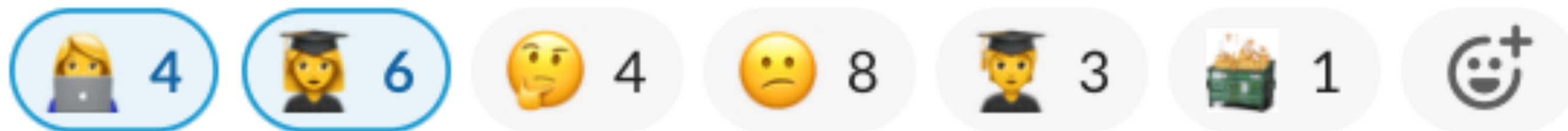


Intro to version control with **git**

~~Pop quiz!~~ Poll! Please rate your familiarity with git using emoji such as 🧑💻 (I use it regularly), 😎 (easy peasy), 🤔 (I've heard of git before), 😞 (I have questions), 😊 (HELP), 🎓 (I'd like to learn more). Multiple answers (and other emojis) encouraged! (edited)



Henrike Fleischhack
June 2nd 2023

Content

- 1. Intro to git**
- 2. Git(hub) for users**
- 3. [Git(hub) for developers]**

Content

- 1. Intro to git**
2. Git(hub) for users
3. [Git(hub) for developers]

Why version control?

Why version control?

- Text documents: Software development, webpages, LaTeX source, ...
- Option to restore old versions, track release versions
- Track changes and history
- "Parallel" feature development
- Collaborative editing
- Backups

Git vs GitHub

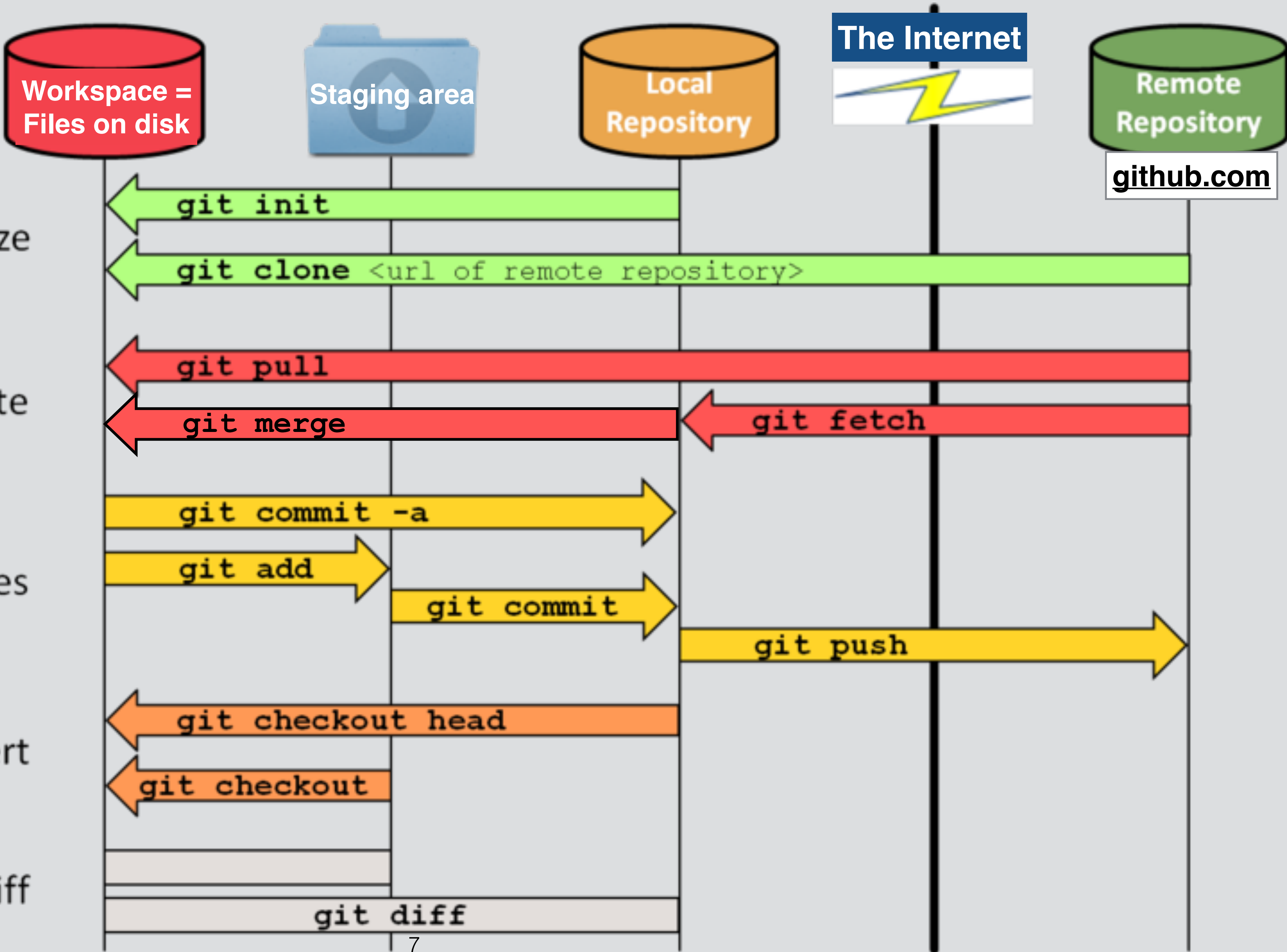
Git

- Lives on your computer (command line or GUI).
- Handles branching, adding and committing changes, merging, conflicts, ...
- Can be used offline or online with remote host
- Git keeps the full history of the repository (commits & branches) locally!

Github

- One of several commercial online hosts for git repositories
- Provides off-site backup, online access, web interface.
- Handles user permissions, "forks"
- Optional add-ons: Issue tracking, code review, automated testing etc.

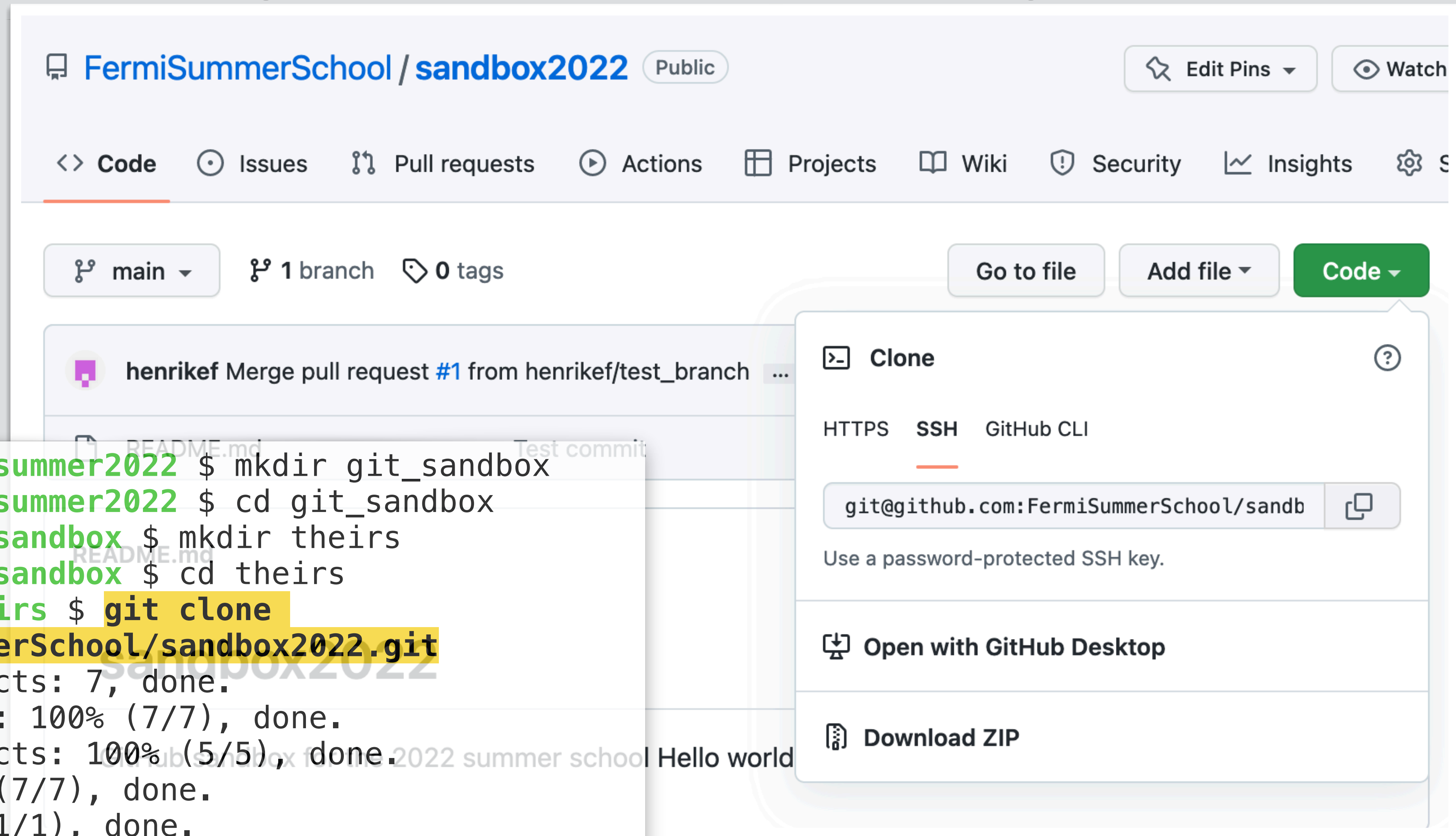
Git principles



Content

1. Intro to git
- 2. Git(hub) for users**
3. [Git(hub) for developers]

Initializing a local repo (cloning)



```
(base) ✓ fermi_software/summer2022 $ mkdir git_sandbox
(base) ✓ fermi_software/summer2022 $ cd git_sandbox
(base) ✓ summer2022/git_sandbox $ mkdir theirs
(base) ✓ summer2022/git_sandbox $ cd theirs
(base) ✓ git_sandbox/theirs $ git clone
git@github.com:FermiSummerSchool/sandbox2022.git
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (5/5), done.
Receiving objects: 100% (7/7), done.
Resolving deltas: 100% (1/1), done.
remote: Total 7 (delta 1), reused 3 (delta 0), pack-reused
0
(base) ✓ git_sandbox/mine $ cd sandbox2022
```

Adding (or changing) a file

```
(base) ✓ theirs/sandbox2022 $ echo "This is a new file" > new_file.md
```

```
(base) ✓ theirs/sandbox2022 $ git status
```

```
On branch main
```

```
Your branch is up to date with 'origin/main'.
```

```
Untracked files:
```

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

```
new_file.md
```

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

```
(base) ✓ theirs/sandbox2022 $ git add new_file.md
```

```
(base) ✓ theirs/sandbox2022 $ git commit -m "Adding new file directly to main branch"
```

```
[main 962174e] Adding new file directly to main branch
```

```
1 file changed, 1 insertion(+)
```

```
create mode 100644 new_file.md
```

```
(base) ✓ theirs/sandbox2022 $ git push
```

```
Counting objects: 100% (4/4), done.
```

```
Delta compression using up to 8 threads
```

```
Compressing objects: 100% (2/2), done.
```

```
Writing objects: 100% (3/3), 325 bytes | 325.00 KiB/s, done.
```

```
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
```

```
To github.com:FermiSummerSchool/sandbox2022.git
```

```
39b4531..962174e main -> main
```

Each commit is
assigned a unique hash

Please use meaningful
commit messages!

Keeping up with remote changes

```
(base) ✓ summer-school-2023/sandbox2022 $ git pull
hint: Pulling without specifying how to reconcile divergent branches is
hint: discouraged. You can squelch this message by running one of the following
hint: commands sometime before your next pull:
hint:
hint:   git config pull.rebase false  # merge (the default strategy)
hint:   git config pull.rebase true   # rebase
hint:   git config pull.ff only       # fast-forward only
hint:
hint: You can replace "git config" with "git config --global" to set a default
hint: preference for all repositories. You can also pass --rebase, --no-rebase,
hint: or --ff-only on the command line to override the configured default per
hint: invocation.
Already up to date.
```

Helpful hint - read
and pick an option

Git history

```
(base) ✓ theirs/sandbox2022 $ git log
commit 962174e031292b5fa64872ecafea540f0cdf8023 (HEAD -> main, origin/main, origin/HEAD)
Author: Henrike Fleischhack <henrike.fleischhack@googlemail.com>
Date:   Fri May 27 16:42:45 2022 -0400

    Adding new file directly to main branch

commit 39b453170dd5883f5bab1ee804bfec8a18f174cc
Merge: d4f71a4 278ea10
Author: Henrike F <henrikef@users.noreply.github.com>
Date:   Tue May 24 18:34:40 2022 -0400

    Merge pull request #1 from henrikef/test_branch

    Test commit
...
```

main ▾

Commits on May 27, 2022

Adding new file directly to main branch

henrikef committed 12 minutes ago

962174e <>

Commits on May 24, 2022

Merge pull request #1 from henrikef/test_branch ...

henrikef committed 3 days ago

Verified 39b4531 <>

Test commit

henrikef committed 3 days ago

278ea10 <>

Initial commit

henrikef committed 3 days ago

Verified d4f71a4 <>

Each commit is assigned a unique hash

Git history (details)

Adding new file directly to main branch [Browse files](#)

🔗 main

henrikef committed 12 minutes ago 1 parent [39b4531](#) commit [962174e031292b5fa64872ecafea540f0cdf8023](#)

Showing 1 changed file with 1 addition and 0 deletions. Split Unified

1 new_file.md <> 📄 ...

```
... @@ -0,0 +1 @@  
1 + This is a new file
```

0 comments on commit [962174e](#) 🔒 Lock conversation

H B I \equiv <> 🔗 \equiv @ 🗨️ ↩

Write Preview

Leave a comment

Attach files by dragging & dropping, selecting or pasting them. M+

[Comment on this commit](#)

```
(base) ✓ theirs/sandbox2022 $ git diff 39b453170dd5883f5bab1ee804bfec8a18f174cc 962174e031292b5fa64872ecafea540f0cdf8023  
diff --git a/new_file.md b/new_file.md  
new file mode 100644  
index 0000000..6dfa057  
--- /dev/null  
+++ b/new_file.md  
@@ -0,0 +1 @@  
+This is a new file
```

Installing git

- See <https://www.atlassian.com/git/tutorials/install-git>
- MacOS:
 - Included with Xcode
 - Homebrew or standalone installer
- Linux & friends:
 - Use your favorite package installer: `sudo apt-get install git`
- Windows e.g.:
 - <https://gitforwindows.org/>
 - <https://git-scm.com/download/win>

Getting set up on your computer

- Git configuration (system-wide, per user, per repository): **git config**
- See <https://git-scm.com/book/en/v2/Getting-Started-First-Time-Git-Setup>

- Add your identity:

```
git config --global user.name "Jane Doe"
```

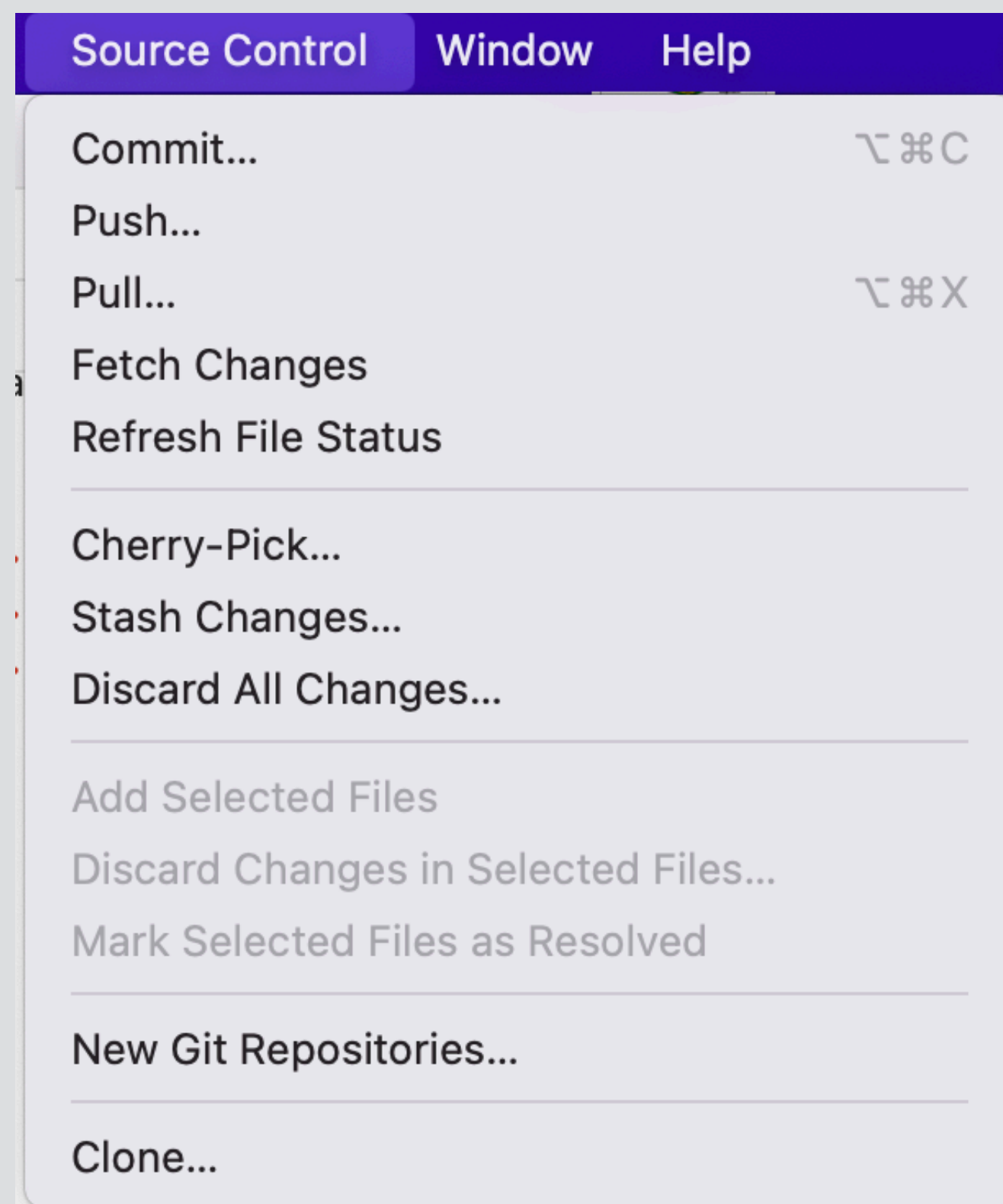
```
git config --global user.email janedoe@example.com
```

- Set editor for commit messages:

```
git config --global core.editor emacs (or whatever)
```

Rather use a GUI?

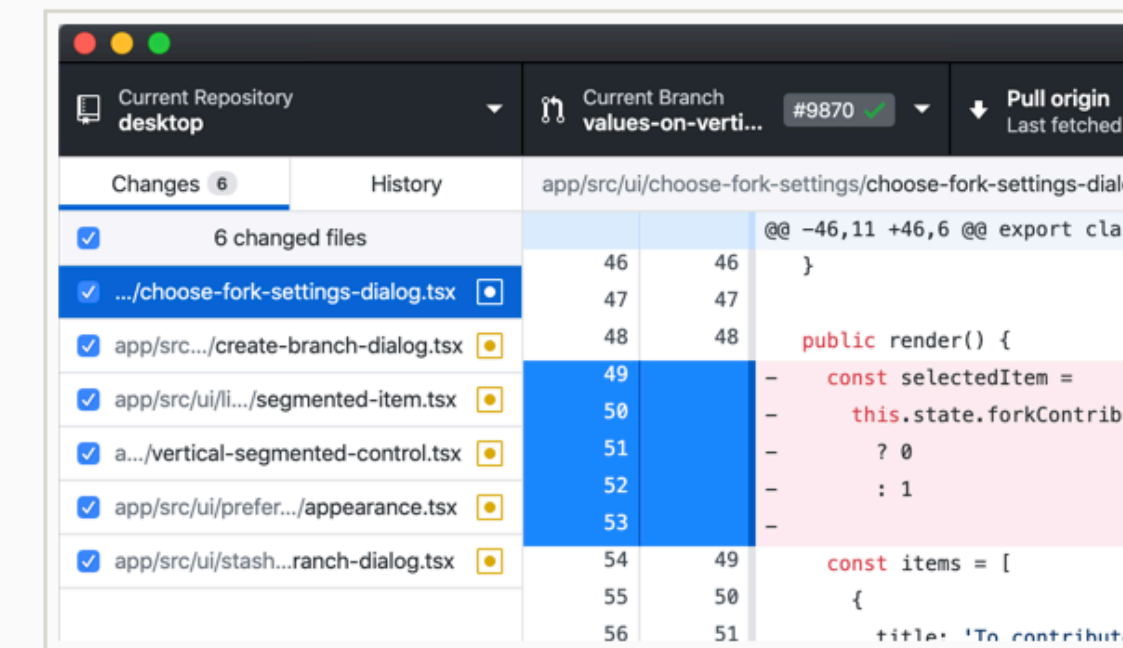
- See <https://git-scm.com/downloads/guis>.
- Many IDEs also offer git integration.



GUI Clients

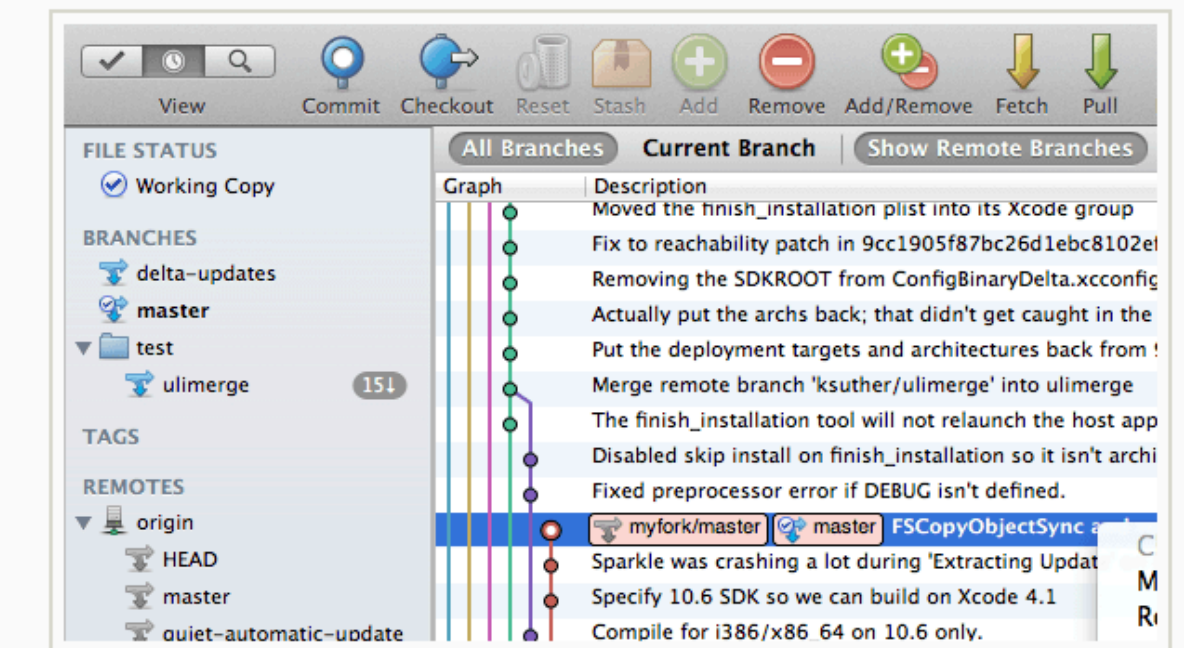
Git comes with built-in GUI tools for committing ([git-gui](#)) and browsing ([gitk](#)), but there are several third-party tools for users looking for platform-specific experience.

If you want to add another GUI tool to this list, just [follow the instructions](#).



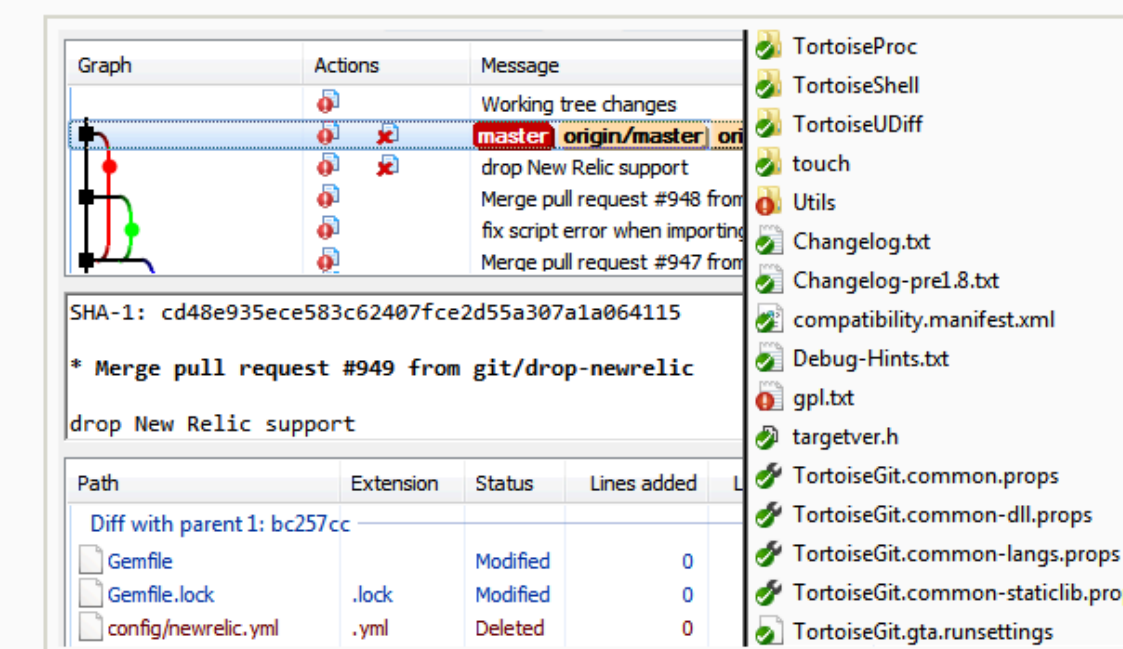
GitHub Desktop

Platforms: Mac, Windows
Price: Free
License: MIT



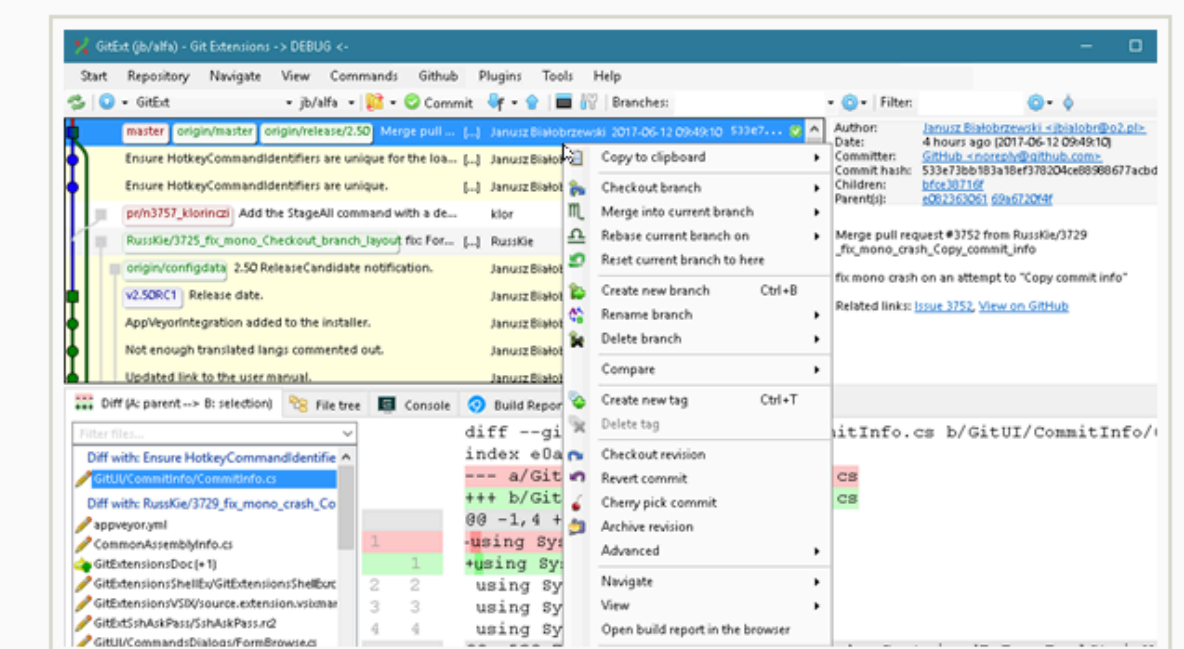
SourceTree

Platforms: Mac, Windows
Price: Free
License: Proprietary



TortoiseGit

Platforms: Windows
Price: Free
License: GNU GPL

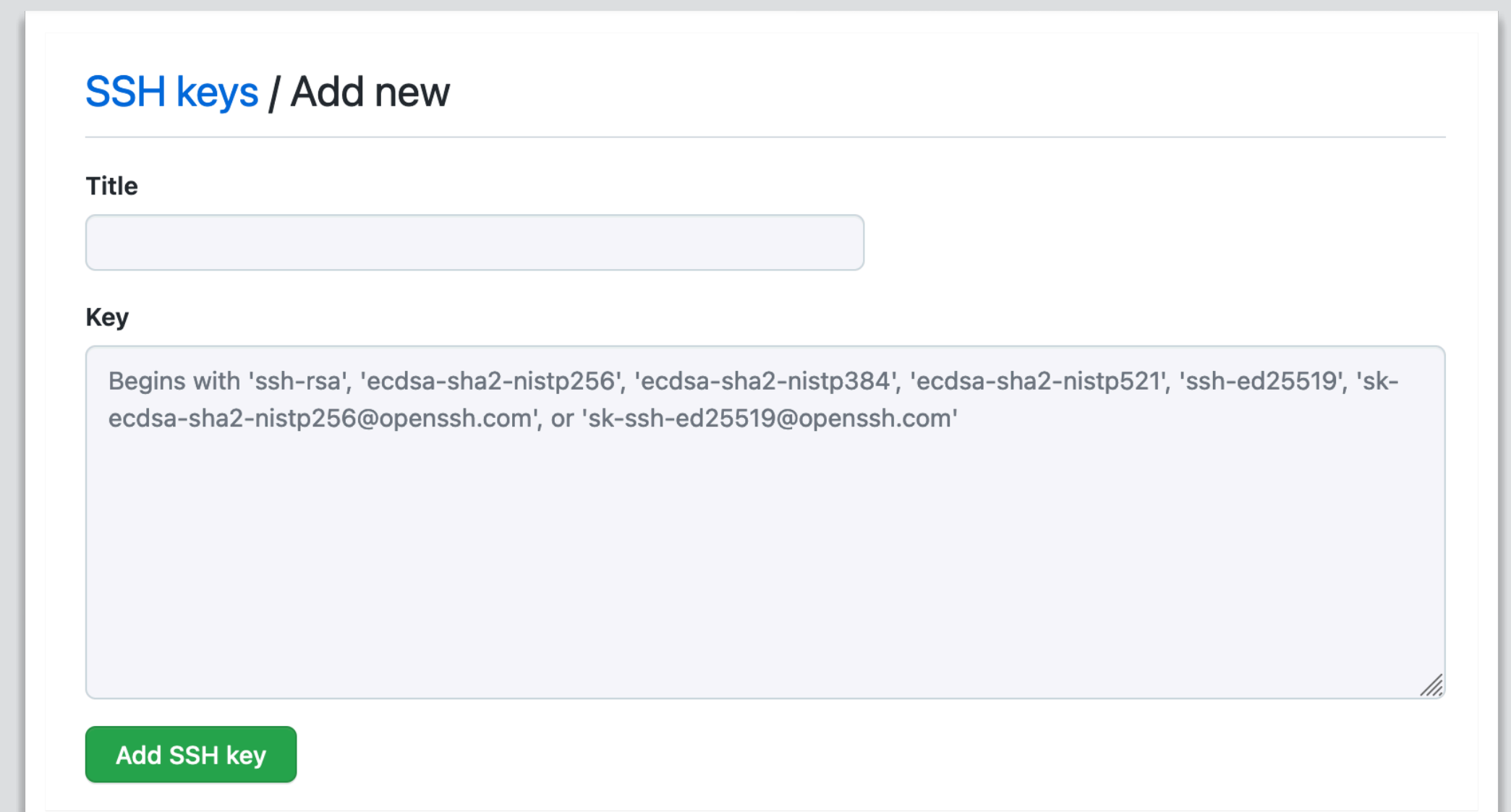
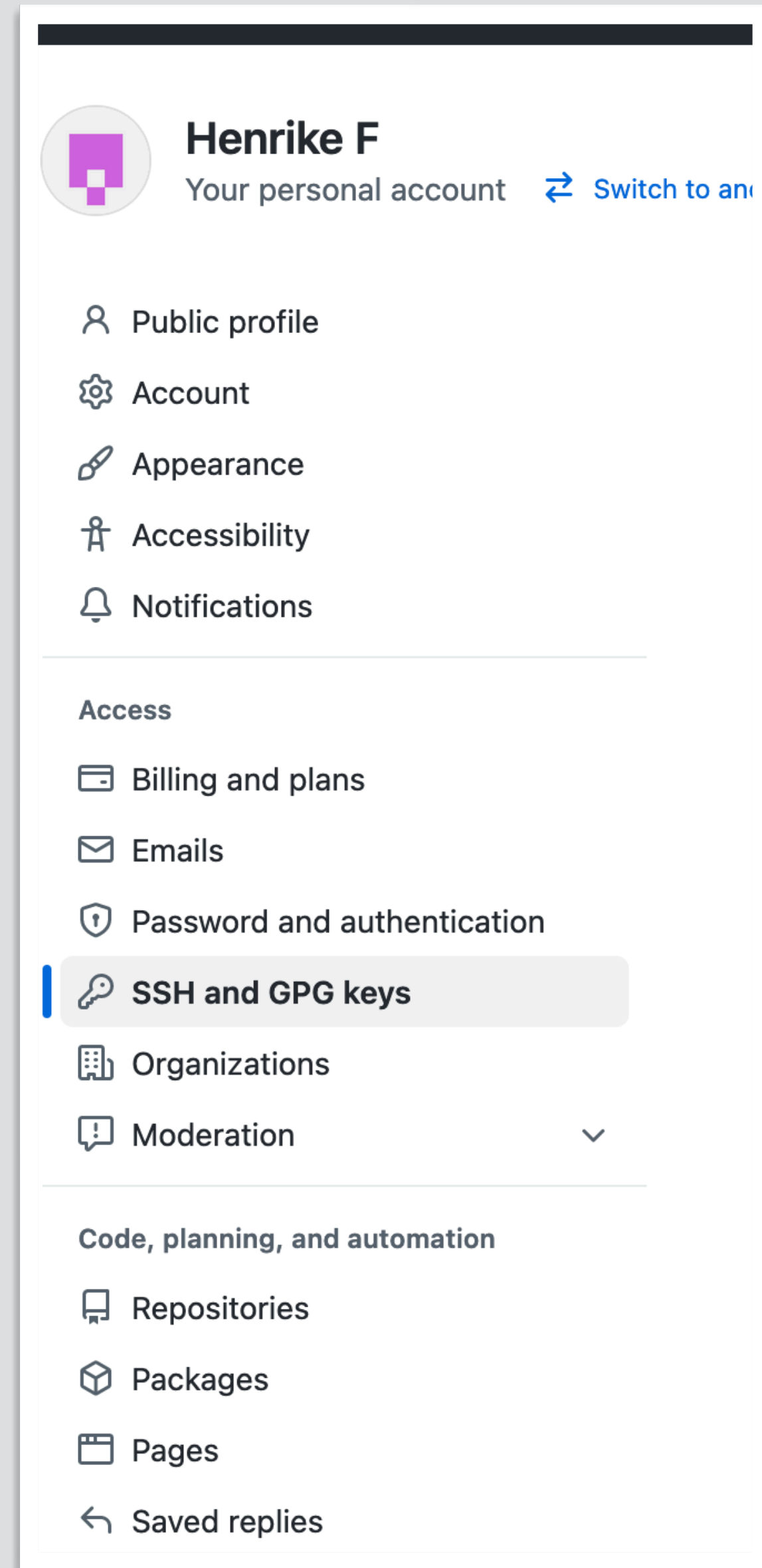
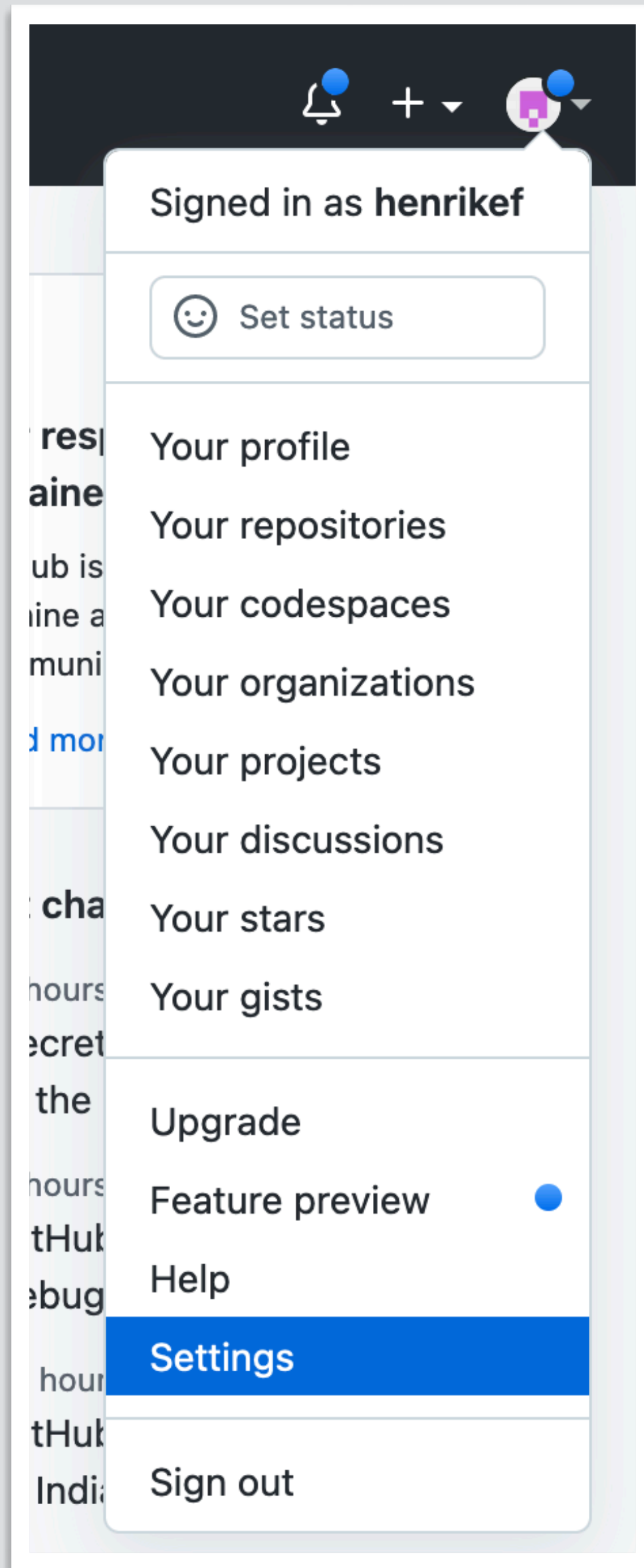


Git Extensions

Platforms: Linux, Mac, Windows
Price: Free
License: GNU GPL

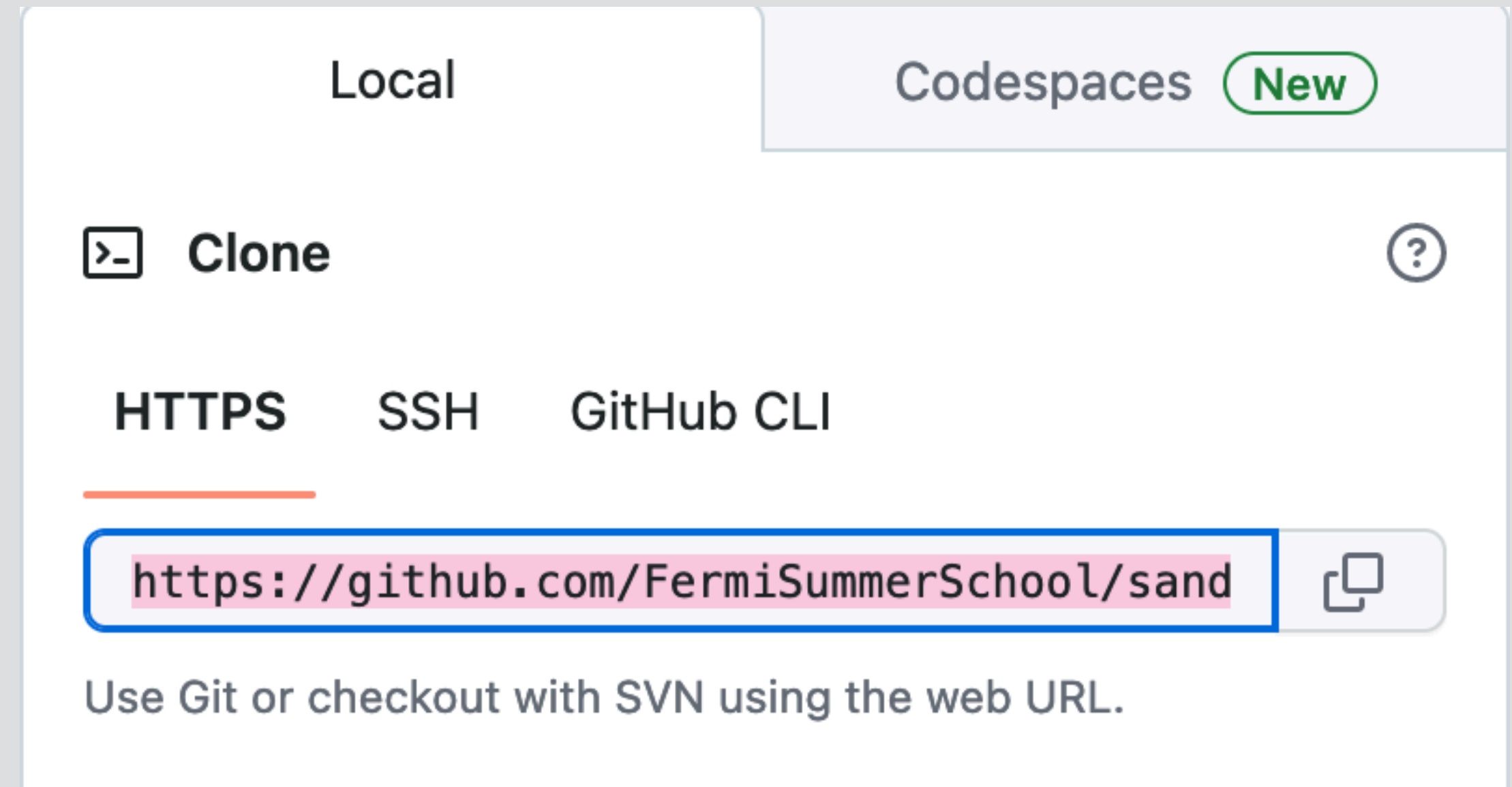
Getting set up on github

- [GitHub.com](https://github.com) -> sign up or sign in
- Make sure to set up an ssh key
- Inside and outside the bottle
- Instructions linked on GitHub
- Never share your private key!



Ssh vs https authentication

- I could not get ssh access to work on the UDel guest wifi.
- Use https for now instead.
- Change to ssh later:



```
(base) ✓ summer-school-2023/sandbox2022 $ git remote -v
origin https://github.com/FermiSummerSchool/sandbox2022.git (fetch)
origin https://github.com/FermiSummerSchool/sandbox2022.git (push)
(base) ✓ summer-school-2023/sandbox2022 $ git remote set-url origin
git@github.com:FermiSummerSchool/sandbox2022.git
(base) ✓ summer-school-2023/sandbox2022 $ git remote -v
origin git@github.com:FermiSummerSchool/sandbox2022.git (fetch)
origin git@github.com:FermiSummerSchool/sandbox2022.git (push)
```

What's next?

Go to <https://github.com/FermiSummerSchool/sandbox2022>

Make your own fork, play around and make a pull request!

Fermi summer school repo (with notebooks) is here:

<https://github.com/FermiSummerSchool/fermi-summer-school>

If you notice an error, please open an issue or send a pull request with the fix!

Questions?

<https://www.atlassian.com/git/tutorials>
(Skip the "Bitbucket Cloud" parts)
<https://docs.github.com/en>

"I know that I know nothing"

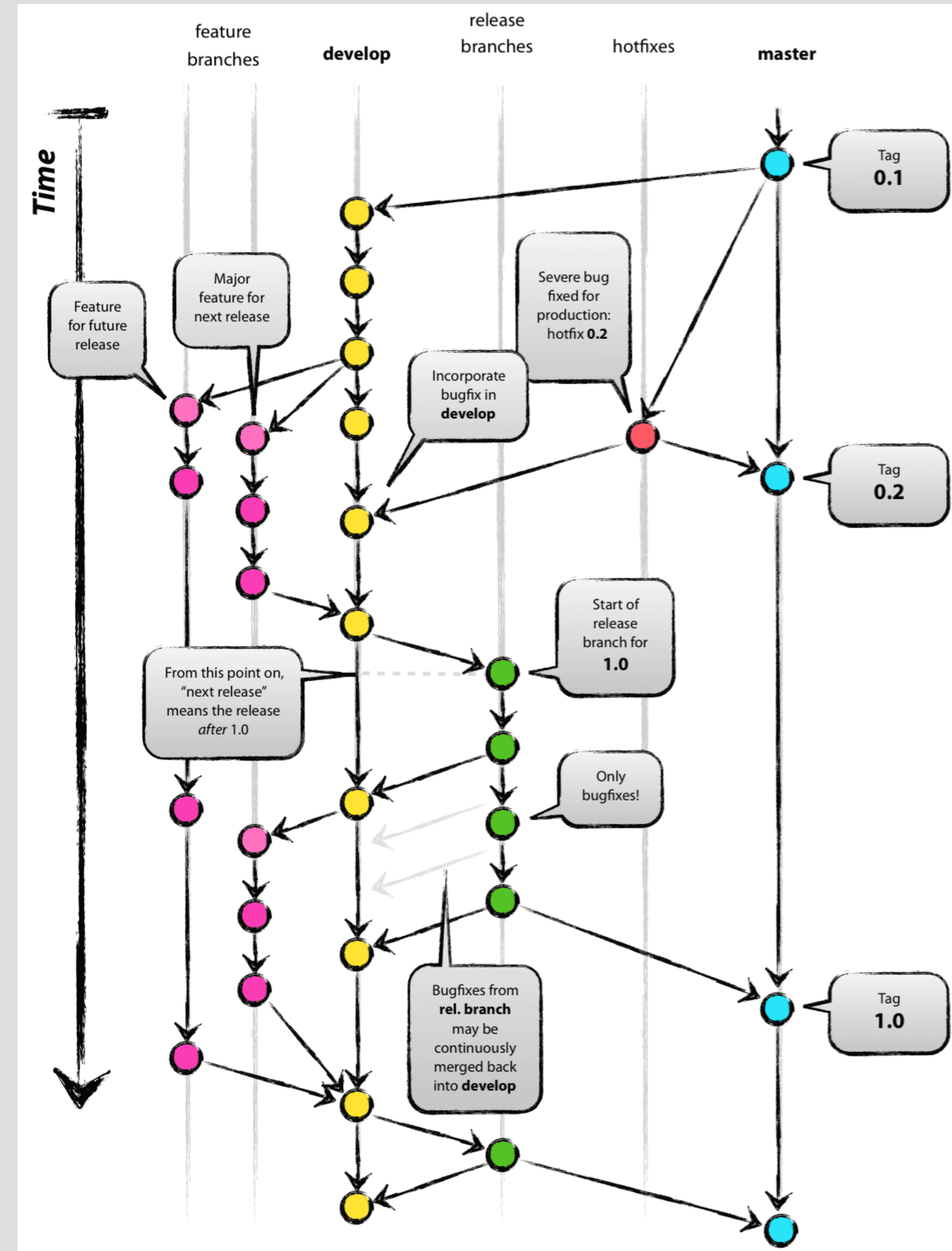
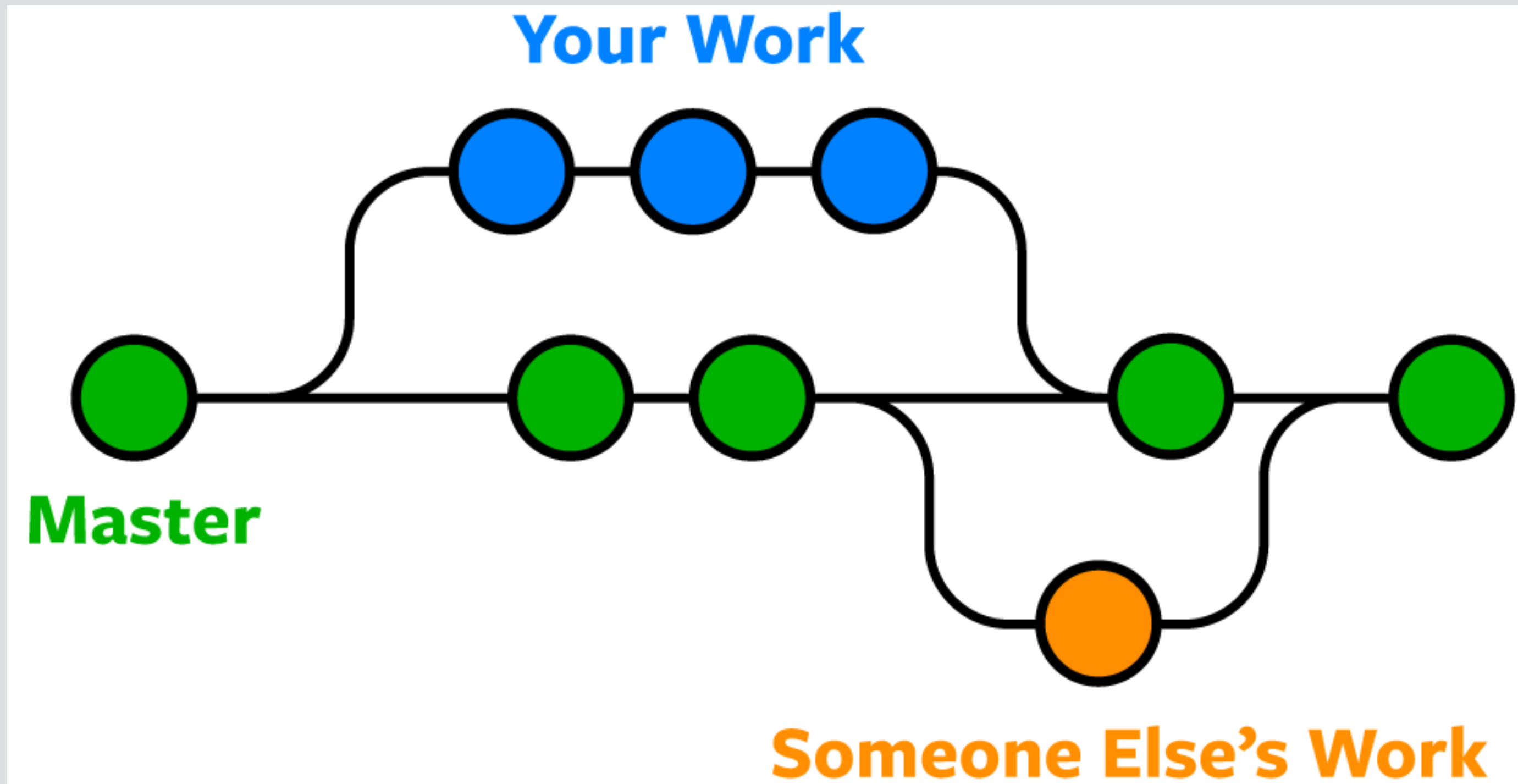
"The unexamined life is not worth living"

Gadfly • Trial of Socrates

Content

1. Intro to git
2. Git(hub) for users
3. **Git(hub) for developers**
 - A. **Branching, forking, and pull requests**
 - B. Undoing changes

Branching and parallel development



Development on a branch

```
Equivalent:  
git branch test_branch  
git checkout test_branch
```

```
(base) ✓ theirs/sandbox2022 $ git checkout -b test_branch  
Switched to a new branch 'test_branch'  
(base) ✓ theirs/sandbox2022 $ echo "Hello world\!" >> README.md  
(base) ✓ theirs/sandbox2022 $ git status  
On branch test_branch  
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
```

```
(base) ✓ theirs/sandbox2022 $ git add README.md  
(base) ✓ theirs/sandbox2022 $ git commit -m "Test commit"  
[test_branch 278ea10] Test commit  
1 file changed, 2 insertions(+)  
(base) ✓ theirs/sandbox2022 $ git push  
fatal: The current branch test_branch has no upstream branch.  
To push the current branch and set the remote as upstream, use  
  
git push --set-upstream origin test_branch
```



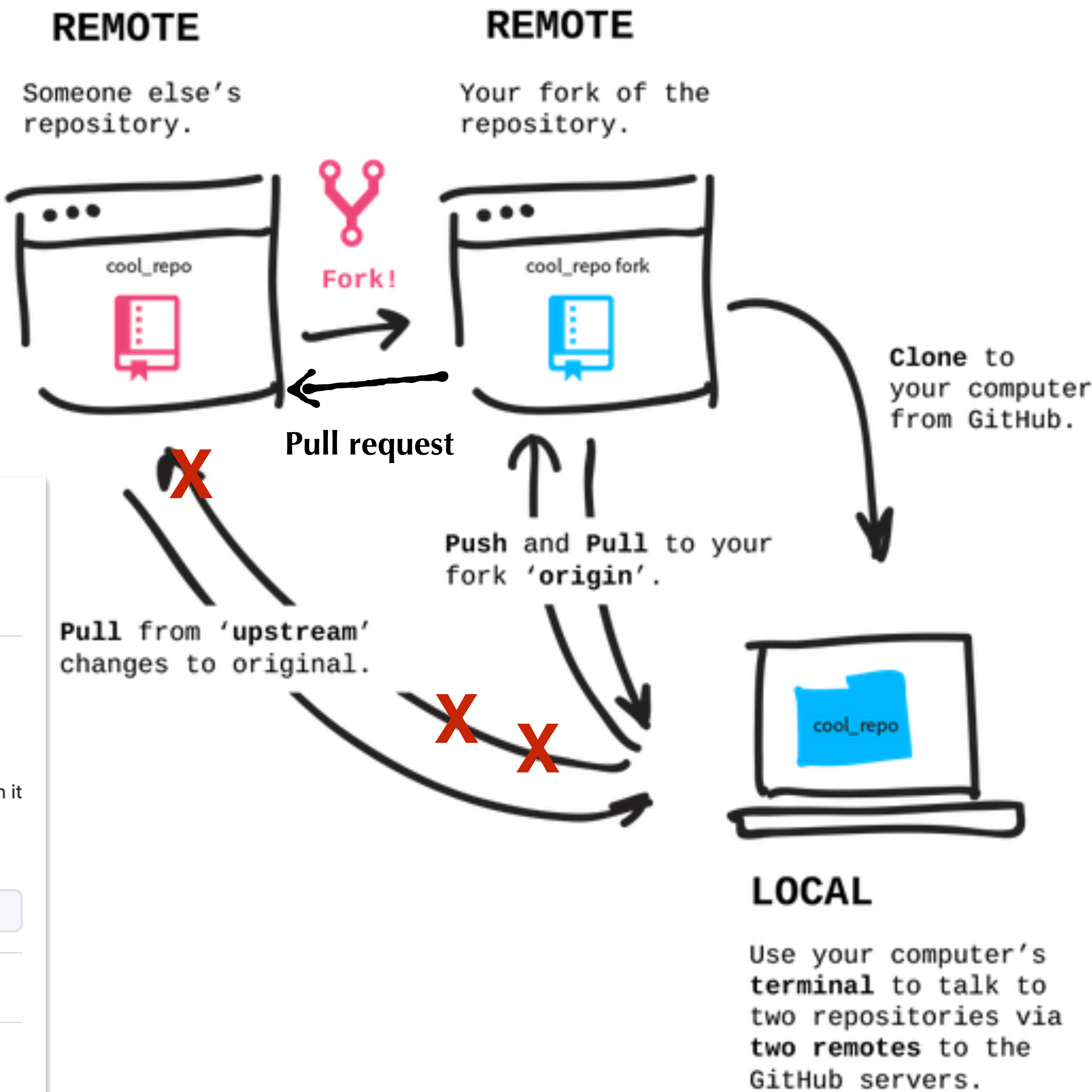
Pushing a branch

Only need to do this the first time you push a local branch.

```
(base) ?128 mine/sandbox2022 $ git push --set-upstream origin test_branch
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 317 bytes | 317.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'test_branch' on GitHub by visiting:
remote:   https://github.com/henrikef/sandbox2022/pull/new/test_branch
remote:
To github.com:henrikef/sandbox2022.git
 * [new branch]      test_branch -> test_branch
branch 'test_branch' set up to track 'origin/test_branch'.
```



Forking



Create a new fork

A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.

Owner * / **Repository name ***

By default, forks are named the same as their parent repository. You can customize the name to distinguish it further.

Description (optional)

i You are creating a fork in your personal account.

Create fork

Initializing a local repo (cloning)

```
(base) ✓ fermi_software/summer2022 $ mkdir git_sandbox
(base) ✓ fermi_software/summer2022 $ cd git_sandbox
(base) ✓ summer2022/git_sandbox $ mkdir mine
(base) ✓ summer2022/git_sandbox $ cd mine
(base) ✓ git_sandbox/mine $ git clone git@github.com:henrikef/sandbox2022.git
Cloning into 'sandbox2022'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
(base) ✓ git_sandbox/mine $ cd sandbox2022
(base) ✓ mine/sandbox2022 $ git remote -v
origin git@github.com:henrikef/sandbox2022.git (fetch)
origin git@github.com:henrikef/sandbox2022.git (push)
(base) ✓ mine/sandbox2022 $ git remote add upstream git@github.com:FermiSummerSchool/sandbox2022.git
(base) ✓ mine/sandbox2022 $ git remote -v
origin git@github.com:henrikef/sandbox2022.git (fetch)
origin git@github.com:henrikef/sandbox2022.git (push)
upstream git@github.com:FermiSummerSchool/sandbox2022.git (fetch)
upstream git@github.com:FermiSummerSchool/sandbox2022.git (push)
```

Nickname and url of "official" repository

Can access branches in the official repository as **upstream/branchname**

Remember this?

Only need to do this the first time you push a local branch.


```
(base) ?128 mine/sandbox2022 $ git push --set-upstream origin test_branch
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 317 bytes | 317.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'test_branch' on GitHub by visiting:
remote:   https://github.com/henrikef/sandbox2022/pull/new/test_branch
remote:
To github.com:henrikef/sandbox2022.git
 * [new branch]      test_branch -> test_branch
branch 'test_branch' set up to track 'origin/test_branch'.
```




Pull request

Open a pull request








Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

 base repository: FermiSummerSchool/sandbo... ▾ base: main ▾ ← head repository: henrikef/sandbox2022 ▾ compare: test_branch ▾


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


 **Test commit**

Write Preview


H B I       


Demonstrating branching and pull requests.

Attach files by dragging & dropping, selecting or pasting them. 

Allow edits by maintainers 

Create pull request ▾


 Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

Reviewers 

No reviews

Assignees 

No one—assign yourself

Labels 

None yet

Projects 

None yet

Milestone 

No milestone

Development 

Use [Closing keywords](#) in the description to automatically close issues

Tests and code review

1 commit 1 file changed 1 contributor

Commits on May 24, 2022

Test commit
henrikef committed 1 minute ago 278ea10

Showing 1 changed file with 2 additions and 0 deletions. Split Unified

2 README.md

@@ -1,2 +1,4 @@	
1 # sandbox2022	1 # sandbox2022
2 GitHub sandbox for the 2022 summer school	2 GitHub sandbox for the 2022 summer school
	3 + Hello world!
	4 + Hello world!

Add more commits by pushing to the **test_branch** branch on **henrikef/sandbox2022**.



Continuous integration has not been set up

[GitHub Actions](#) and [several other apps](#) can be used to automatically catch bugs and enforce style.



This branch has no conflicts with the base branch

Merging can be performed automatically.

Merge pull request



You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

Merging a pull request



Pull request successfully merged and closed

You're all set — the `henrikef:test_branch` branch can be safely deleted.

If you wish, you can also delete this fork of `FermiSummerSchool/sandbox2022` in the [settings](#).

Delete branch

Pull request requirements:

- Automated tests (GitHub actions) pass
- Code review
- Description/documentation of changes
- ...

Conflicts

Showing 1 changed file with 2 additions and 1 deletion.

3 README.md

Feature branch (in fork)

@@ -1,4 +1,5 @@

```
1 # sandbox2022
2 GitHub sandbox for the 2022 summer school
3 Hello world!
4 - Hello world!
```

```
1 # sandbox2022
2 GitHub sandbox for the 2022 summer school
3 Hello world!
4 +
5 + Hallo Welt!
```

Change to create a conflict

main

Meanwhile on main

henrikef committed 2 minutes ago

1 parent 962174e commit 580d1c1a5e5b6733df39c

Showing 1 changed file with 6 additions and 1 deletion.

7 README.md

@@ -1,4 +1,9 @@

```
1 # sandbox2022
2 GitHub sandbox for the 2022 summer school
3 Hello world!
4 - Hello world!
```

```
1 # sandbox2022
2 GitHub sandbox for the 2022 summer school
3 +
4 + ## Subheading
5 +
6 + Hello world!
7 Hello world!
8 +
9 + This is a new paragraph (note the empty line).
```

Conflicts

Showing 1 changed file with 2 additions and 1 deletion.

```
Feature branch (in fork)
... @@ -1,4 +1,5 @@
1 # sandbox2022
2 GitHub sandbox for the 2022 summer school
```



This branch has conflicts that must be resolved

Use the [web editor](#) or the [command line](#) to resolve conflicts.

Conflicting files

README.md

Resolve conflicts

Showing 1 changed file with 6 additions and 1 deletion.

```
7 README.md
... @@ -1,4 +1,9 @@
1 # sandbox2022
2 GitHub sandbox for the 2022 summer school
3 Hello world!
4 - Hello world!
1 # sandbox2022
2 GitHub sandbox for the 2022 summer school
3 +
4 + ## Subheading
5 +
6 + Hello world!
7 Hello world!
8 +
9 + This is a new paragraph (note the empty line).
```


Resolving conflicts

```
(base) ✓ mine/sandbox2022 $ git fetch upstream
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 7 (delta 0), reused 6 (delta 0), pack-reused 0
Unpacking objects: 100% (7/7), 1.26 KiB | 161.00 KiB/s, done.
From github.com:FermiSummerSchool/sandbox2022
 * [new branch]      main      -> upstream/main
(base) ?1 mine/sandbox2022 $ git merge upstream/main
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.
(base) ✓ mine/sandbox2022 $ cat README.md
# sandbox2022
GitHub sandbox for the 2022 summer school

## Subheading

Hello world!
Hello world!

<<<<<<< HEAD
Hallo Welt!
=====
This is a new paragraph (note the empty line).
>>>>>>> upstream/main
```

Ideally, we would get the upstream changes regularly during development, and before making the pull request.

Merging the changes in the main branch of the original repo into the current branch on the forked repo.

Conflicting lines are surrounded by
<<<<<<<<
=====
>>>>>>>>

Resolving conflicts

Manually fix conflicts and
remove conflict markers

```
(base) ✓ mine/sandbox2022 $ open README.md
(base) ✓ mine/sandbox2022 $ cat README.md
# sandbox2022
GitHub sandbox for the 2022 summer school
```

```
## Subheading
```

```
Hello world!
Hallo Welt!
```

```
This is a new paragraph (note the empty line).
```

```
(base) ✓ mine/sandbox2022 $ git add README.md
(base) ✓ mine/sandbox2022 $ git commit -m "Fixed conflicts"
[test_branch d9edcdd] Fixed conflicts
(base) ✓ mine/sandbox2022 $ git push
```

```
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 359 bytes | 359.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:henrikef/sandbox2022.git
 758872a..d9edcdd test_branch -> test_branch
```

The screenshot shows a GitHub pull request interface. At the top, there are navigation tabs for 'Conversation' (0), 'Commits' (2), 'Checks' (0), and 'Files changed' (1). A comment from 'henrikef' is visible, stating 'No description provided.' Below the comment, a commit history is shown with two entries: 'localization' (758872a) and 'Fixed conflicts' (d9edcdd). A green notification box at the bottom indicates 'This branch has no conflicts with the base branch' and provides a 'Merge pull request' button. A warning box above it states 'Continuous integration has not been set up'.

Resolving conflicts

```
(base) ✓ mine/sandbox2022 $ open README.md
(base) ✓ mine/sandbox2022 $ cat README.md
# sandbox2022
GitHub sandbox for the 2022 summer school

## Subheading

Hello world!
Hallo Welt!

This is a new paragraph (note the empty line)
(base) ✓ mine/sandbox2022 $ git add README.md
(base) ✓ mine/sandbox2022 $ git commit -m "Add new paragraph"
[test_branch d9edcdd] Fixed conflicts
(base) ✓ mine/sandbox2022 $ git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 359 bytes
Total 3 (delta 1), reused 0 (delta 0), pushed 3
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:henrikef/sandbox2022.git
 758872a..d9edcdd test_branch -> test_branch
```

Manually fix conflicts and

The screenshot shows a GitHub pull request interface. At the top, there are navigation tabs: Conversation (0), Commits (2), Checks (0), and Files changed (1). Below this, a comment from user 'henrikef' is shown, stating 'No description provided.' The commit history shows two commits: 'localization' (758872a) and 'Fixed conflicts' (d9edcdd). A green box at the bottom contains the following information: 'Continuous integration has not been set up' (with a link to GitHub Actions), 'This branch has no conflicts with the base branch' (with a note that merging can be performed automatically), and a green 'Merge pull request' button. A link to 'open this in GitHub Desktop' or view 'command line instructions' is also present.

More on conflicts

- Conflicts appear if the same line(s) are edited in multiple branches.
- Conflicts can also appear when deleting/moving files.
- Git will not catch all *logical* conflicts (e.g., incrementing the same number multiple times in different places!)
- Other ways of solving conflicts:
 - To retain upstream version: **git checkout --theirs <filename>**
 - To retain your version: **git checkout --ours <filename>**
 - Web interface, see [GitHub documentation](#).

Content

1. Intro to git
2. Git(hub) for users
- 3. Git(hub) for developers**
 - A. Branching, forking, and pull requests
 - B. Undoing changes**

Undoing changes

- Undo local changes: **git checkout -- <file>** or **git restore <file>**
- Undo staging: **git restore --staged <file>**
- Undo a (local) commit: **git revert <commit sha>**
- Change last commit message: **git commit --amend**
- Lots more options including **git reset** and **git stash**: See https://docs.gitlab.com/ee/topics/git/numerous_undo_possibilities_in_git/
- You can easily close (and re-open) pull requests made in error.
- It is even possible to revert merged pull requests!
- If needed, you can even change history to remove all traces of a given commit (e.g. if passwords are committed in plaintext)

Undoing changes

```
(base) ✓ mine/sandbox2022 $ open README.md
(base) ✓ mine/sandbox2022 $ git diff
diff --git a/README.md b/README.md
index 677f9fa..a38af9c 100644
--- a/README.md
+++ b/README.md
@@ -3,7 +3,7 @@ GitHub sandbox for the 2022 summer school

## Subheading

-Hello world!
+Hello wrld!
Hallo Welt!

This is a new paragraph (note the empty line).
(base) ✓ mine/sandbox2022 $ git status
On branch test_branch
Your branch is up to date with 'origin/test_branch'.

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")
```

Undoing changes

```
(base) ✓ mine/sandbox2022 $ git add README.md
(base) ✓ mine/sandbox2022 $ git status
On branch test_branch
Your branch is up to date with 'origin/test_branch'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
   modified:   README.md

(base) ✓ mine/sandbox2022 $ git commit -m "Committing a typo on accident"
[test_branch 2cc3c7b] Committing a typo on accident
 1 file changed, 1 insertion(+), 1 deletion(-)
(base) ✓ mine/sandbox2022 $ git revert 2cc3c7b
[test_branch 814087a] Revert "Committing a typo on accident"
 1 file changed, 1 insertion(+), 1 deletion(-)
(base) ✓ mine/sandbox2022 $ git status
On branch test_branch
Your branch is ahead of 'origin/test_branch' by 2 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

A new commit is created to reverse the previously committed changes.

Reverting pull requests

localization #2

Merged henrikef merged 2 commits into `FermiSummerSchool:main` from `henrikef:test_branch` 22 minutes ago

Conversation 0 Commits 2 Checks 0 Files changed 1

henrikef commented 42 minutes ago Member

No description provided.

henrikef added 2 commits 43 minutes ago

- localization 758872a
- Fixed conflicts d9edcdd

henrikef merged commit `0cf17aa` into `FermiSummerSchool:main` 22 minutes ago Revert

A new pull request is created to reverse the previously merged changes.

Pull request successfully merged and closed
You're all set — the `henrikef:test_branch` branch can be safely deleted.
If you wish, you can also delete this fork of `FermiSummerSchool/` [settings](#).

Revert "localization" #3

Open henrikef wants to merge 1 commit into `main` from `revert-2-test_branch`

Conversation 0 Commits 1 Checks 0 Files changed 1 +1 -1

Changes from all commits File filter Conversations Jump to

0 / 1 files viewed Review changes

```
@@ -4,6 +4,6 @@ GitHub sandbox for the 2022 summer school
4   ## Subheading
5
6   Hello world!
7 - Hallo Welt!
8
9   This is a new paragraph (note the empty line).
```

What's next?

Go to <https://github.com/FermiSummerSchool/sandbox2022>

Make your own fork, play around and make a pull request!

Fermi summer school repo (with notebooks) is here:

<https://github.com/FermiSummerSchool/fermi-summer-school>

If you notice an error, please open an issue or send a pull request with the fix!

Backup

git/svn Differences

- ▶ **Branches** are used *much more* in git than in svn.
 - All work you do should be in a branch.
 - Switching between branches is easy; you can switch between multiple topics or back out a set of changes without losing them.
- ▶ **Committing changes** affects *only your local copy* of the repository.
 - Changes are only shared when you “push” them to github.
- ▶ Before changes are entered into the master (trunk) branch, you make a “**pull request**,” which allows others to see the changes and comment on them.

Retroactively Branching

- ▶ Suppose you make changes to the master branch and then realize you want to retroactively turn the changes into a branch. Don't worry, here is how you do it:

```
# Commit what you've got
```

```
git commit -am "commit message"
```

```
# Create a new branch @ this point
```

```
git branch newbranch
```

```
# Reset your local master to GitHub
```

```
git reset --hard origin/master
```

```
# Checkout your branch and keep working!
```

```
git checkout newbranch
```

Getting Someone's Branch

- ▶ Here is how to get a local copy of a branch that someone else pushed to GitHub:

```
git fetch
```

```
git checkout branchname
```

Syncing with the Master

- ▶ Suppose updates were made to the GitHub master while you were working on your branch. Here is how to keep things in sync:

```
# Commit what you've got  
git commit -am "my change log"
```

```
# Checkout master & pull changes  
git checkout master  
git pull origin master
```

```
# Checkout your branch and merge in changes:  
git checkout your_branch  
git merge --no-ff master
```

Workflow

1. Open an **issue** on GitHub explaining what you want to do (avoids duplication).
2. Create a **branch** in your local copy.
3. Make changes in the branch.
4. **Add** and **commit** your changes frequently.
5. **Push** your branch to github.
6. Submit a **merge** ~~pull~~ **request** when you're ready to have your changes reviewed and merged with the master branch. **Use "WIP" tag!**