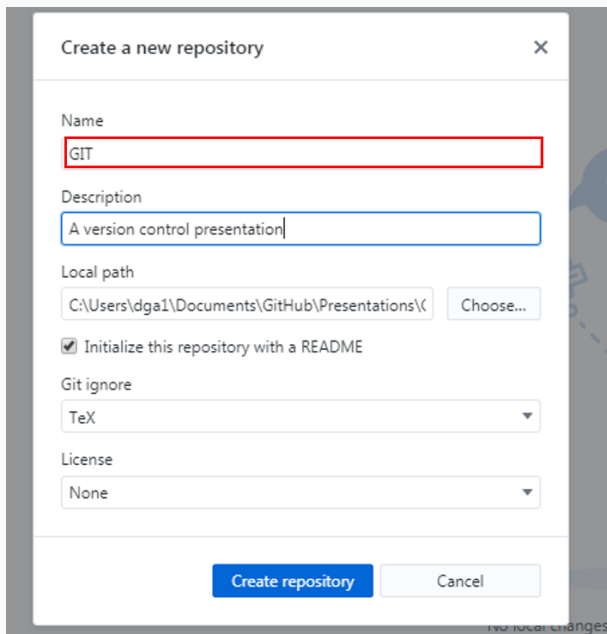
new repository  
menu



The screenshot shows a 'Create a new repository' dialog box with the following fields and options:

- Name:** A text input field containing 'GIT'.
- Description:** A text input field containing 'A version control presentation'.
- Local path:** A text input field containing 'C:\Users\dga1\Documents\GitHub\Presentations\(' and a 'Choose...' button to the right.
- Initialize this repository with a README:** A checked checkbox.
- Git ignore:** A dropdown menu showing 'TeX'.
- License:** A dropdown menu showing 'None'.
- Buttons:** 'Create repository' (blue) and 'Cancel' (grey) at the bottom.

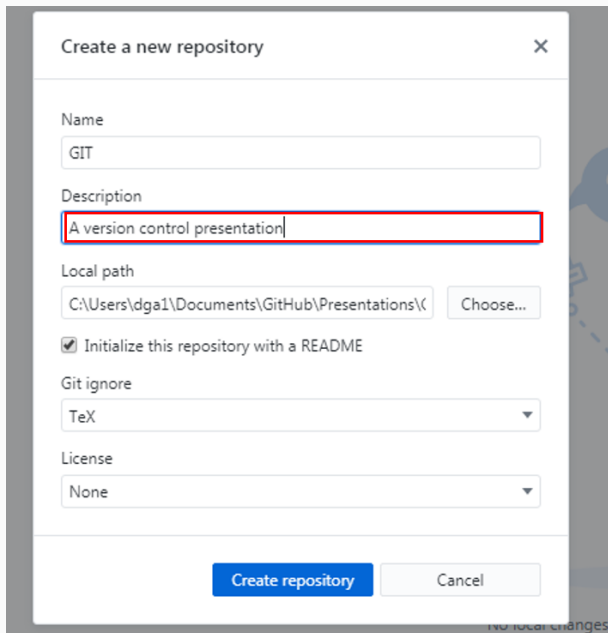
## Adding a new repository



The screenshot shows a 'Create a new repository' dialog box with the following fields and options:

- Name:** A text input field containing 'GIT', which is highlighted with a red rectangular border.
- Description:** A text input field containing 'A version control presentation'.
- Local path:** A text input field containing 'C:\Users\dga1\Documents\GitHub\Presentations\(' and a 'Choose...' button to its right.
- Initialize this repository with a README:** A checked checkbox.
- Git ignore:** A dropdown menu with 'TeX' selected.
- License:** A dropdown menu with 'None' selected.
- Buttons:** 'Create repository' (blue) and 'Cancel' (grey) buttons at the bottom.

## Adding a new repository



The screenshot shows a 'Create a new repository' dialog box with the following fields and options:

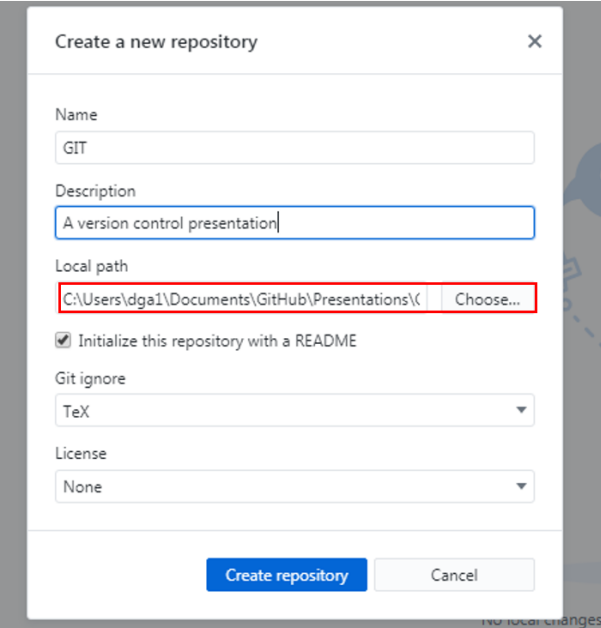
- Name:** A text input field containing the text 'GIT'.
- Description:** A text input field containing the text 'A version control presentation'. This field is highlighted with a red rectangular border.
- Local path:** A text input field containing the path 'C:\Users\dga1\Documents\GitHub\Presentations\'. To the right of this field is a 'Choose...' button.
- Initialize this repository with a README:** A checkbox that is checked.
- Git ignore:** A dropdown menu with 'TeX' selected.
- License:** A dropdown menu with 'None' selected.

At the bottom of the dialog are two buttons: 'Create repository' (a blue button) and 'Cancel' (a grey button).



## Adding a new repository

The folder **in which** to store the git repository folder.

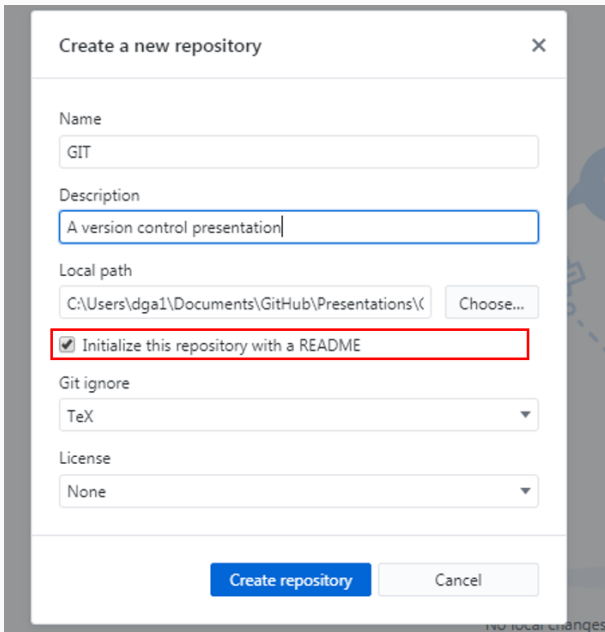


The screenshot shows a 'Create a new repository' dialog box with the following fields and options:

- Name:** A text input field containing 'GIT'.
- Description:** A text input field containing 'A version control presentation'.
- Local path:** A text input field containing 'C:\Users\dga1\Documents\GitHub\Presentations\(' and a 'Choose...' button. This entire row is highlighted with a red rectangle.
- Initialize this repository with a README:** A checked checkbox.
- Git ignore:** A dropdown menu showing 'TeX'.
- License:** A dropdown menu showing 'None'.
- Buttons:** 'Create repository' (blue) and 'Cancel' (grey) at the bottom.

## Adding a new repository

This will initialise the repository with a readme containing the discription.



Create a new repository

Name  
GIT

Description  
A version control presentation

Local path  
C:\Users\dga1\Documents\GitHub\Presentations\(\ Choose...

☒ Initialize this repository with a README

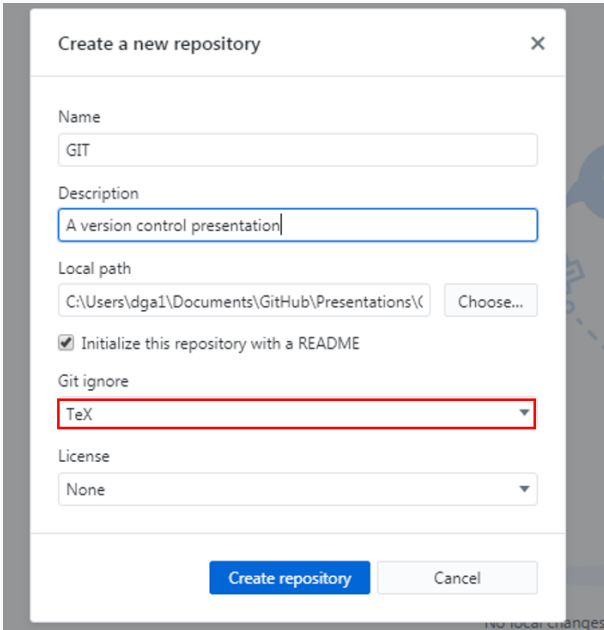
Git ignore  
TeX

License  
None

Create repository Cancel

# Adding a new repository

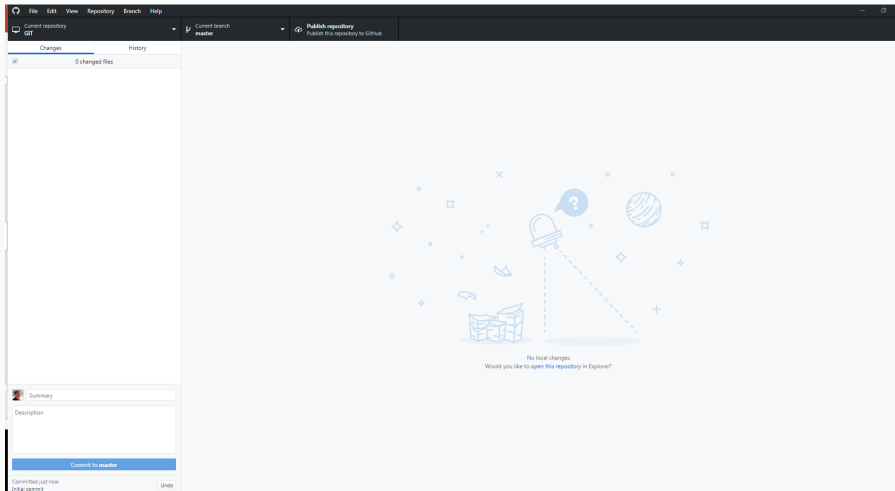
This is a file called  
'`.gitignore`'  
which lists files  
& folders to not  
version control



The screenshot shows a 'Create a new repository' dialog box with the following fields and options:

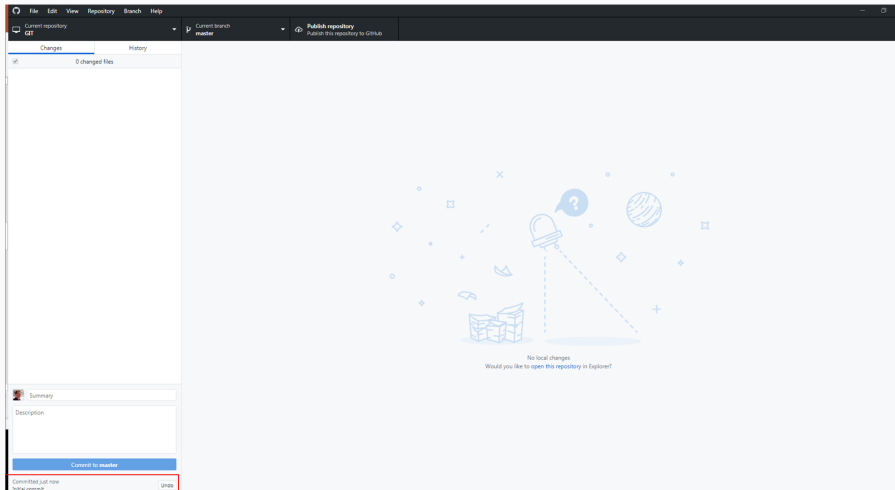
- Name:** A text input field containing 'GIT'.
- Description:** A text input field containing 'A version control presentation'.
- Local path:** A text input field containing 'C:\Users\dga1\Documents\GitHub\Presentations\(' and a 'Choose...' button.
- Initialize this repository with a README:** A checked checkbox.
- Git ignore:** A dropdown menu with 'TeX' selected and highlighted by a red rectangle.
- License:** A dropdown menu with 'None' selected.
- Buttons:** 'Create repository' (blue) and 'Cancel' (gray) buttons at the bottom.

# Using the repostry



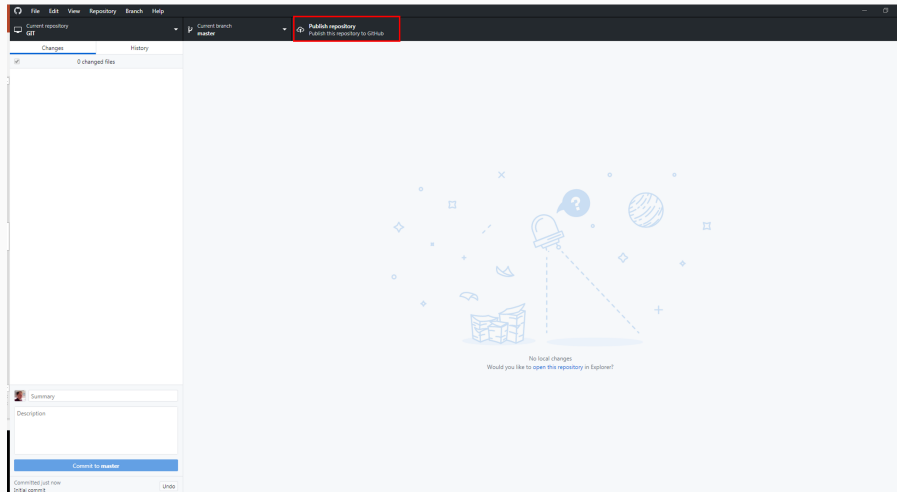
# Using the repostry

Now we have a inshal commit of the readme done



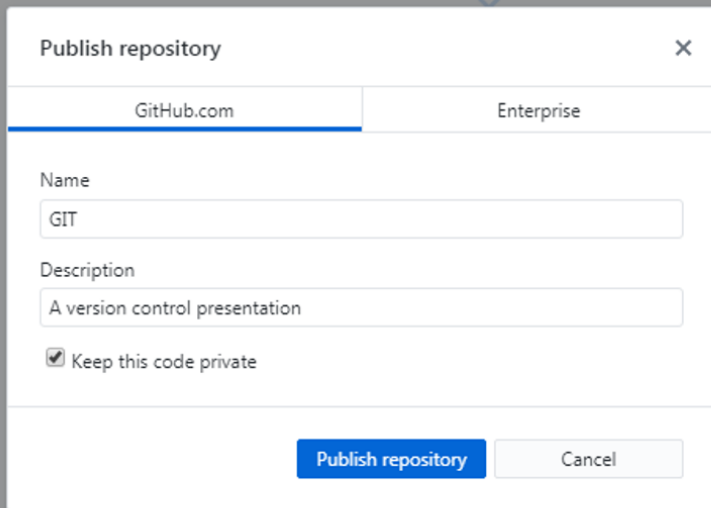
# Using the repostry

We can carry on making local commits or or publish the repostry to github



# Using the repostry

We can publish to github



A screenshot of a 'Publish repository' dialog box. The dialog has a title bar with the text 'Publish repository' and a close button (X). Below the title bar are two tabs: 'GitHub.com' (selected) and 'Enterprise'. The main area contains three input fields: 'Name' with the value 'GIT', 'Description' with the value 'A version control presentation', and a checkbox labeled 'Keep this code private' which is checked. At the bottom right are two buttons: 'Publish repository' (blue) and 'Cancel' (gray).

Publish repository

GitHub.com Enterprise

Name

GIT

Description

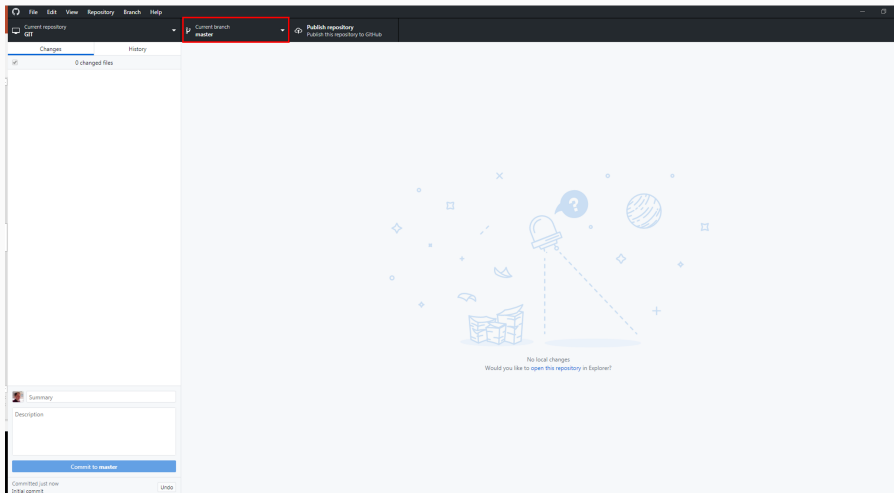
A version control presentation

☒ Keep this code private

Publish repository Cancel

# Using the repostry

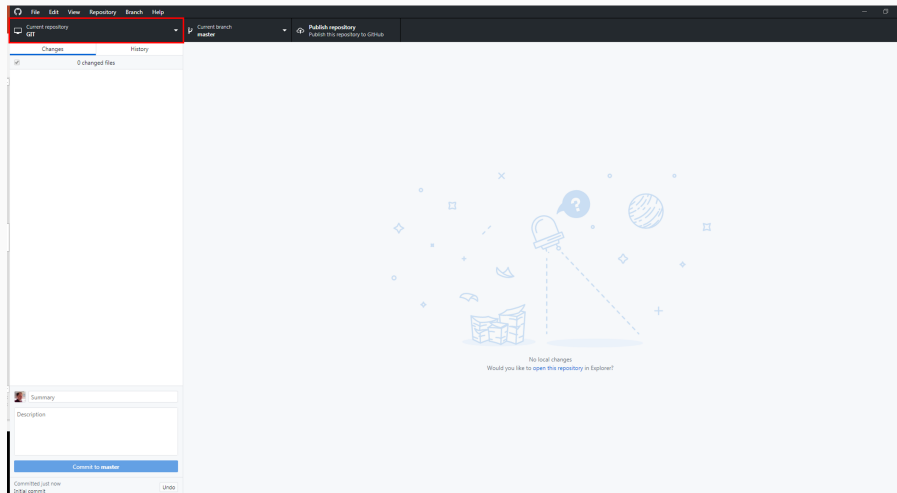
## Make branches





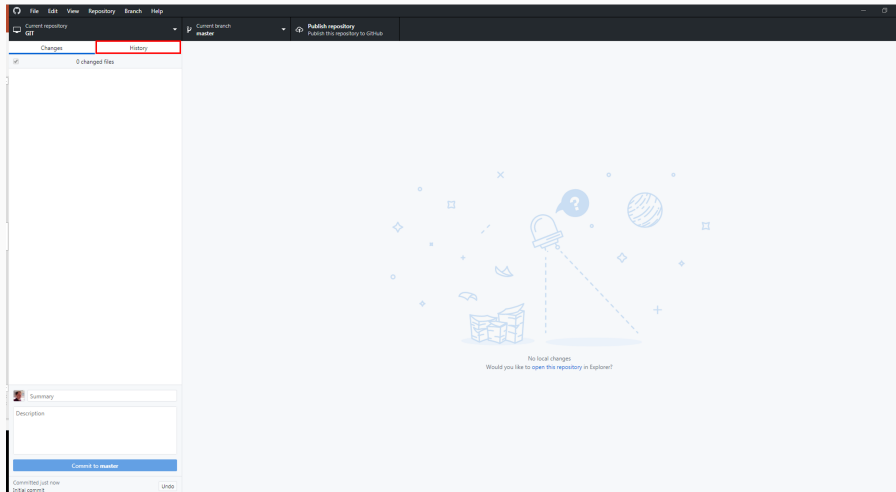
# Using the repostry

## switch to another repository



# Using the repostry

look at the history of this repository



# Using the repostry

look at the history of this repository

The screenshot displays the GitHub web interface for a repository. The top navigation bar includes the repository name, current branch (master), and a 'Push origin' button. Below this, the 'History' tab is selected, showing a list of commits. The most recent commit is titled 'A figure for committing changes on GHD' by Danny George Awty-Carroll, committed just now. Below this, two other commits are listed: 'figures for Git Hub Desktop' (committed 8 minutes ago) and 'Initial commit' (committed 19 minutes ago). To the right, a file diff view for 'Figs\GHD\outline\_08.png' is shown, indicating 1 changed file. The diff view includes a menu bar with 'File', 'Edit', 'View', 'Repository', 'Branch', and 'Help'. The diff shows the file content, with a red box highlighting the 'Change' button in the right-hand pane.

Current repository: GIT

Current branch: master

Push origin: Last fetched 2 minutes ago

Changes | History

A figure for committing changes on GHD  
Danny George Awty-Carroll [dga1] committed just now

figures for Git Hub Desktop  
Danny George Awty-Carroll [dga1] committed 8 minutes ago

Initial commit  
Danny George Awty-Carroll [dga1] committed 19 minutes ago

Figs\GHD\outline\_08.png

1 changed file

Figs\GHD\outline\_08.png

File Edit View Repository Branch Help

Current repository: GIT

Current branch: master

Changes | History

1 changed file

Figs\GHD\outline\_08.png

File Edit View Repository Branch Help

Current repository: GIT

Current branch: master

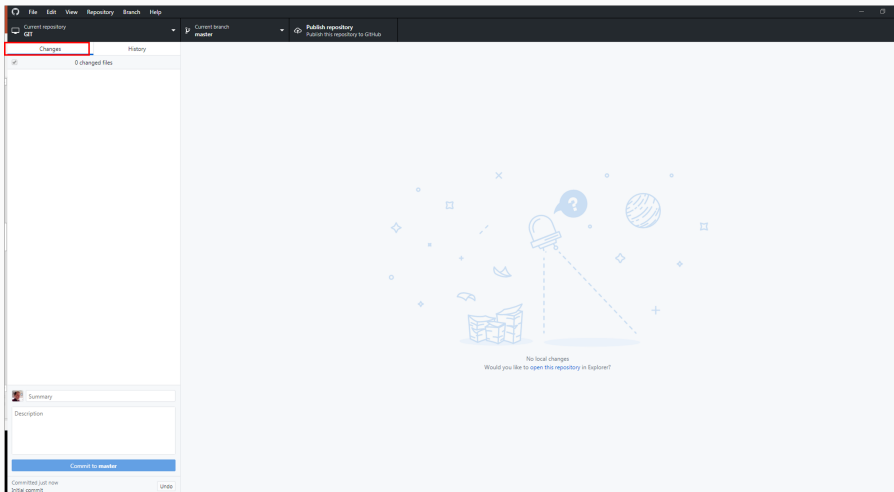
Changes | History

1 changed file

Figs\GHD\outline\_08.png

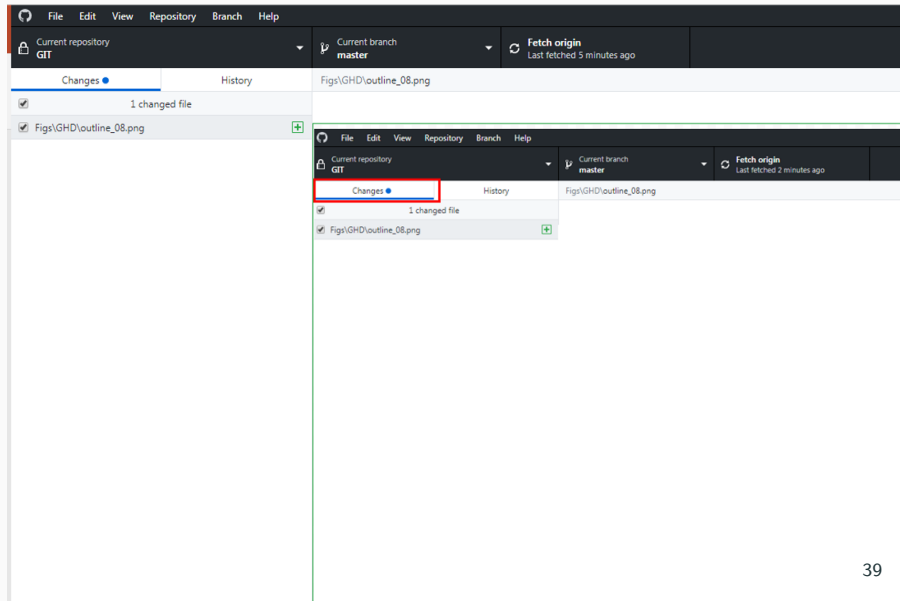
# Using the repostry

we can carry on making local commits or or publish the repostry to github



# Using the repostry

we can carry on making local commits or or publish the repostry to github



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.



