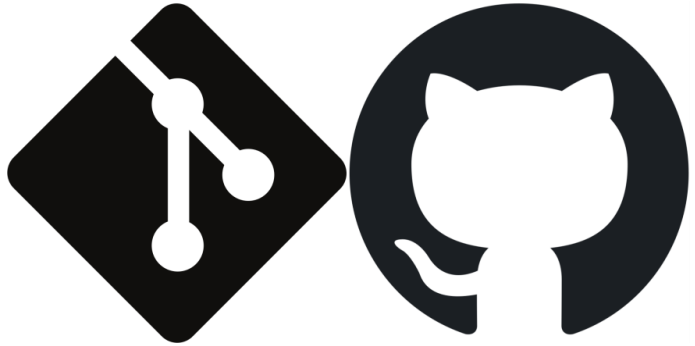
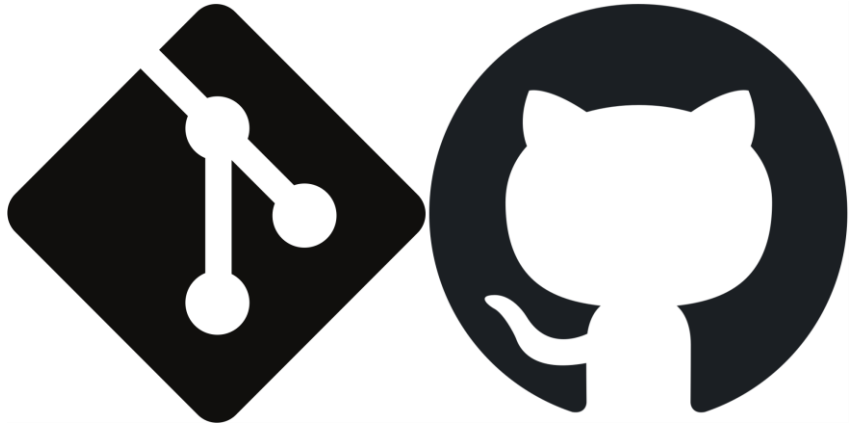
Welcome!



Some initial course info:

- You can find links to our draft schedule and all our course materials through our wiki.
  - https://github.com/CefasRepRes/Git_Training/wiki/Schedule
  - *We will take breaks during the day!*
- Feel free to interrupt and ask questions as we go.
- H&S…



Introduction to Git & GitHub

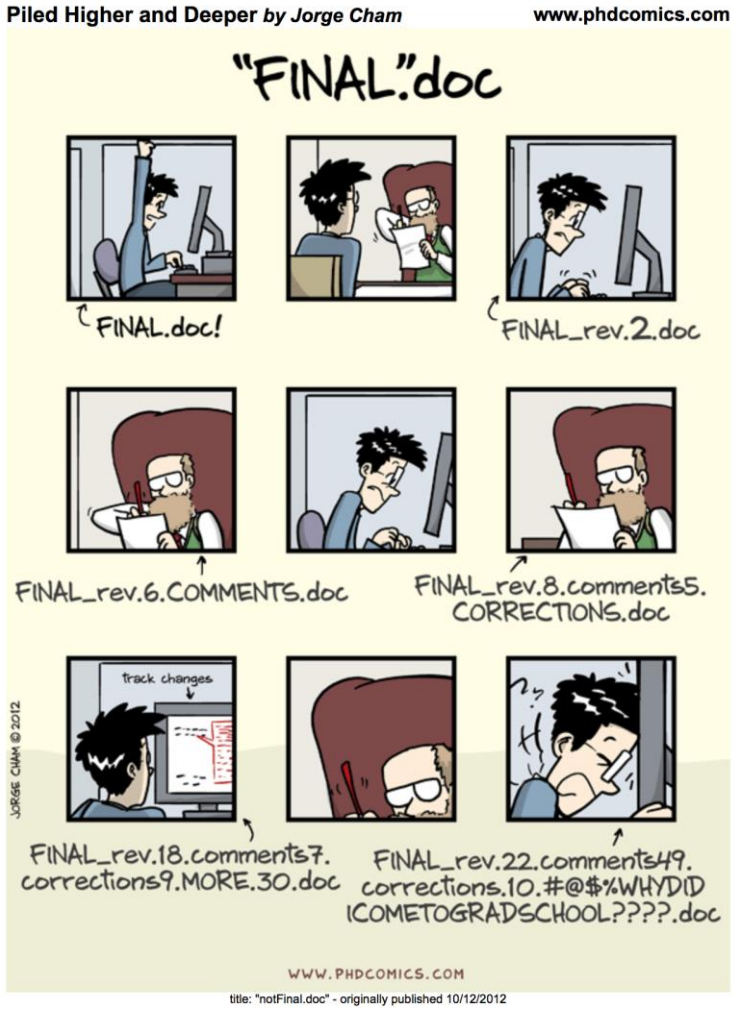
Jennifer Graham
Tiago Silva
David Ryder

Session 1
23rd of June 2023

Why use version control?

- How many people have directories that include files like this?
  - `File_16062020.doc, File_30062020.doc, …`
  - `File_old.doc, File_new.doc`
  - `File.doc.orig`
  - `File.doc, File_test.doc`

- Version control allows you to:
  - Back up your code.
  - Keep track of changes.
  - Share your code with others.
  - Develop code with collaborators.



title: "notFinal.doc" - originally published 10/12/2012

Changes are documented

- As code is backed up, can be confident in testing new ideas.
- Version control software ensures that changes are attributable to individuals.
  - If the code breaks, you know why!
- Code can be shared along with the documentation and file history.
  - Documentation can explain how and why the code should be used.
  - Can create fixed versions/releases, along with DOIs.
    - Sharing now often required for publication.

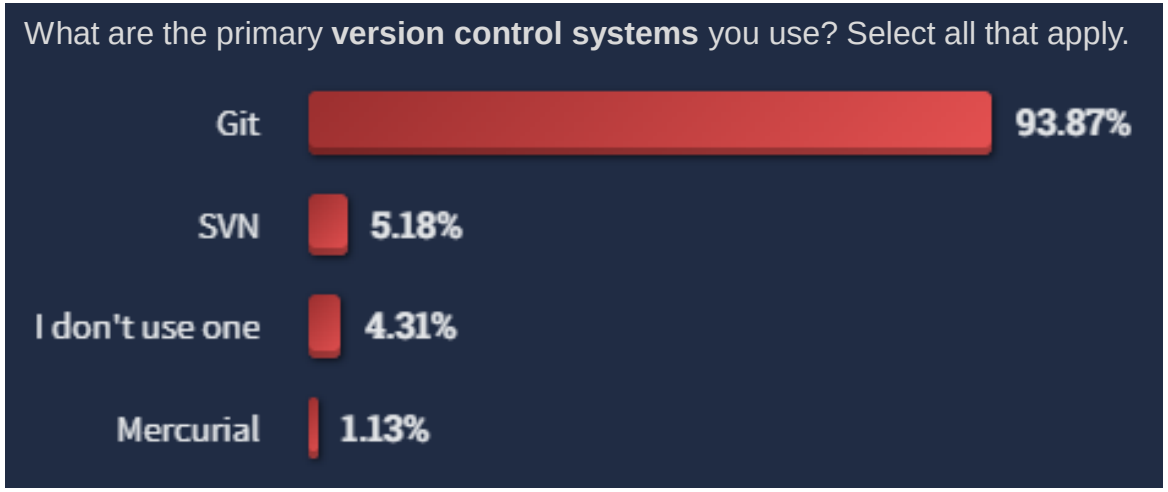
- All the above provides “quality assurance” in the final product.

Version control software

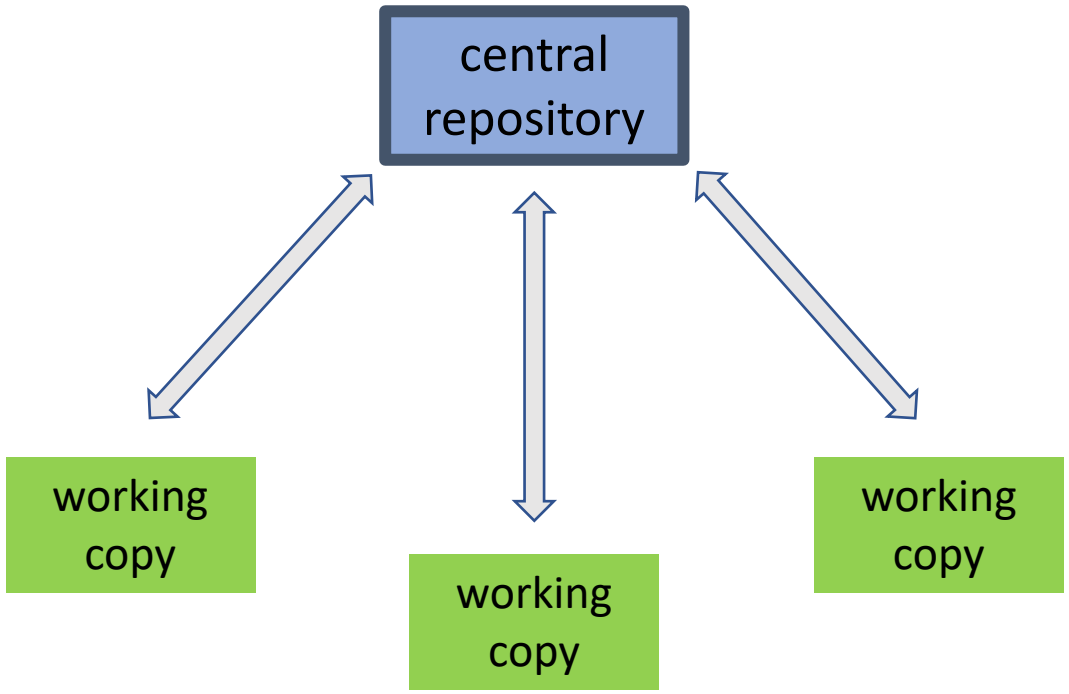
- Many options out there…
  - https://en.wikipedia.org/wiki/List_of_version-control_software
  - All are *agnostic* in terms of the programming language
- Client-server model :: all users share a single code repository.
  - e.g., SVN (subversion)
    - Legacy choice, depending on when project started (many now switching)?
- Distributed model :: all users have their own local repositories. Changes can be shared/merged as a separate step.
  - e.g., **Git**
    - What we're using today!
    - Well supported, large user community, e.g.,

    From Stack Overflow user survey (2022),

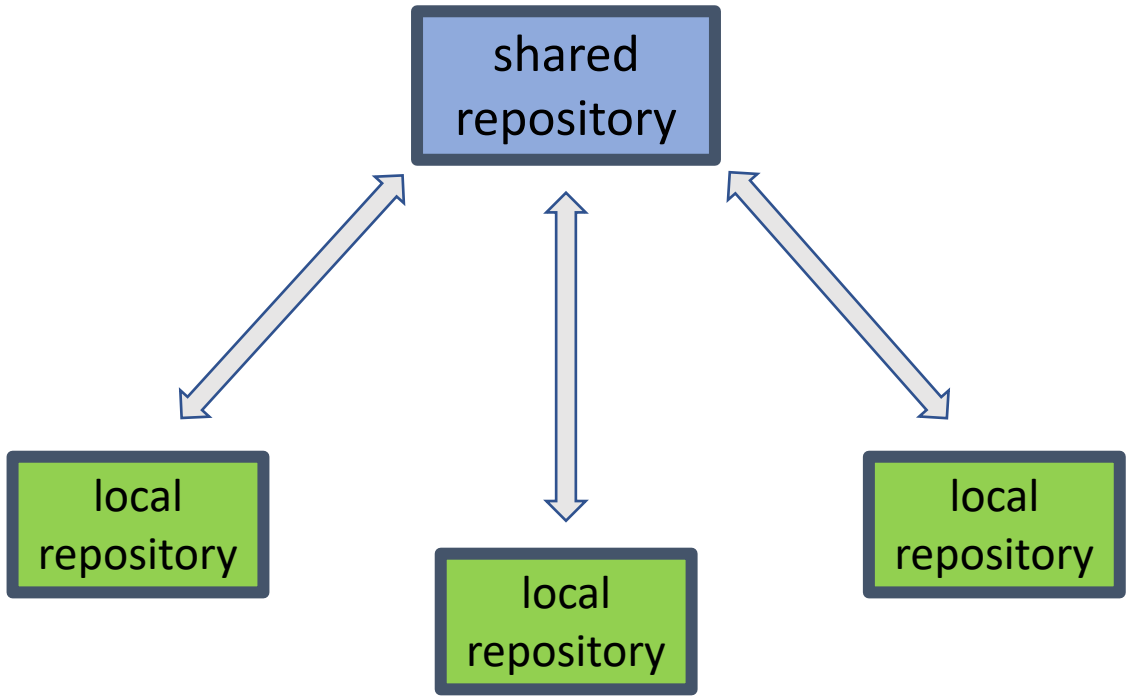
    "No other technology is as widely used as Git."



Central vs Distributed workflow?



Centralised/client-server model (e.g., SVN)



Distributed model (e.g., Git)

So, what are Git & GitHub?

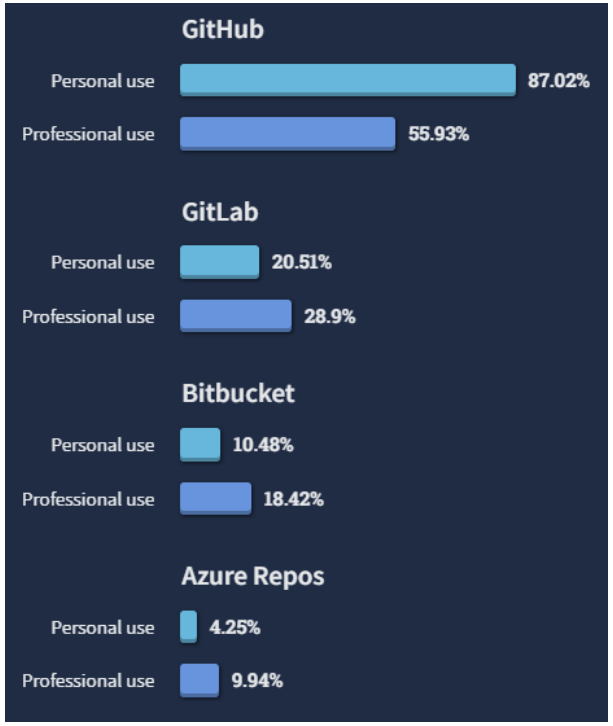
NB. These are <u>two different things</u>

- Git = version control software.
- GitHub = web-based repository hosting service*.

i.e. Git is the software behind the GitHub web service.

You can use Git without GitHub. However, GitHub provides useful tools (especially for sharing your code).

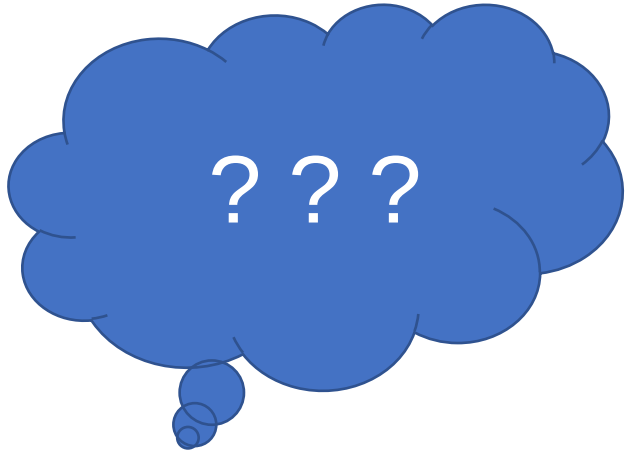
* Other providers exist, but GitHub is the most popular, e.g.:



Useful resources

Git and GitHub are widely used, so there are lots of useful websites, training videos, online courses etc. e.g.,

- GitHub guides & help:
  - https://guides.github.com/
- Git guides:
  - https://git-scm.com/doc
- Git cheat sheet:
  - https://education.github.com/git-cheat-sheet-education.pdf
- E-books:
  - https://www.git-tower.com/learn/git/ebook/en/command-line/introduction
- Webinars:
  - https://www.youtube.com/watch?v=v3Y8c2KMay8
  - https://www.youtube.com/watch?v=ShH1g4I9A54
- Internet searching!



Using Git

- Various options depending on preference:
  - **Command line e.g., Git for windows** (bash shell or GUI)
    - Many command line interfaces available e.g., Windows 10 bash shell, HPC, Linux or Mac terminals.
  - GUI e.g., GitHub desktop
  - Rstudio or pycharm (some built-in functionality, similar to GUI)
  - *[NB. Admin rights not needed to install Git]*

- Whatever option you choose, all the above follow the same basic workflow.
  - Choose whatever works for you e.g., are you more comfortable with a GUI or command line?
  - Will initially take some practice, but using git can (should) become a regular activity.

How not to use git… https://xkcd.com/1597/

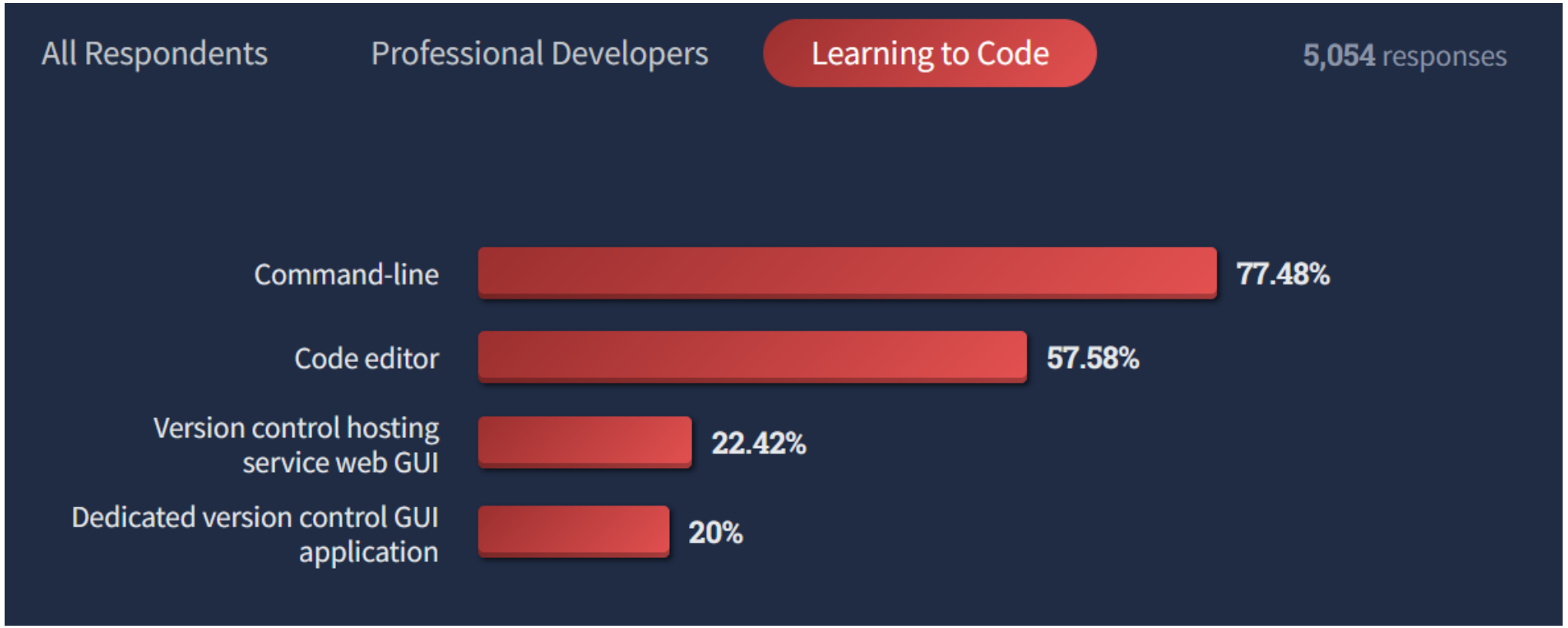
Why are we using the command line?

- **Clearly shows the commands being used at each step in the workflow.**
  - Command name matches exactly what is used on the command line.

- Can apply the workflow to a chosen GUI at a later date (e.g., Rstudio, pycharm, or GitHub Desktop).

- Not all commands are available in every GUI, so always have the option of moving back to the command line.

Why are we using the command line?



https://survey.stackoverflow.co/2022/#technology-version-control

Introduction to command line…

- Link to other slides ->

Basic Git workflow

- Create a local repository (code directory).
- Add/edit a file.
- Commit your changes (i.e. document changes).
- Then repeat!

Optional (but recommended):

- Push changes to remote repository on GitHub (back-up/sharing).

Creating a repository…

- I already have a folder, how do I start tracking my changes?

```
JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world
$ ls
hello_world.py

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world
$
```


MINGW64:/c/Users/JG10/OneDrive/GitHub/Repos/hello_world
JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world
$ ls
hello_world.py
JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world
$ git init
git init
:: turn current directory into git repository.

```
JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world
$ ls
hello_world.py

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world
$ git init
Initialized empty Git repository in C:/Users/JG10/OneDrive - CEFAS/GitHub/Repos
/hello_world/.git/

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$
```

```
JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world
$ ls
hello_world.py

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world
$ git init
Initialized empty Git repository in C:/Users/JG10/OneDrive - CEFAS/GitHub/Repos
/hello_world/.git/

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git status
```

git status

:: show which files have been tracked or modified.

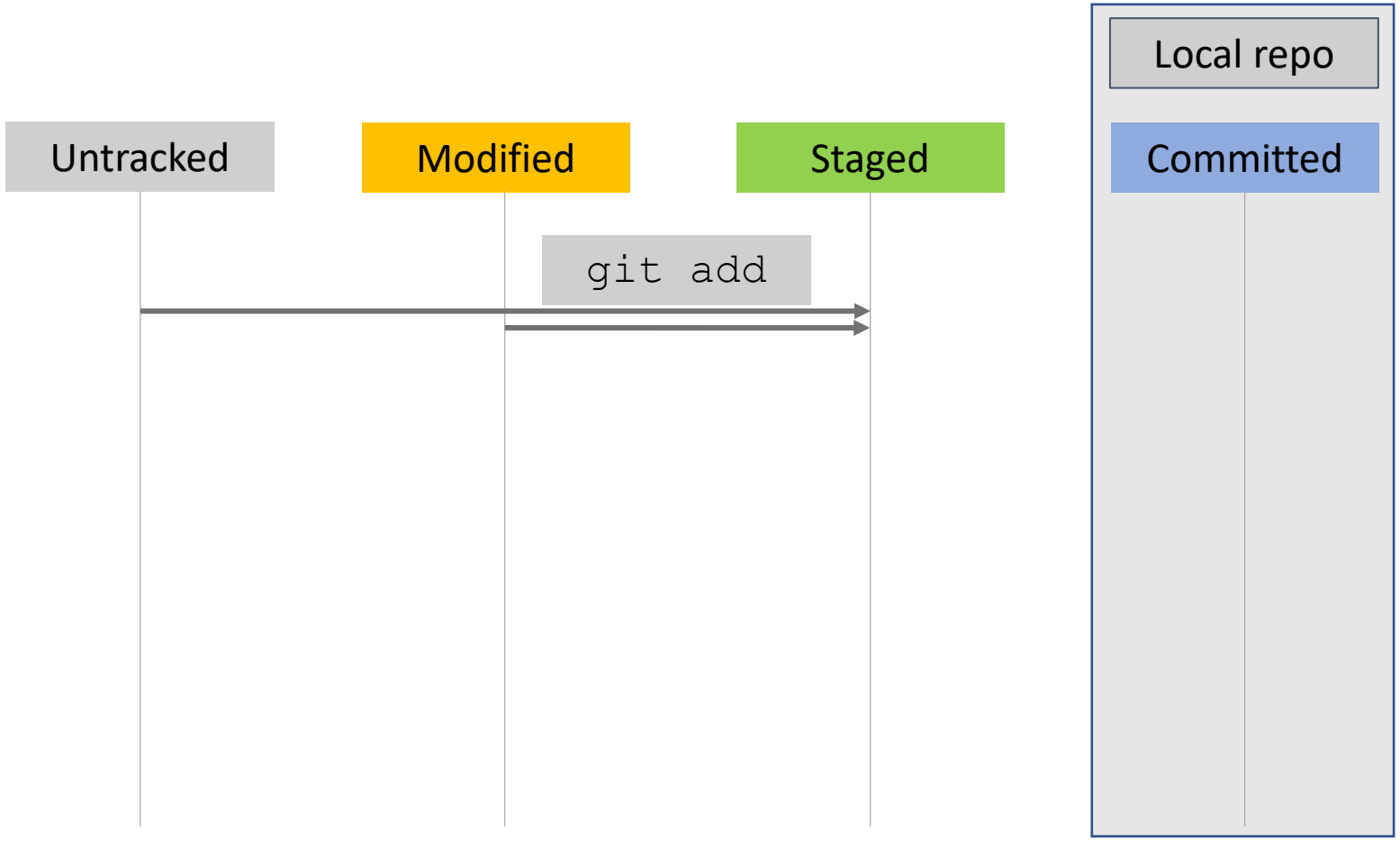
```
JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world
$ ls
hello_world.py

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world
$ git init
Initialized empty Git repository in C:/Users/JG10/OneDrive - CEFAS/GitHub/Repos
/hello_world/.git/

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git status
On branch main

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$
```

The “Staging area”

- Git doesn’t have to track all files in a repository.

- There are three “states” of tracked files in a Git repository.
  - Working directory :: current version of files, containing any modifications.
  - Staging area :: snapshot of files/modifications that you plan to commit.
  - Repository/Git directory :: contains the commit history for all files in the repository.

- You can check the state of each file at any point using “`git status`”


Local repo
Untracked
Modified
Staged
Committed
git add

```
JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world
$ ls
hello_world.py

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world
$ git init
Initialized empty Git repository in C:/Users/JG10/OneDrive - CEFAS/GitHub/Repos
/hello_world/.git/

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git status
On branch main

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git add hello_world.py
```

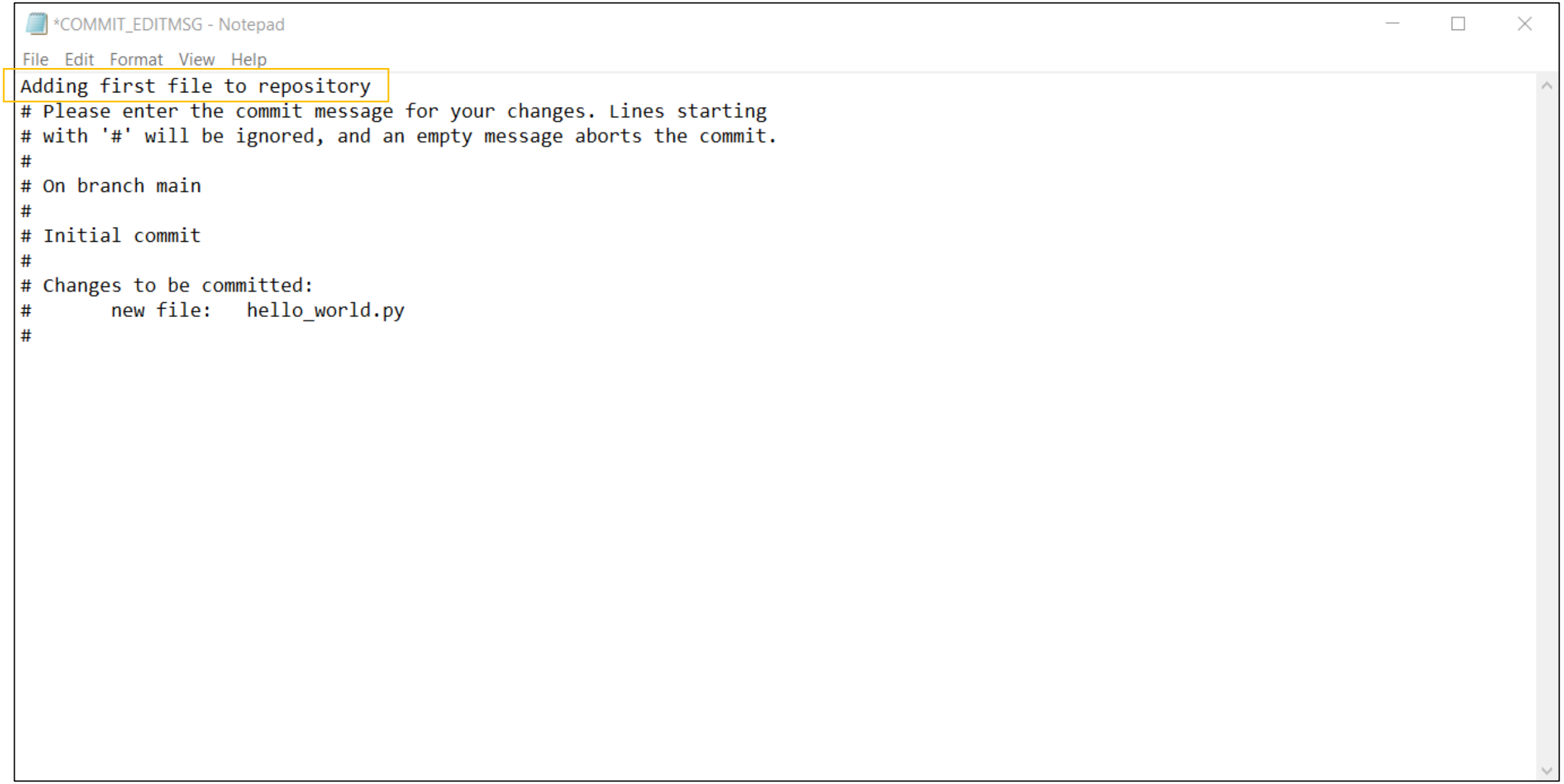
```
git add <file>
:: stage the file for
commit i.e. track changes.
```

```
JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world
$ ls
hello_world.py

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world
$ git init
Initialized empty Git repository in C:/Users/JG10/OneDrive - CEFAS/GitHub/Repos
/hello_world/.git/

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git status
On branch main

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git add hello_world.py
warning: LF will be replaced by CRLF in hello_world.py.
The file will have its original line endings in your working directory

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$
```

```
JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git add hello_world.py
warning: LF will be replaced by CRLF in hello_world.py.
The file will have its original line endings in your working directory

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   hello_world.py


JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$
```

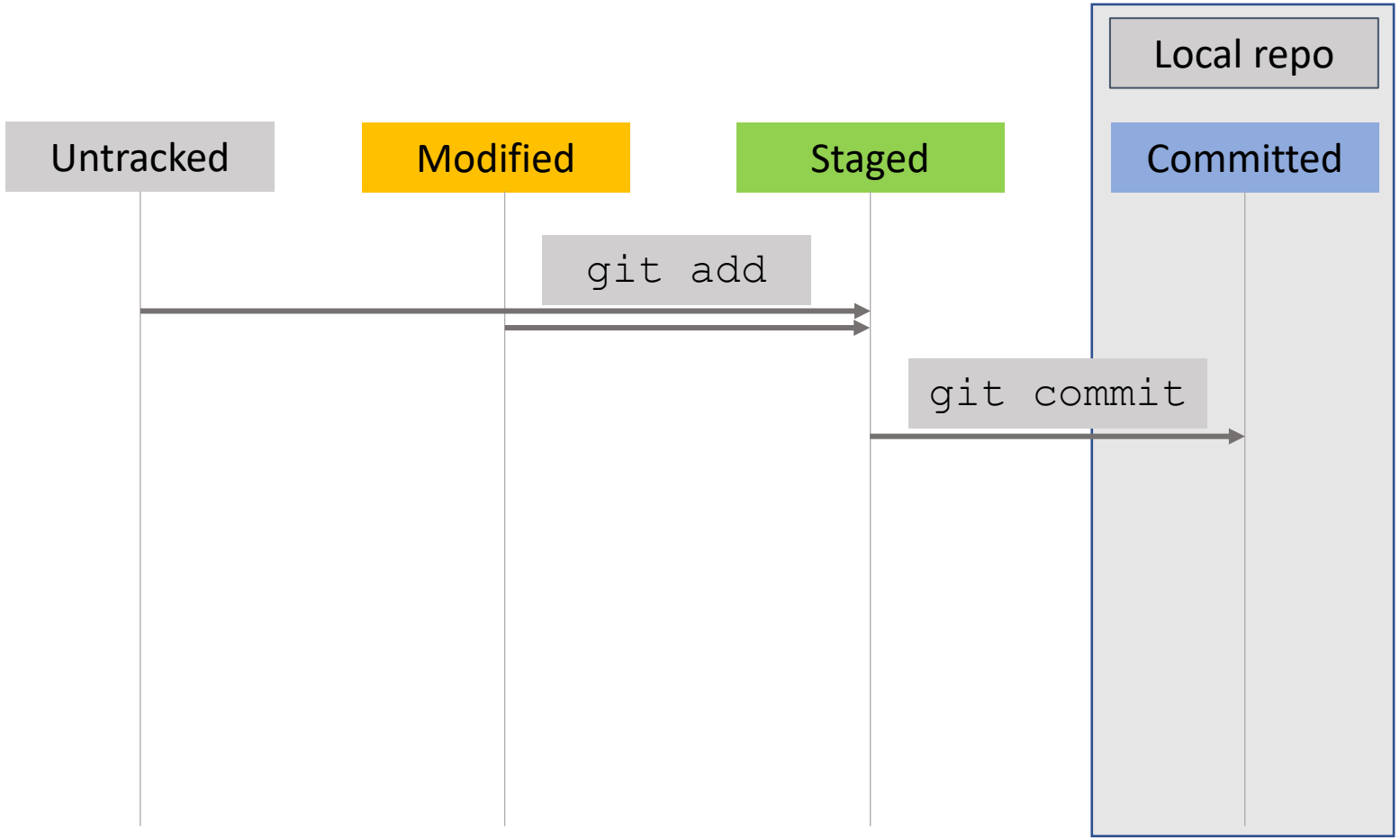
```
JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git add hello_world.py
warning: LF will be replaced by CRLF in hello_world.py.
The file will have its original line endings in your working directory

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   hello_world.py


JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git commit
```

```
git commit
:: confirm changes with message/explanation.
```


*COMMIT_EDITMSG - Notepad
File Edit Format View Help
Adding first file to repository
Please enter the commit message for your changes. Lines starting
with '#' will be ignored, and an empty message aborts the commit.
#
On branch main
#
Initial commit
#
Changes to be committed:
#	new file: hello_world.py
#

```
JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git add hello_world.py
warning: LF will be replaced by CRLF in hello_world.py.
The file will have its original line endings in your working directory

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   hello_world.py


JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git commit
hint: Waiting for your editor to close the file... unix2dos: converting file C:/Users/JG10/OneDrive -
AS/GitHub/Repos/hello_world/.git/COMMIT_EDITMSG to DOS format...
dos2unix: converting file C:/Users/JG10/OneDrive - CEFAS/GitHub/Repos/hello_world/.git/COMMIT_EDITMSG
Unix format...
[main (root-commit) a59cb3f] Adding first file to repository
 1 file changed, 9 insertions(+)
 create mode 100644 hello_world.py

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$
```

```
JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git add hello_world.py
warning: LF will be replaced by CRLF in hello_world.py.
The file will have its original line endings in your working directory

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   hello_world.py


JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git commit
hint: Waiting for your editor to close the file... unix2dos: converting file C:/Users/JG10/OneDrive -
AS/GitHub/Repos/hello_world/.git/COMMIT_EDITMSG to DOS format...
dos2unix: converting file C:/Users/JG10/OneDrive - CEFAS/GitHub/Repos/hello_world/.git/COMMIT_EDITMSG
Unix format...
[main (root-commit) a59cb3f] Adding first file to repository
 1 file changed, 9 insertions(+)
 create mode 100644 hello_world.py

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git status
On branch main
nothing to commit, working tree clean
```


Local repo
Untracked
Modified
Staged
Committed
git add
git commit

Committing changes

- Commit your change.
  - You will be prompted for a commit message.
  - Make a concise, meaningful description of the changes and why they were made.
  - Best practice is to stage only associated modifications in each commit.

- How often should you commit?
  - More often than you think - whenever you make a change that works?
  - The more often you commit, the easier it is to document (with short, meaningful messages), or roll-back if needed.
  - Avoid committing untested or unfinished modifications!

https://xkcd.com/1296/

```
git init                :: turn current directory into git repository.
git add <file>          :: stage the file for commit i.e., track changes.
git commit              :: confirm changes with message/explanation.


git status              :: show which files have been tracked or modified.
```

MINGW64:/c/Users/JG10/OneDrive/GitHub/Repos/hello_world

```
JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ ls
README.md  hello_world.py

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        README.md

nothing added to commit but untracked files present (use "git add" to track)

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git add README.md
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git commit
hint: Waiting for your editor to close the file... unix2dos: converting file C:/Users/JG10/OneDrive - CEF
AS/GitHub/Repos/hello_world/.git/COMMIT_EDITMSG to DOS format...
dos2unix: converting file C:/Users/JG10/OneDrive - CEFAS/GitHub/Repos/hello_world/.git/COMMIT_EDITMSG to
Unix format...
[main a5f1734] Adding README file
 1 file changed, 1 insertion(+)
 create mode 100644 README.md

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git status
On branch main
nothing to commit, working tree clean
```

Create a new file, add and commit …

MINGW64:/c/Users/JG10/OneDrive/GitHub/Repos/hello_world

```
JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ ls
README.md  hello_world.py

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ vi hello_world.py
```

Edit files and make changes …

MINGW64:/c/Users/JG10/OneDrive/GitHub/Repos/hello_world

```
'''
My first code.

@author JGraham
created 11/06/21
'''

hello = 'Hello wold'
print(hello)
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
hello_world.py [unix] (15:32 11/06/2021)                    8,18
-- INSERT --
```

```
JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ ls
README.md  hello_world.py

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ vi hello_world.py

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git diff hello_world.py
```

```
git diff <file>
:: show difference between
current file and last
commit.
```

```
JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ ls
README.md  hello_world.py

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ vi hello_world.py

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git diff hello_world.py
warning: LF will be replaced by CRLF in hello_world.py.
The file will have its original line endings in your working directory
diff --git a/hello_world.py b/hello_world.py
index 0f33ce4..e779e75 100644
--- a/hello_world.py
+++ b/hello_world.py
@@ -5,5 +5,5 @@ My first code.
 created 11/06/21
 '''

-hello = 'Hello wold'
+hello = 'Hello world'
 print(hello)

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$
```

git diff <file>

:: show difference between current file and last commit.

```
-hello = 'Hello wold'
+hello = 'Hello world'
 print(hello)

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ vi README.md

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git diff README.md
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory
diff --git a/README.md b/README.md
index eb6d976..0d08e3c 100644
--- a/README.md
+++ b/README.md
@@ -1 +1 @@
-This is my first repository.
+This is Jenny's first repository.

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$
```

```
-hello = 'Hello wold'
+hello = 'Hello world'
 print(hello)

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ vi README.md

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git diff README.md
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory
diff --git a/README.md b/README.md
index eb6d976..0d08e3c 100644
--- a/README.md
+++ b/README.md
@@ -1 +1 @@
-This is my first repository.
+This is Jenny's first repository.

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git status
On branch main
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md
        modified:   hello_world.py

no changes added to commit (use "git add" and/or "git commit -a")

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$
```

```
JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git add hello_world.py
warning: LF will be replaced by CRLF in hello_world.py.
The file will have its original line endings in your working directory

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$
```

```
JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git add hello_world.py
warning: LF will be replaced by CRLF in hello_world.py.
The file will have its original line endings in your working directory

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello_world.py

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


JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$
```

Don’t have to add all files to commit.

```
JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git add hello_world.py
warning: LF will be replaced by CRLF in hello_world.py.
The file will have its original line endings in your working directory

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello_world.py

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


JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git commit
hint: Waiting for your editor to close the file... unix2dos: converting file C:/Users/JG10/OneDrive - CEFAS/
Hub/Repos/hello_world/.git/COMMIT_EDITMSG to DOS format...
dos2unix: converting file C:/Users/JG10/OneDrive - CEFAS/GitHub/Repos/hello_world/.git/COMMIT_EDITMSG to Uni
ormat...
[main 832863d] Fixing typo
 1 file changed, 1 insertion(+), 1 deletion(-)

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$
```

```
JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git status
On branch main
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$
```

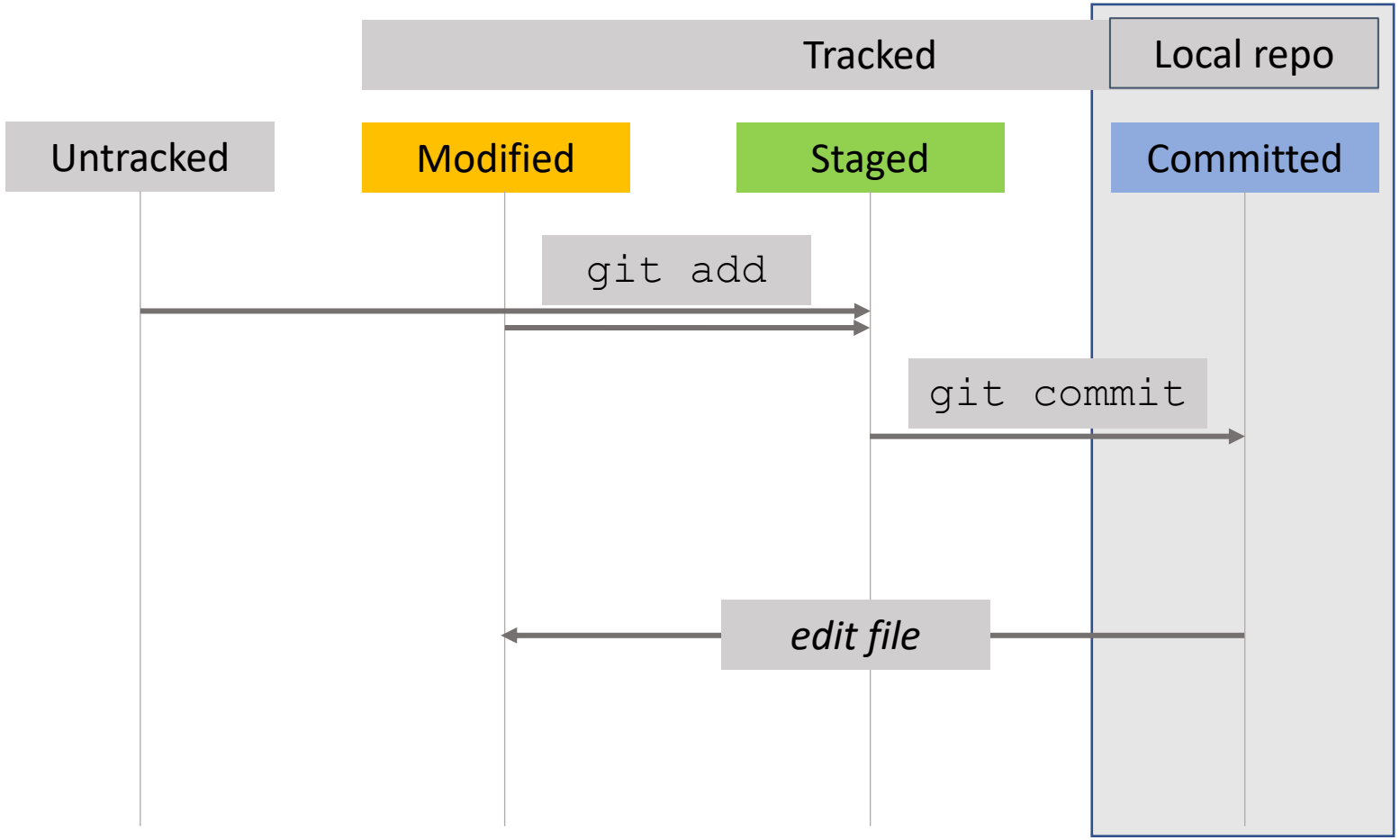
```
JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git status
On branch main
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git log
```

```
git log <file>
:: show commit history
[for file]
```

```
JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git status
On branch main
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git log
commit 832863dd3215b30c3bb7d09eb6ffff229dbff113 (HEAD -> main)
Author: jenniferagraham <jennifer.graham@cefas.co.uk>
Date:   Fri Jun 11 15:47:28 2021 +0100

    Fixing typo

commit a5f173410ae2855cddc0999d58c0e3dfca54a3cf
Author: jenniferagraham <jennifer.graham@cefas.co.uk>
Date:   Fri Jun 11 15:41:13 2021 +0100

    Adding README file

commit a59cb3fc93e0b2efbe607c79340c13549a6cba6e
Author: jenniferagraham <jennifer.graham@cefas.co.uk>
Date:   Fri Jun 11 15:37:05 2021 +0100

    Adding first file to repository

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$
```

```
git log <file>
:: show commit history
[for file]
```

```
git init                :: turn current directory into git repository.
git add <file>          :: stage the file for commit i.e., track changes.
git commit              :: confirm changes with message/explanation.


git status              :: show which files have been tracked or modified.
git diff <file>         :: show difference between current file and last commit.
git log <file>          :: show commit history [for file]
```


Tracked
Local repo
Untracked
Modified
Staged
Committed
git add
git commit
edit file

Aside: repository etiquette

- Git & GitHub are primarily for code/methods -> NOT data storage.
  - Wastes space in the repository, and changes to large files can't be tracked anyway!
  - However, paths/descriptions of data required should be included, either in the code or associated README files.
    - Example data files could be useful when sharing/releasing code.
  - Also, for *small* text or csv files, you may still find git & GitHub useful for version control.

- Give repositories a meaningful name.
  - Choosing meaningful and consistent names will help when searching for repositories.
  - Consider how many repositories could be called "Data_Processing" or "python"?
    - What data is being processed and why? What is the code doing?
  - Consider adding initials as a prefix or suffix
    - Can distinguish between personal "working directories" vs. collaborative projects?
      - e.g. `jag_python_plotting` ?
  - Discuss with others to determine what scripts need to be included as collaborative, central repositories (avoid duplication).

Try it yourself… [15 min]

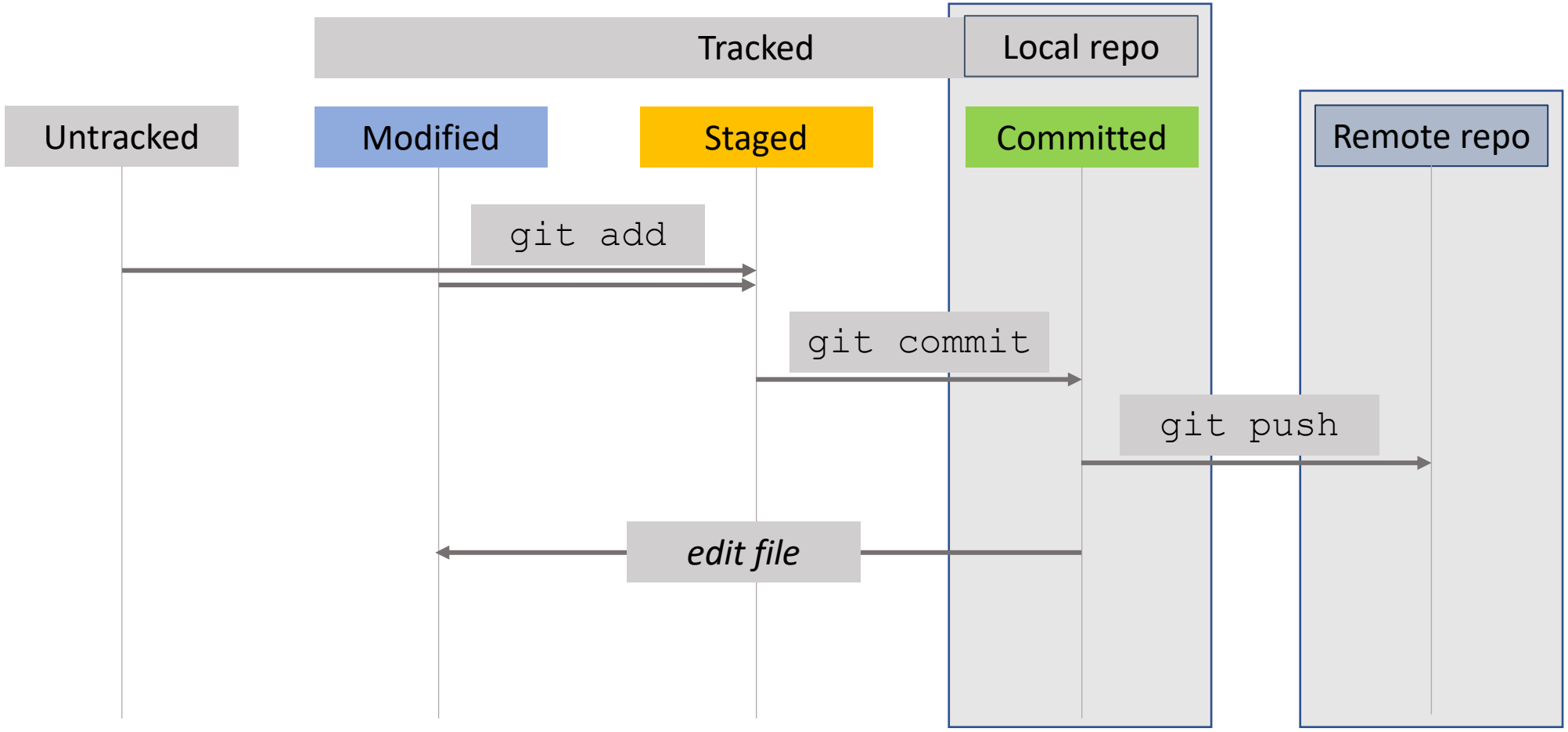
1. Create your own repository locally:
   - Create a directory, e.g., named “My_First_Repo”.
   - Use “git init” to turn it into a repository.
2. Follow the basic workflow to add a simple txt file, make some edits, and then commit changes as you go…

```
git init          :: turn current directory into git repository.
git add <file>    :: stage the file for commit i.e. track changes.
git commit        :: confirm changes with message/explanation.

git status        :: show which files have been tracked or modified.
git diff <file>:: show difference between current file and last commit.
git log <file>    :: show commit history [for file]
```

Syncing/publishing changes

- Nothing will be visible on GitHub until you “push” your commits.

- You don’t need to “push” every commit (they are still recorded), but remember to do so regularly.
  - Make sure changes are visible for others to see & use.
  - Back-up your code.


Tracked
Local repo
Untracked
Modified
Staged
Committed
Remote repo
git add
git commit
git push
edit file

```
JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git status
On branch main
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git push
```

```
JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git status
On branch main
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git push
fatal: No configured push destination.
Either specify the URL from the command-line or configure a remote repository using

    git remote add <name> <url>

and then push using the remote name

    git push <name>


JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$
```

GitHub: Where the world builds s

github.com

Why GitHub? Team Enterprise Explore Marketplace Pricing

Search GitHub /

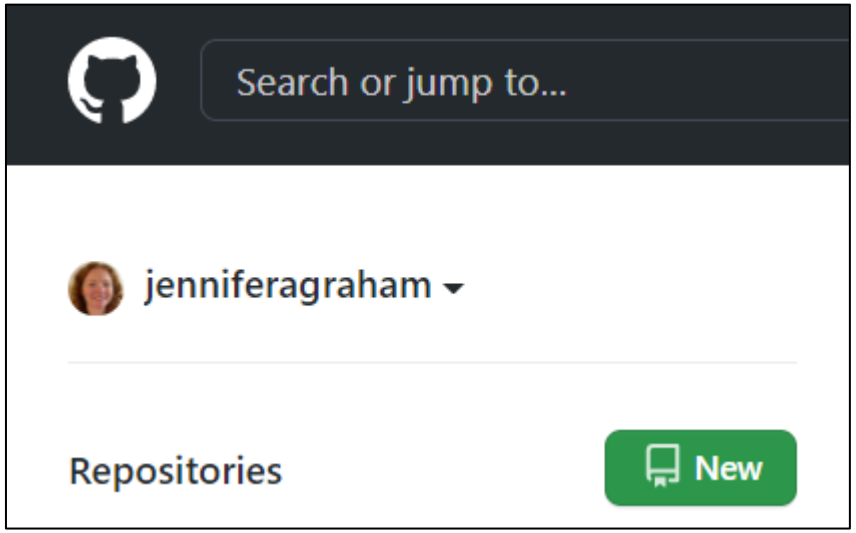
Sign in Sign up

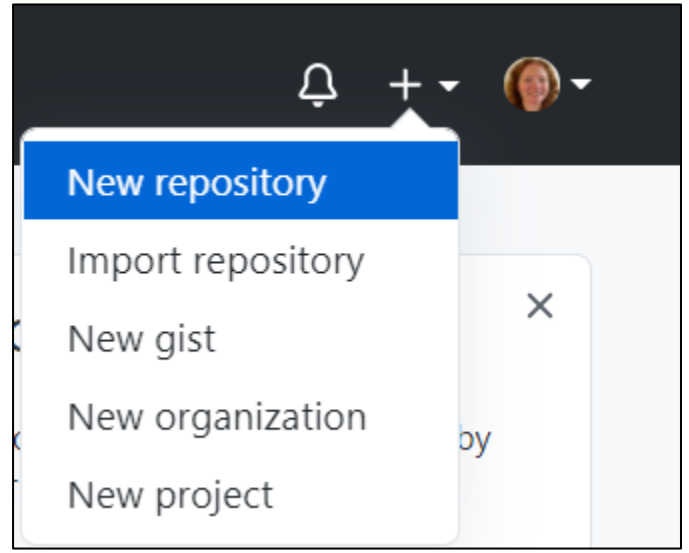
Where the world builds software

Millions of developers and companies build, ship, and maintain their software on GitHub—the largest and most advanced development platform in the world.

Email address

Sign up for GitHub


Search or jump to...
jenniferagraham
Repositories
New


New repository
Import repository
New gist
New organization
New project

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
Import a repository.

Required fields are marked with an asterisk (*).

Owner *

jenniferagraham /

Repository name *

Filter...

jenniferagraham

CefasRepRes

ueapy

morable. Need inspiration? How about symmetrical-goggles ?

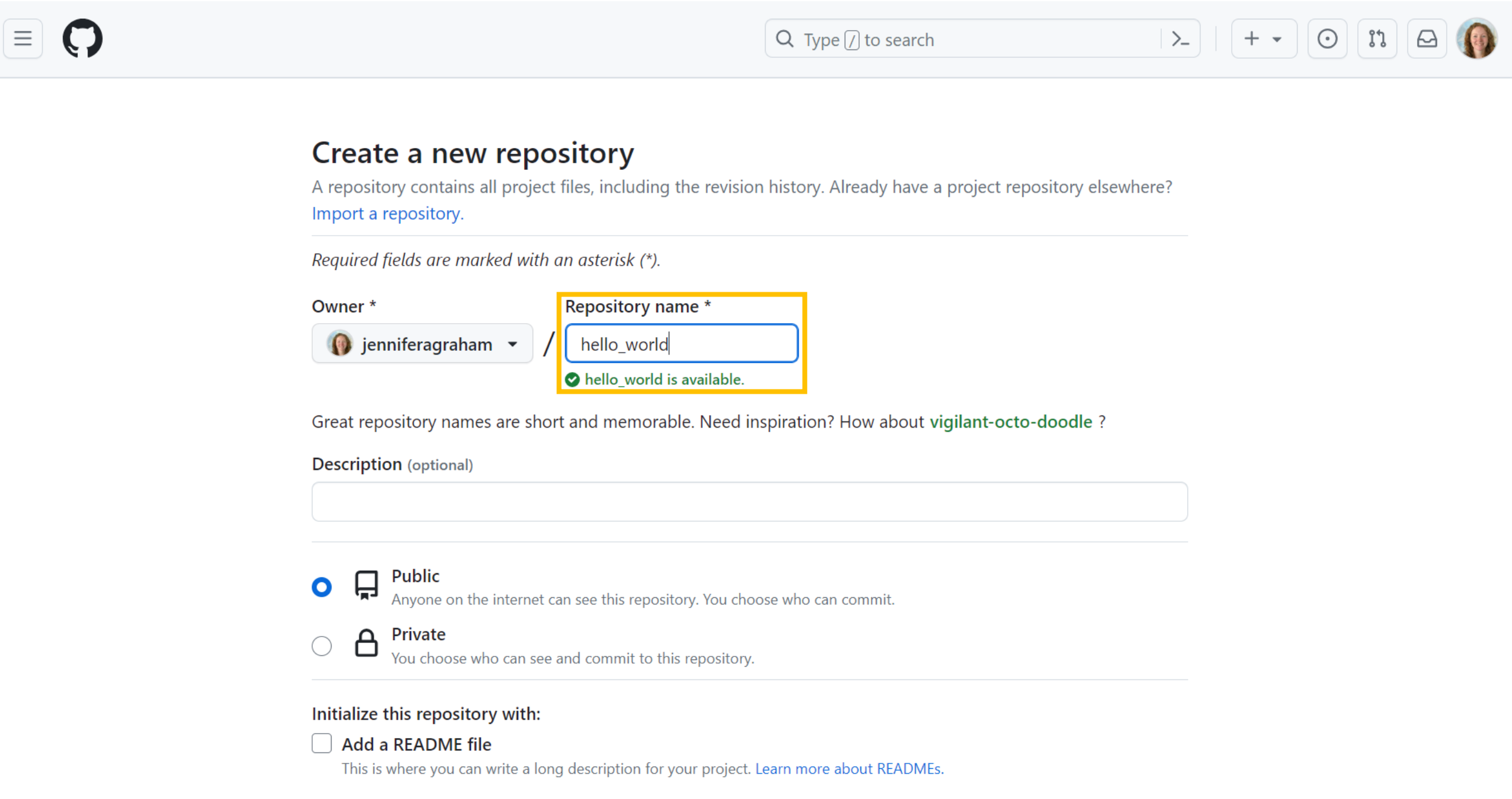
epository. You choose who can commit.

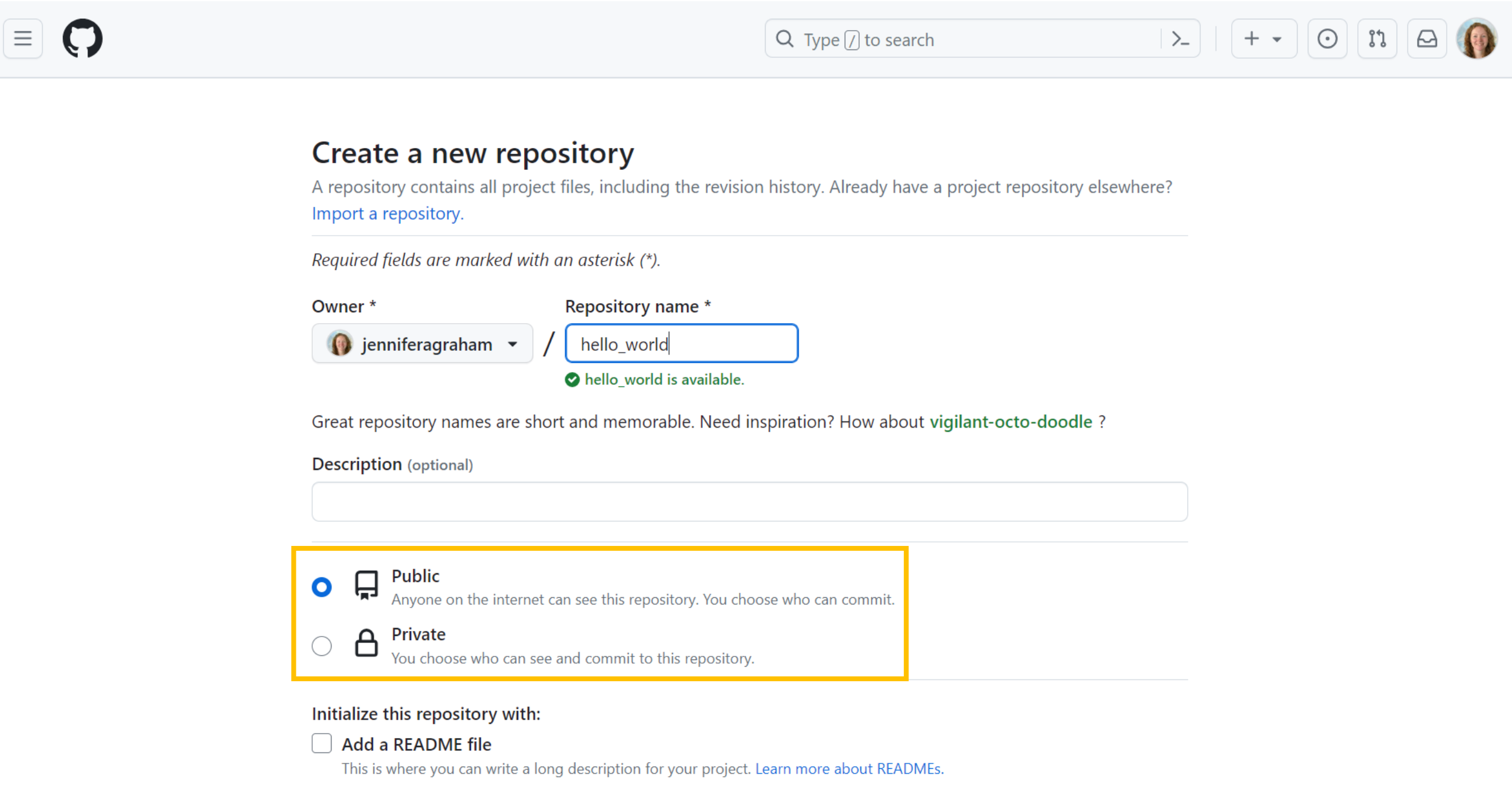
Private
You choose who can see and commit to this repository.

Initialize this repository with:

Add a README file
This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore


Type / to search
Create a new repository
A repository contains all project files, including the revision history. Already have a project repository elsewhere?
Import a repository.
Required fields are marked with an asterisk (*).
Owner *
jenniferagraham
/
Repository name *
hello_world
hello_world is available.
Great repository names are short and memorable. Need inspiration? How about vigilant-octo-doodle ?
Description (optional)
Public
Anyone on the internet can see this repository. You choose who can commit.
Private
You choose who can see and commit to this repository.
Initialize this repository with:
Add a README file
This is where you can write a long description for your project. Learn more about READMEs.


Type / to search
Create a new repository
A repository contains all project files, including the revision history. Already have a project repository elsewhere?
Import a repository.
Required fields are marked with an asterisk (*).
Owner *
jenniferagraham
/
Repository name *
hello_world
hello_world is available.
Great repository names are short and memorable. Need inspiration? How about vigilant-octo-doodle ?
Description (optional)
Public
Anyone on the internet can see this repository. You choose who can commit.
Private
You choose who can see and commit to this repository.
Initialize this repository with:
Add a README file
This is where you can write a long description for your project. Learn more about READMEs.

Description (optional)

Public
Anyone on the internet can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

DO NOT TICK OR ADD anything in this section

Initialize this repository with:

Add a README file
This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. Learn more about ignoring files.

Choose a license

License: None

A license tells others what they can and can't do with your code. Learn more about licenses.

You are creating a private repository in your personal account.

Create repository

Description (optional)

Public
Anyone on the internet can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

Initialize this repository with:

Add a README file
This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. Learn more about ignoring files.

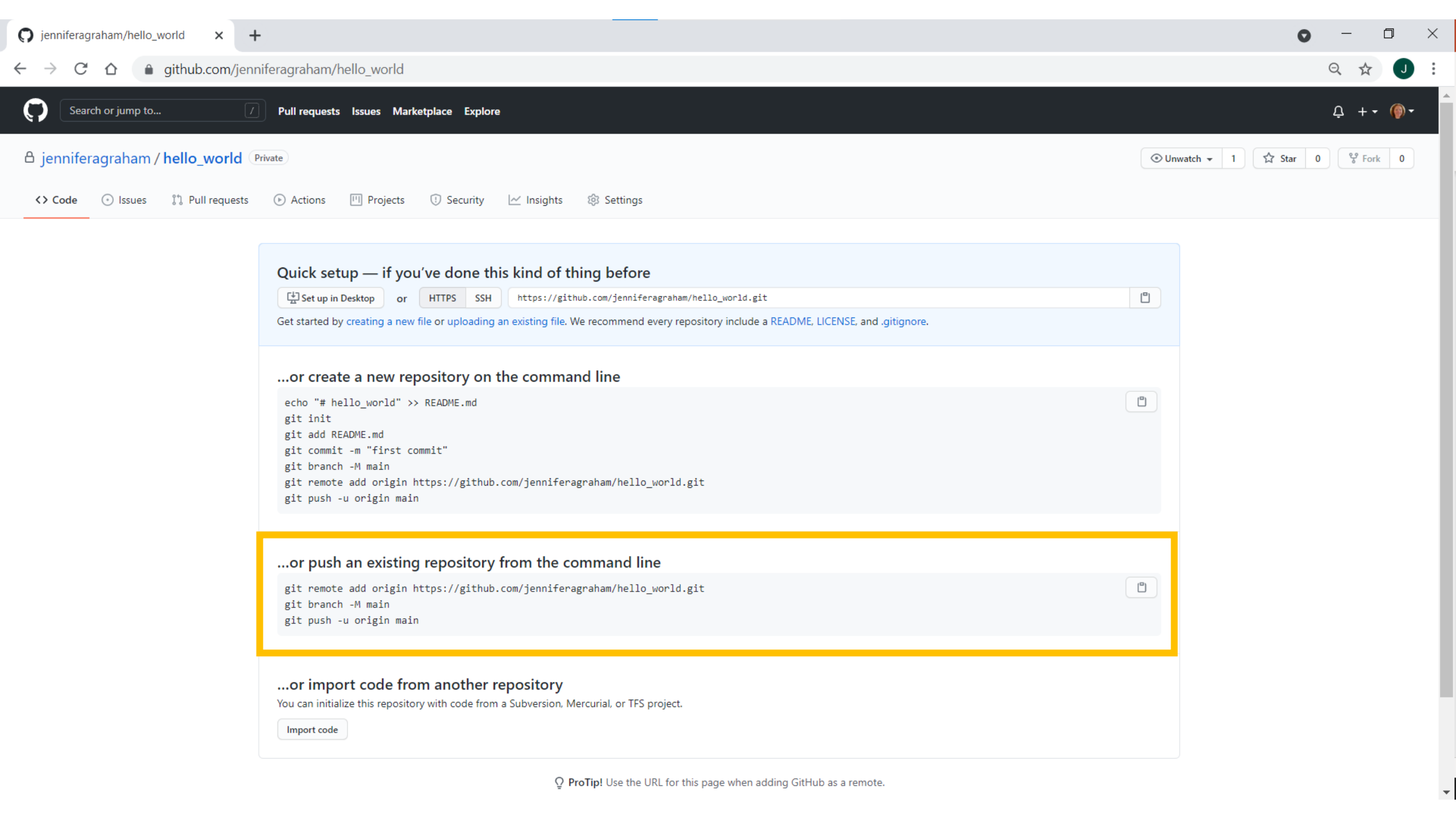
Choose a license

License: None

A license tells others what they can and can't do with your code. Learn more about licenses.

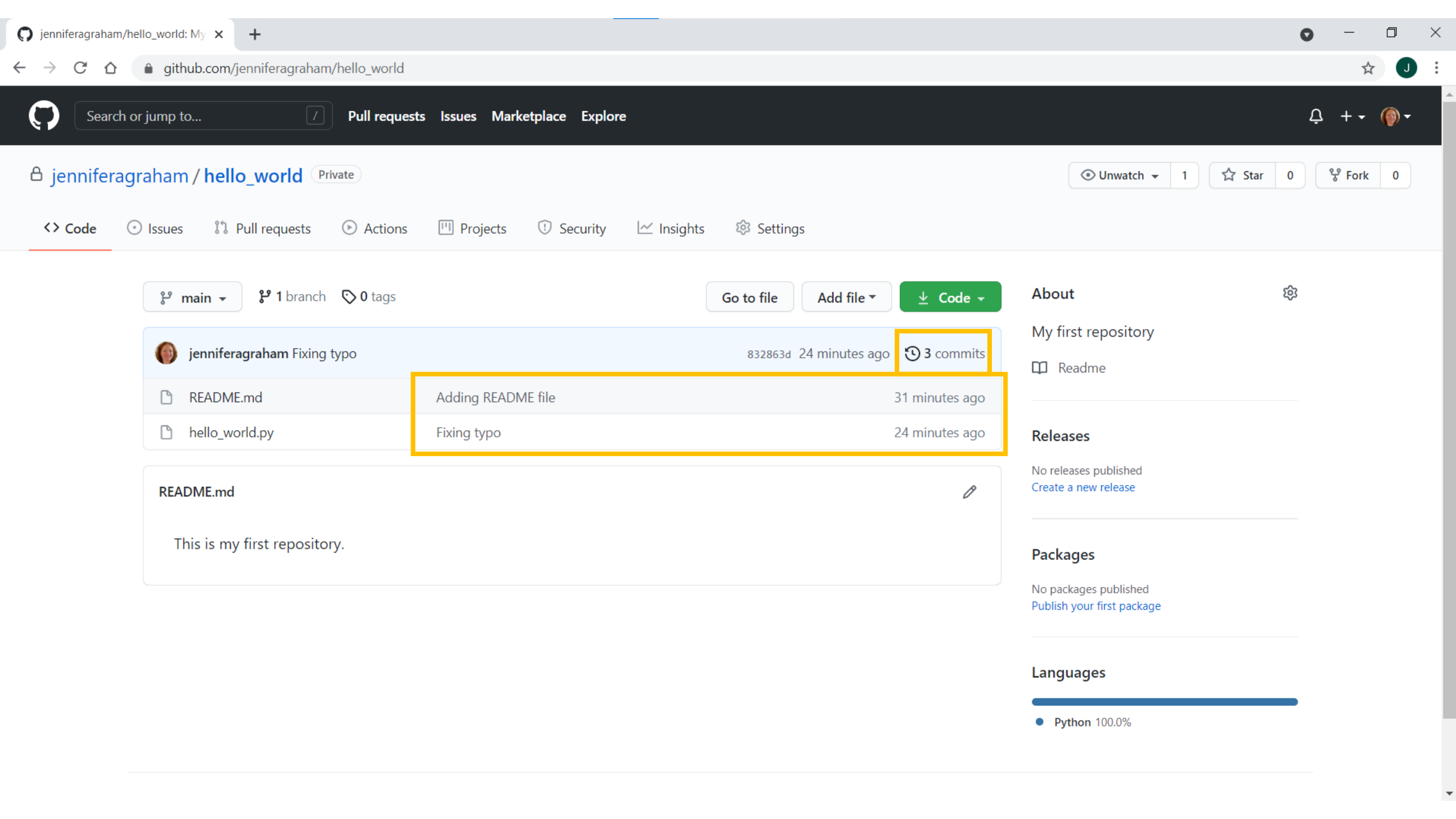
You are creating a private repository in your personal account.

Create repository


jenniferagraham/hello_world
github.com/jenniferagraham/hello_world
Search or jump to...
Pull requests
Issues
Marketplace
Explore
jenniferagraham / hello_world
Private
Unwatch 1
Star 0
Fork 0
Code
Issues
Pull requests
Actions
Projects
Security
Insights
Settings
Quick setup — if you've done this kind of thing before
Set up in Desktop
or
HTTPS
SSH
https://github.com/jenniferagraham/hello_world.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 "# hello_world" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/jenniferagraham/hello_world.git
git push -u origin main
...or push an existing repository from the command line
git remote add origin https://github.com/jenniferagraham/hello_world.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
ProTip! Use the URL for this page when adding GitHub as a remote.

```
JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git push
fatal: No configured push destination.
Either specify the URL from the command-line or configure a remote repository using

    git remote add <name> <url>

and then push using the remote name

    git push <name>


JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git remote add origin https://github.com/jenniferagraham/hello_world.git
git push -u origin main
JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git branch -M main

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git push -u origin main
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 4 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (9/9), 892 bytes | 297.00 KiB/s, done.
Total 9 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/jenniferagraham/hello_world.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$
```

Use commands shown on GitHub to set up link to remote repository.


jenniferagraham/hello_world: My
github.com/jenniferagraham/hello_world
Search or jump to...
Pull requests
Issues
Marketplace
Explore
jenniferagraham / hello_world
Private
Unwatch 1
Star 0
Fork 0
Code
Issues
Pull requests
Actions
Projects
Security
Insights
Settings
main
1 branch
0 tags
Go to file
Add file
Code
jenniferagraham Fixing typo
832863d 24 minutes ago
3 commits
README.md
Adding README file
31 minutes ago
hello_world.py
Fixing typo
24 minutes ago
README.md
This is my first repository.
About
My first repository
Readme
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package
Languages
Python 100.0%

MINGW64:/c/Users/JG10/OneDrive/GitHub/Repos/hello_world

```
JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

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

no changes added to commit (use "git add" and/or "git commit -a")

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git diff README.md
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory
diff --git a/README.md b/README.md
index eb6d976..0d08e3c 100644
--- a/README.md
+++ b/README.md
@@ -1 +1 @@
-This is my first repository.
+This is Jenny's first repository.

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$
```

```
--- a/README.md
+++ b/README.md
@@ -1 +1 @@
-This is my first repository.
+This is Jenny's first repository.

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git add README.md
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git commit
hint: Waiting for your editor to close the file... unix2dos: converting file C:/Users/JG10/OneDrive - CEFAS/
Hub/Repos/hello_world/.git/COMMIT_EDITMSG to DOS format...
dos2unix: converting file C:/Users/JG10/OneDrive - CEFAS/GitHub/Repos/hello_world/.git/COMMIT_EDITMSG to Uni
ormat...
[main 5fa034d] Specify Jenny's repository
 1 file changed, 1 insertion(+), 1 deletion(-)

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 333 bytes | 333.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/jenniferagraham/hello_world.git
   832863d..5fa034d  main -> main

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$
```

jenniferagraham/hello_world: My

github.com/jenniferagraham/hello_world

Search or jump to... / Pull requests Issues Marketplace Explore

jenniferagraham / hello_world Private

Unwatch 1 Star 0 Fork 0

Code Issues Pull requests Actions Projects Security Insights Settings

main 1 branch 0 tags Go to file Add file Code

jenniferagraham Specify Jenny's repository 5fa034d 2 minutes ago 4 commits

README.md	Specify Jenny's repository	2 minutes ago
hello_world.py	Fixing typo	27 minutes ago

README.md

This is Jenny's first repository.

About

My first repository

Readme

Releases

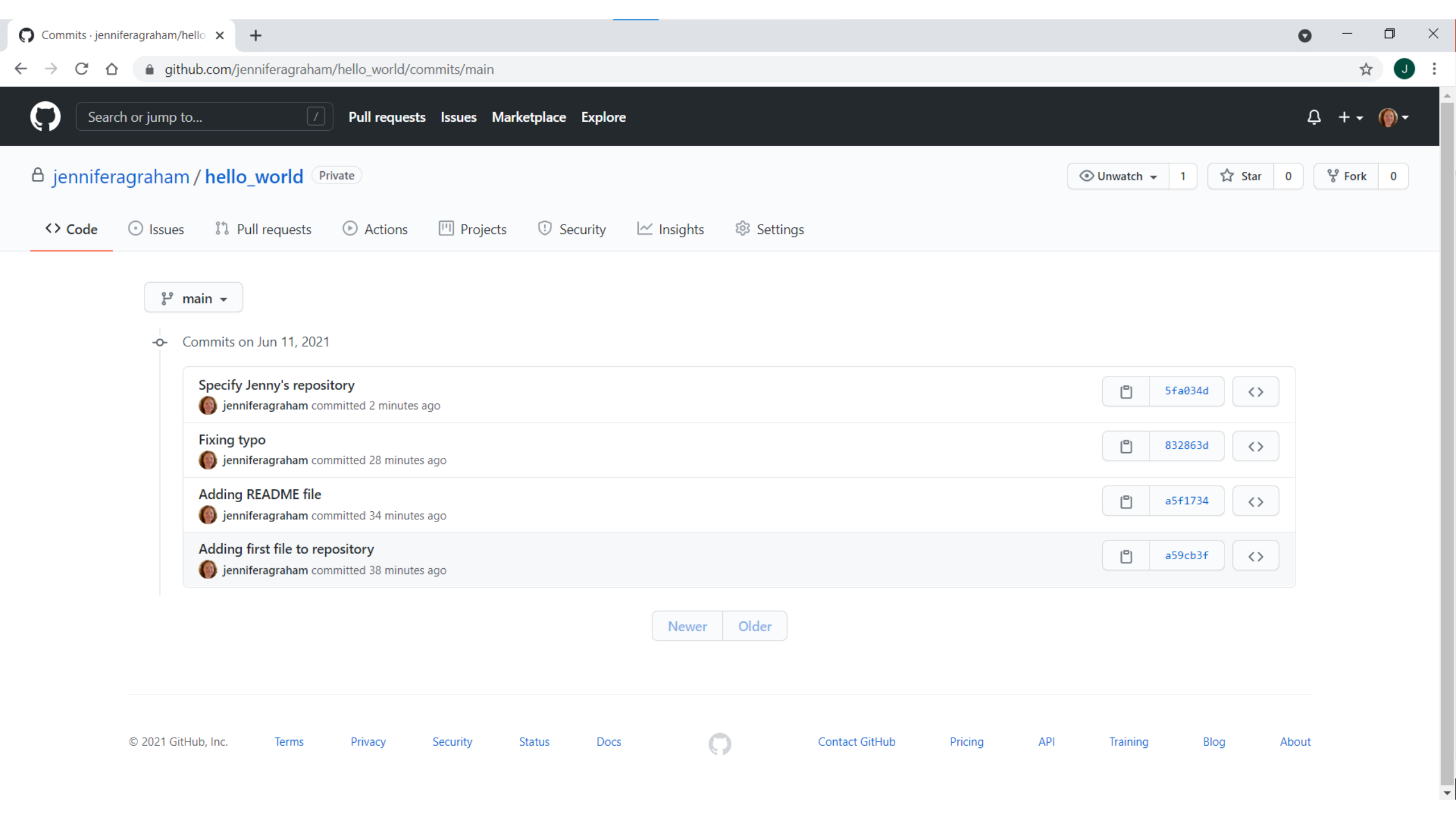
No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

Python 100.0%


Commits · jenniferagraham/hello
github.com/jenniferagraham/hello_world/commits/main
Search or jump to...
Pull requests
Issues
Marketplace
Explore
jenniferagraham / hello_world
Private
Unwatch 1
Star 0
Fork 0
Code
Issues
Pull requests
Actions
Projects
Security
Insights
Settings
main
Commits on Jun 11, 2021
Specify Jenny's repository
jenniferagraham committed 2 minutes ago
5fa034d
Fixing typo
jenniferagraham committed 28 minutes ago
832863d
Adding README file
jenniferagraham committed 34 minutes ago
a5f1734
Adding first file to repository
jenniferagraham committed 38 minutes ago
a59cb3f
Newer
Older
© 2021 GitHub, Inc.
Terms
Privacy
Security
Status
Docs
Contact GitHub
Pricing
API
Training
Blog
About

```
git init                :: turn current directory into git repository.
git add <file>          :: stage the file for commit i.e., track changes.
git commit              :: confirm changes with message/explanation.
git push                :: publish/back-up changes on GitHub (remote server).


git status              :: show which files have been tracked or modified.
git diff <file>         :: show difference between current file and last commit.
git log <file>          :: show commit history [for file]
```

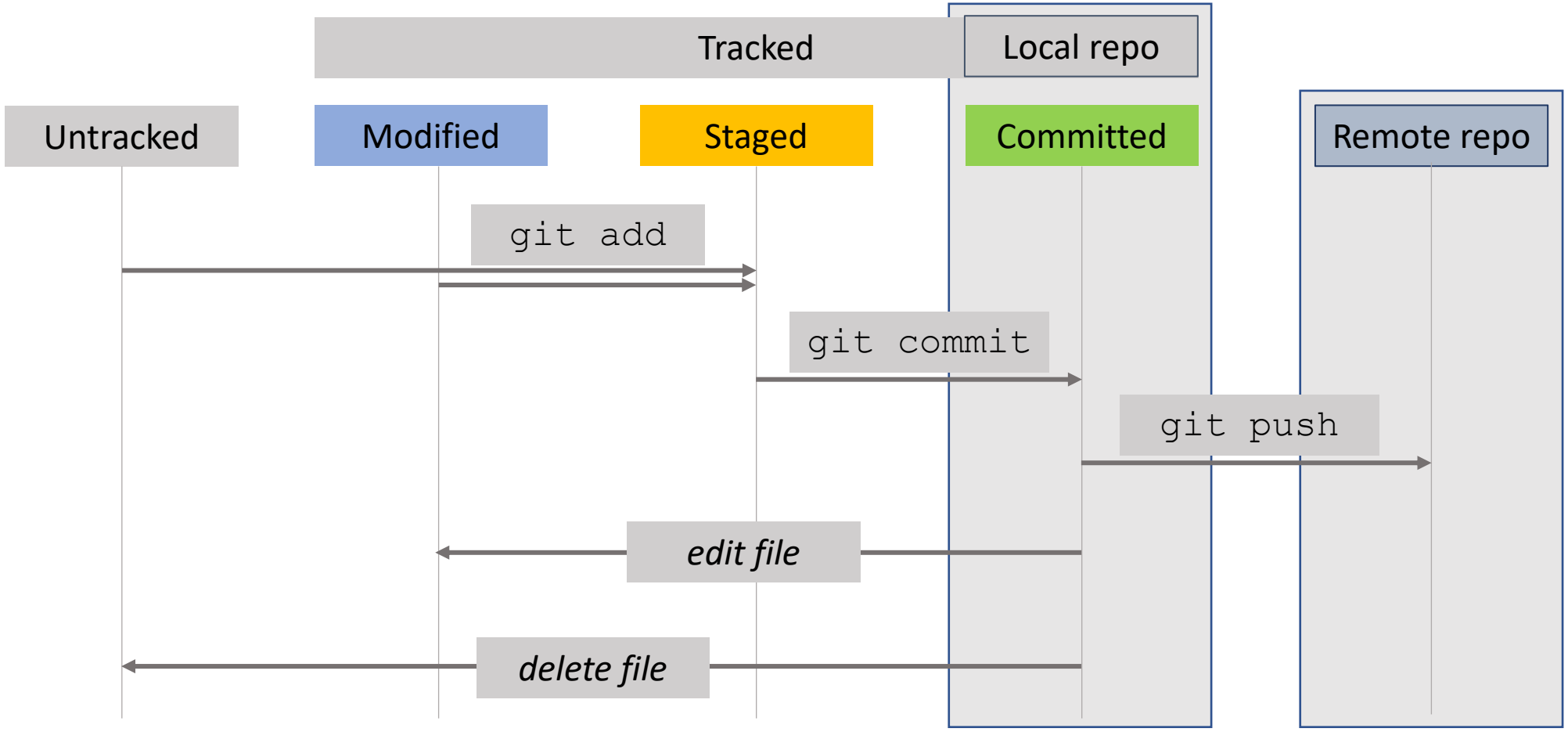
Try this yourself… [15 min]

- Push your repository to GitHub, and view changes.
  - Create an *empty* repository on GitHub (with the name of your local repository).
  - Use the commands that GitHub provides to push your local repository up to the GitHub site.

```
git init          :: turn current directory into git repository.
git add <file>    :: stage the file for commit i.e., track changes.
git commit        :: confirm changes with message/explanation.
git push          :: publish/back-up changes on GitHub (remote server).

git status        :: show which files have been tracked or modified.
git diff <file>:: show difference between current file and last commit.
git log <file>    :: show commit history [for file]
```

Further useful commands…

```
git rm <file>                  :: delete file and remove from repository.
git rm --cached <file>         :: delete file from repository, but keep
                                  local copy.

git mv <file> <new_name>       :: rename file, and keep tracking.

git remote -v                  :: check url for tracked remote repository

git <command> --help           :: open help documentation for any command!
```


Tracked
Local repo
Untracked
Modified
Staged
Committed
Remote repo
git add
git commit
git push
edit file
delete file

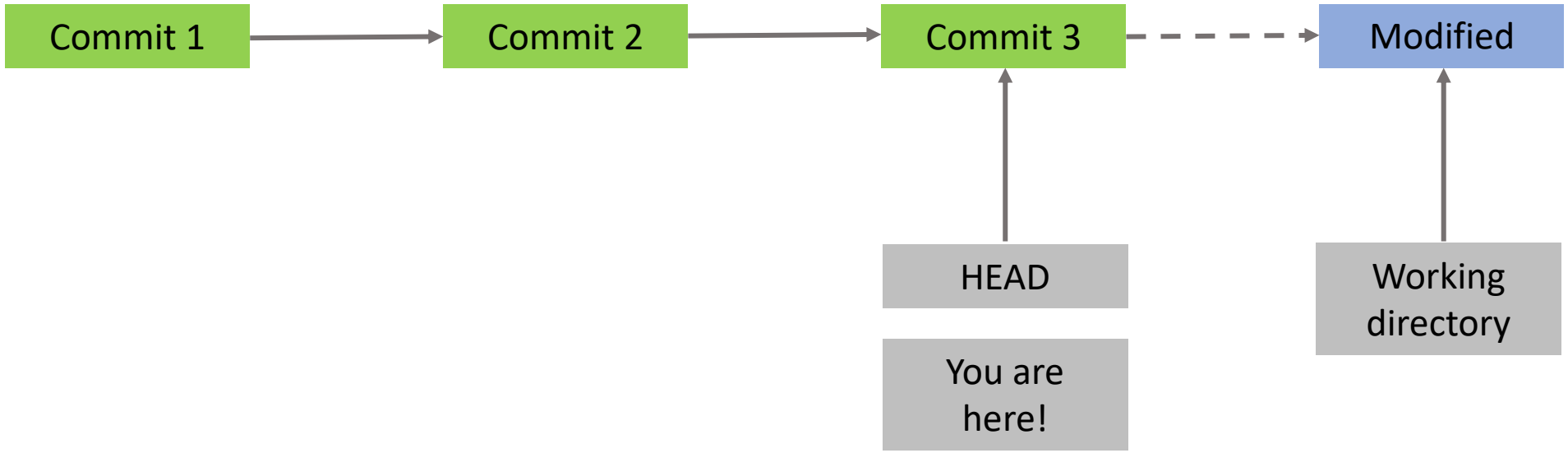
Summary: Terminology

- Repository: Folder containing collection of files to be stored/tracked (any sort of text/code).
- Local repository: Repository being tracked on your local machine (i.e. desktop, hpc).
- Remote repository: a separate copy of your repository stored in a remote location e.g. on GitHub.

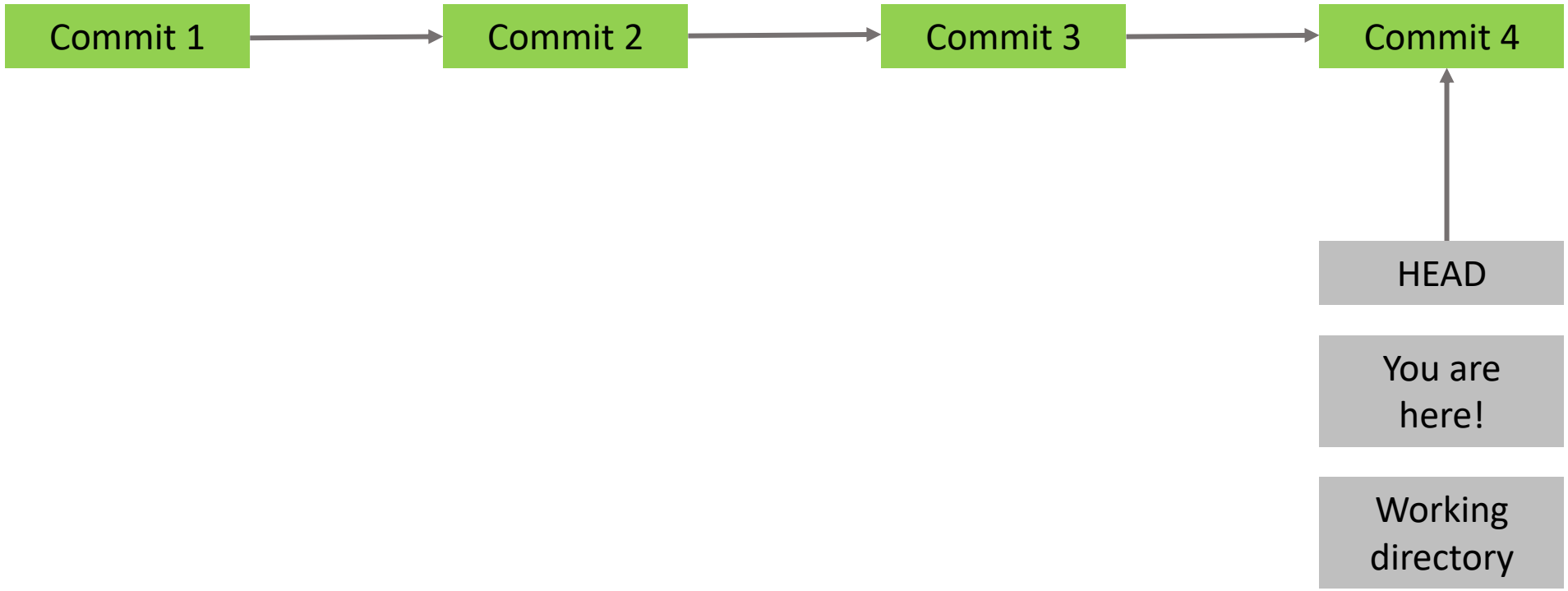
- Add: Tells git what files to track / stage files for commit.
- Commit: Saves a snapshot of files, with explanation for any changes.
- Push: Updates from your local machine synced to the remote location.

- Branch: Repositories can be branched into parallel copies, to safely test large developments to the code.
  - The default/central branch is named the “main” (or “master”) branch.
  - This central branch should always contain a working version of the code.
- Head: Latest commit made, tip of the branch in your local repository (current working revision)

Where is my HEAD?



Where is my HEAD?



More useful tips…

Contents of `.git` folder

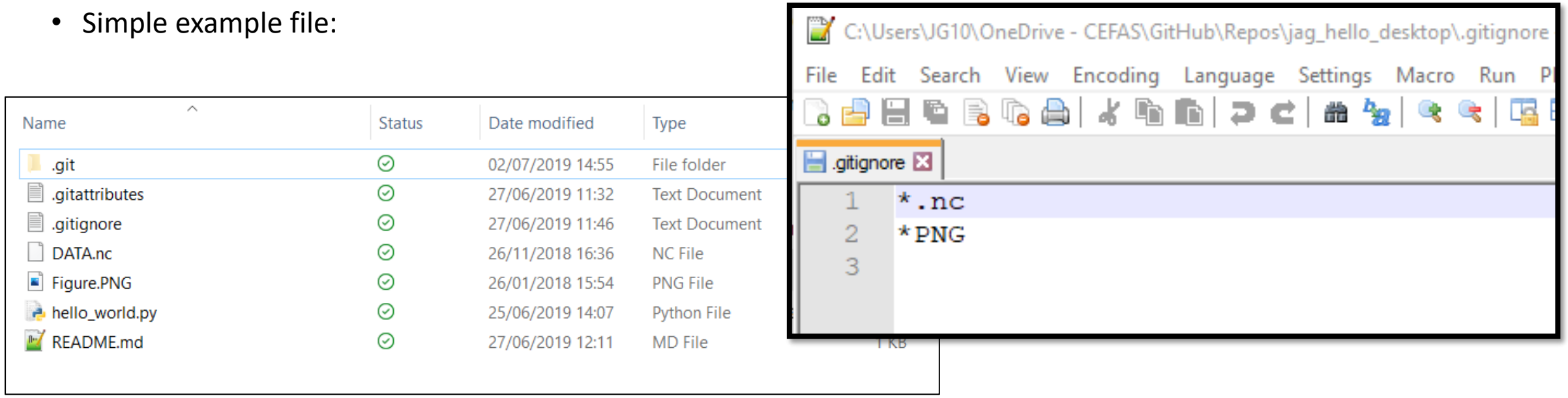
- When a repository is created, this creates a .git folder in the directory.
- .git contains e.g.:

```
JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ ls -a
./  ../  .git/  README.md  hello_world.py

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ ls .git
COMMIT_EDITMSG  HEAD  config  description  hooks/  index  info/  logs/  objects/  refs/
```

- What are these files?
  - They contain everything that git needs to know about the repository!
  - Whenever you run a command, it will use and/or edit files here.

.gitignore

- .gitignore files tell git what to ignore!
- Can ignore any file, directory, or file extensions…
  - e.g. executables, log files, data, figures… (any secrets)
  - Standard examples are provided on GitHub.
- Simple example file:



git config

- Check global username and email settings: `git config --global -l`

```
JG10@G6W1YF2 MINGW64 ~
$ git config --global -l
filter.lfs.required=true
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
user.name=jenniferagraham
user.email=jennifer.graham@cefas.co.uk
```

- Change default editor?

```
JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world
$ git config --get core.editor
vi

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world
$ git config --global core.editor notepad

JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world
$ git config --get core.editor
notepad
```

git config

- Check local repository settings e.g. path to remote repository

```
git config --local -l
```

```
JG10@G6W1YF2 MINGW64 ~/OneDrive/GitHub/Repos/hello_world (main)
$ git config --local -l
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.ignorecase=true
remote.origin.url=https://github.com/jenniferagraham/hello_world.git
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
branch.main.remote=origin
branch.main.merge=refs/heads/main
```

GitHub settings

- Global settings
  - Your account details, notifications, organisation links, etc.
- Repository settings
  - Access, features, visibility, etc.

Your Profile

github.com/settings/profile

Search or jump to... / Pull requests Issues Marketplace Explore

Signed in as **jenniferagraham**

Set status

Your profile
Your repositories
Your codespaces
Your organizations
Your projects
Your stars
Your gists

Upgrade
Feature preview
Help
Settings

Sign out

Jennifer Graham
Your personal account

Go to your personal profile

Account settings

- Profile
- Account
- Appearance
- Account security
- Billing & plans
- Security log
- Security & analysis
- Emails
- Notifications
- Scheduled reminders
- SSH and GPG keys
- Repositories
- Packages
- Organizations
- Saved replies

Public profile

Name

Jennifer Graham

Your name may appear around GitHub where you contribute or are mentioned. You can

Public email

Select a verified email to display

You have set your email address to private. To toggle email privacy, go to email settings and uncheck "Keep my email address private."

Bio

Tell us a little bit about yourself

You can **@mention** other users and organizations to link to them.

URL

Twitter username

Company

Profile picture

Edit

https://github.com/settings/profile

Options

github.com/jenniferagraham/hello_world/settings

Search or jump to... / Pull requests Issues Marketplace Explore

jenniferagraham / hello_world Private

Unwatch 1 Star 0 Fork 0

Code Issues Pull requests Actions Projects Security Insights Settings

- Options
- Manage access
- Security & analysis
- Branches
- Webhooks
- Notifications
- Integrations
- Deploy keys
- Actions
- Secrets
- Pages

Settings

Repository name

hello_world Rename

☐ **Template repository**

Template repositories let users generate new repositories with the same directory structure and files. Learn more.

Social preview

⚠ You can upload a social image, but it will not be visible publicly while **jenniferagraham/hello_world** is private.

Upload an image to customize your repository's social media preview.

Images should be at least 640×320px (1280×640px for best display).

Download template

Adding documentation

- README
  - Automatically the front page of your repository.
  - Markdown formatting can be used
- Wiki
  - Broader information on use of repository.
  - Only available within public repositories or organisations.
  - e.g. https://github.com/CefasRepRes/Git_Training/wiki/
- Issues
  - Flagging development stages or known problems.
- Projects
  - Organise work/issues on repositories?

- Some repository tracking info only available within organisations or paid accounts
  - More info here: https://github.com/pricing

Issues

- Useful to flag development stages or known problems/bugs.
- Can be referred to elsewhere e.g. in commit messages.