

Version control using Git

<https://github.com/nih-fmrif/git-training>

Session 2

12/12/2019

Data Science and Sharing Team
NIMH

Logistics

One more session:

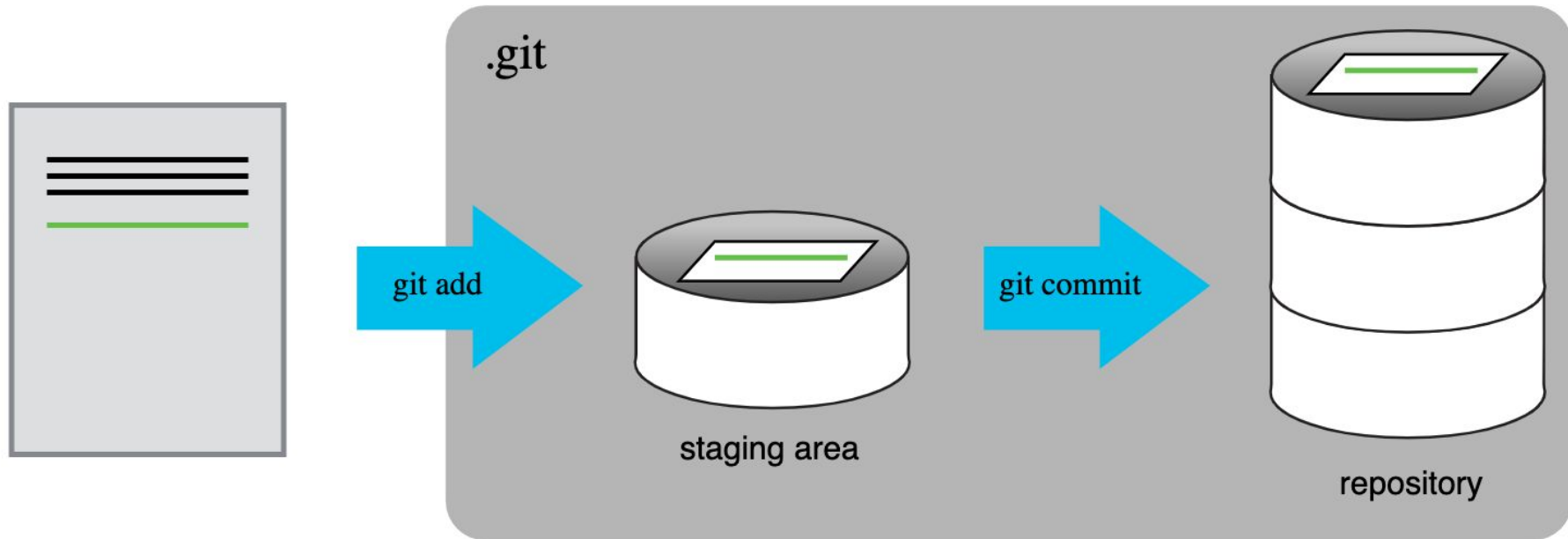
2/19 from 1:30 - 3:30, same room (B1C208)

Today:

- Lecture - Review, going back in time in your local repo
- Walkthrough - Going back in time
- Lecture - Github, documentation, and remote changes
- Walkthrough - Fork, edit, pull
- Exercise

<https://github.com/nih-fmrif/git-training>

Review: repositories and .git



Review: git workflow

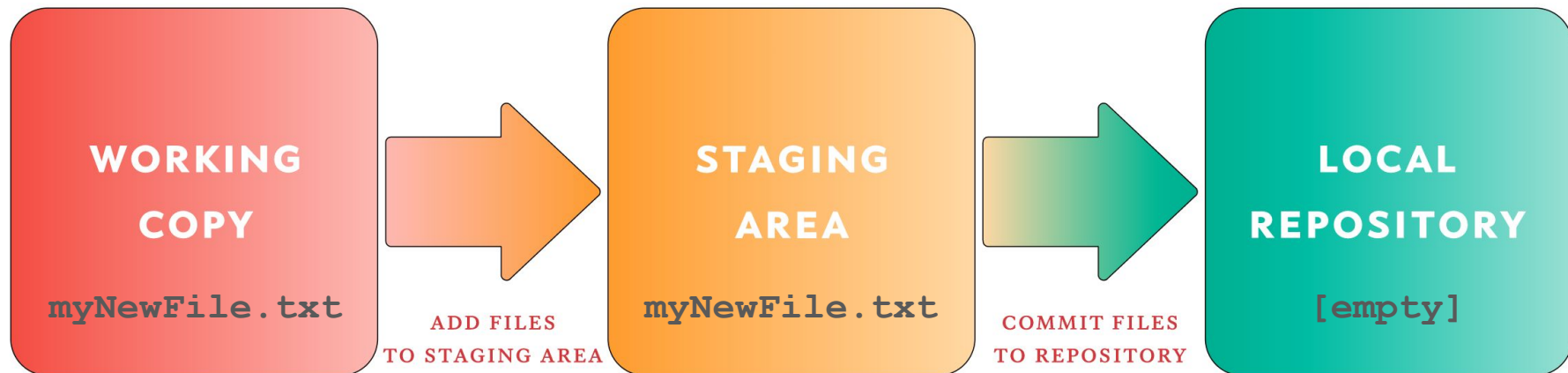
- Change your files
(edit or create new)



Review: git workflow

- Add changes to staging area

```
git add myNewFile.txt
```



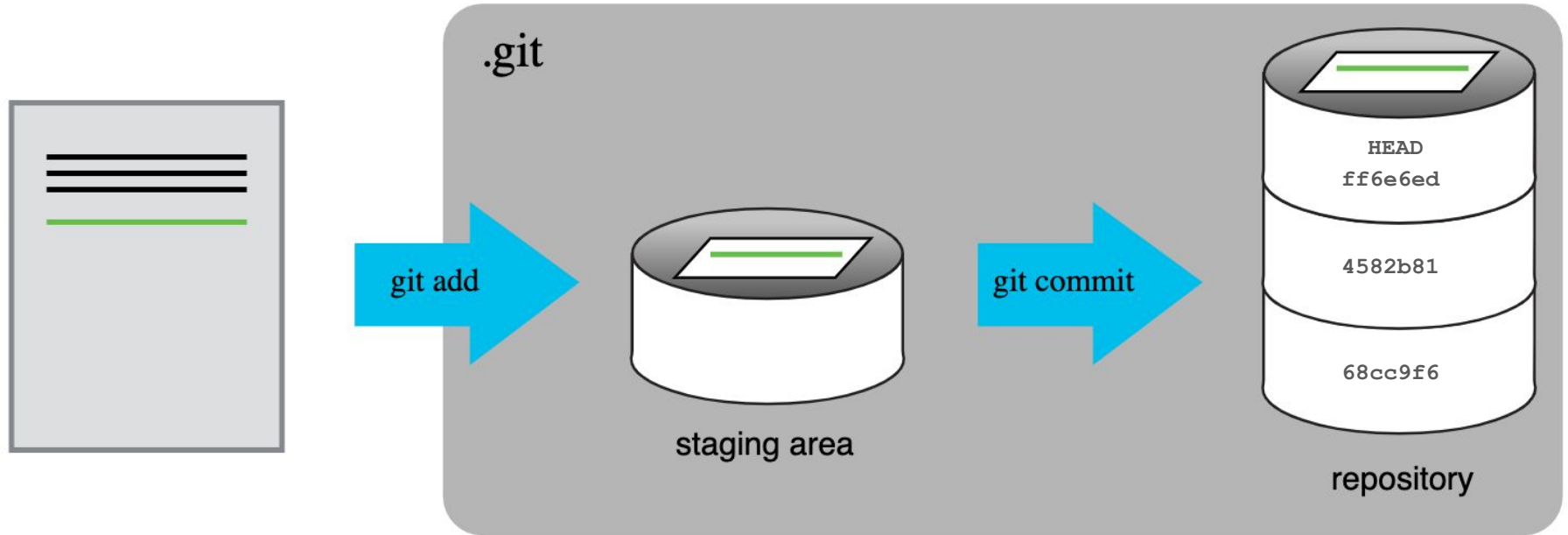
Review: git workflow

- Commit the changes

```
git commit -m "adding  
myNewFile.txt"
```



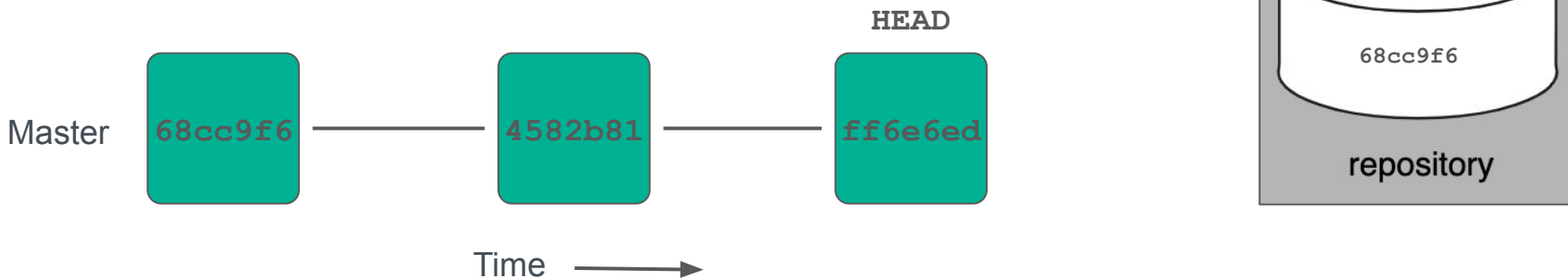
Review: git workflow



Commit history

- *HEAD* - most recent commit
- Commit IDs
 - 40-character Secure Hash Algorithm 1 (SHA-1)
 - Typically referenced using first 7 characters

ff6e6edf28e4947a6f38c3dfac5614e215725b52



Git commands: checkout

Switch between different versions of a target entity

- **Files**, commits, or branches
- Changes the contents of the files and working directory to reflect target time point

```
git checkout <commit> <file>
```

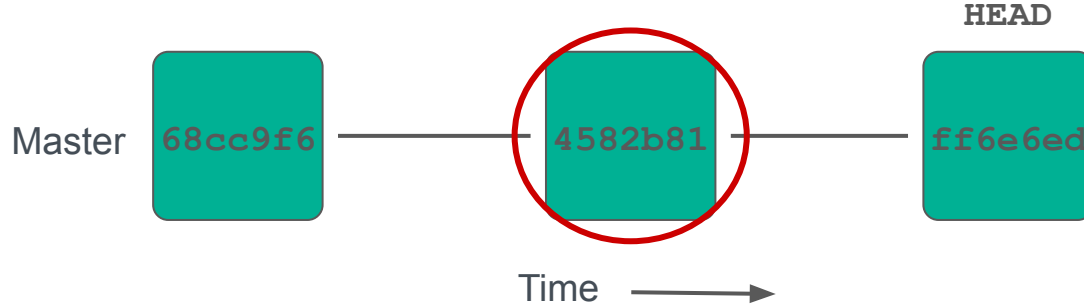
Git commands: checkout

Switch between different versions of a target entity

- Files, **commits**, or branches
- Changes the contents of the files and working directory to reflect target time point

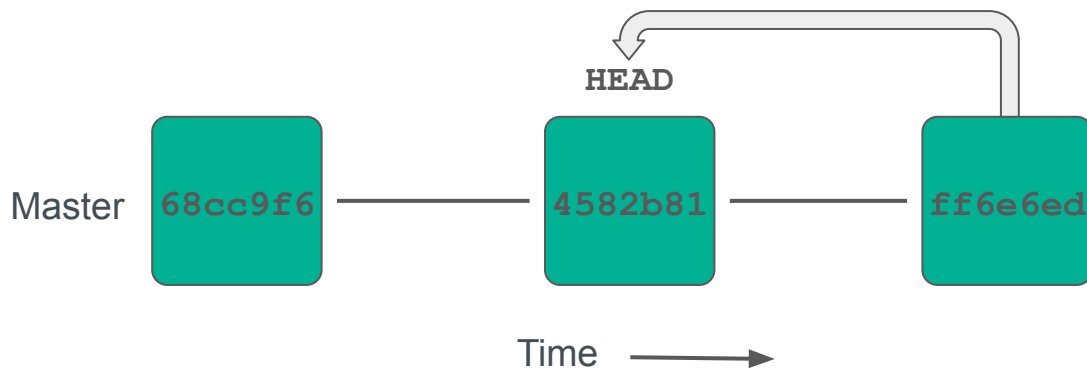
```
git checkout <commit>
```

Git commands: checkout



* You can checkout individual file histories or the entire repository

Git commands: checkout

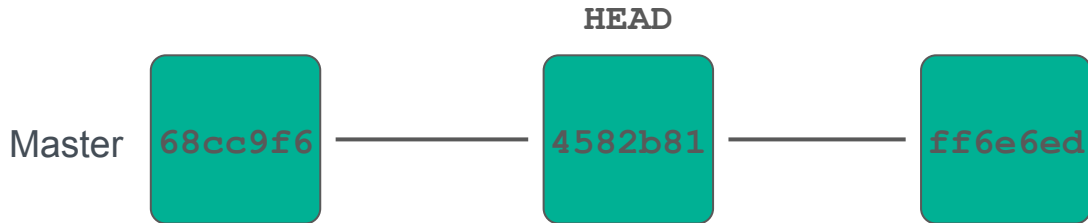


```
git checkout 4582b81
```

Git commands: checkout

Caution!

- `checkout` is useful for *inspecting*
- Changing files will put the HEAD in a detached state and you will lose changes if you commit them or change HEAD
- Solution: create a branch (more next week)



Where do changes go?

Git concept: ignoring files

What about files you do not want to track?
.gitignore

- Hidden text file in the root of the repository that contains filenames to be ignored from versioning
 - Hidden files created by OS
 - Temporary or intermediate files
 - Data
- Accepts wildcards and directories

```
sample/  
├── .RData  
├── .Rhistory  
├── .git  
└── myNewFile.R
```

```
sample/  
├── .RData  
├── .Rhistory  
├── .git  
├── .gitignore  
└── myNewFile.R
```

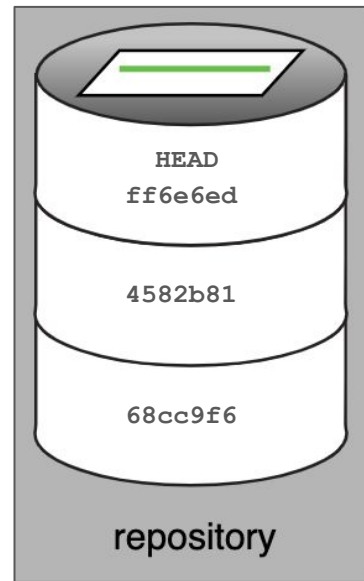
Example

Going back in time
and ignoring things

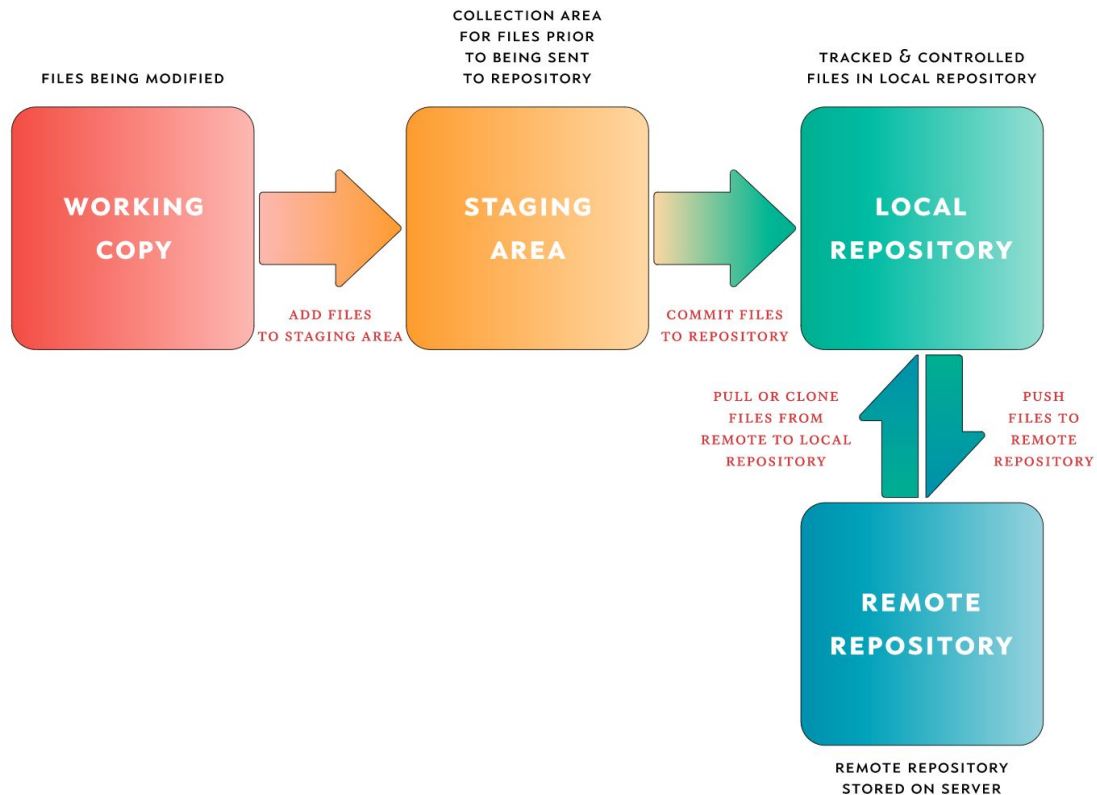
[https://github.com/nih-fmrif/git-training/tree/master/
session_2#example-1](https://github.com/nih-fmrif/git-training/tree/master/session_2#example-1)

Best practices: commits

- Commit often
- Commit related changes
- Do not commit half done work*
- Use informative commit messages
 - Agree on a workflow
 - Possible message tagging (eg *incremental build*)
- Test before you commit



Review: git workflow



GitHub

The screenshot shows the GitHub interface for the repository 'practicalseries-lab/lab-01-website'. The page is divided into several sections:

- NAVIGATION BAR:** Includes the GitHub logo, a search bar, and links for 'Pull requests', 'Issues', and 'Gist'.
- Collaboration buttons:** Buttons for 'Watch' (1), 'Star' (0), and 'Fork' (0).
- PROJECT TABS:** A row of tabs including 'Code', 'Issues' (0), 'Pull requests' (0), 'Projects' (0), 'Wiki', 'Pulse', 'Graphs', 'Settings', and 'PROJECT TABS'.
- Repository description and edit:** A text area with the description 'A simple website repository used as an example to demonstrate how the Brackets-Git extension works' and an 'Edit' button.
- STATUS BAR:** A horizontal bar showing repository statistics: '14 commits', '1 branch', '13 releases', and '1 contributor'.
- CONTROL BUTTONS:** A row of buttons including 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find file', and 'Clone or download'.
- LATEST COMMIT (HEAD INFORMATION):** A section showing the latest commit 'P04: Published (merged d-03-contact)' with the commit hash 'da038e3' and the time '2 days ago'.
- PROJECT CONTENT AREA:** A table listing files and their commit history.
- README.md FILE CONTENTS:** A section for the README file.

File	Commit	Time
11-resources	D: Incremental build - 03-contact page	3 days ago
.gitignore	D01: Initial Commit	11 days ago
01-intro.html	D: Incremental Build - 01-intro page	10 days ago
02-about.html	D: Incremental Build - 02-about page	9 days ago
03-contact.html	D: Incremental Build - 03-contact page	4 days ago
README.md	D: Incremental Build - README.md modified	3 days ago
index.html	P01: First Publication	10 days ago

GitHub

dmoracze / HPC_helper_tools

Unwatch 1 Star 1 Fork 1

Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security Insights Settings

Branch: master

Commits on Oct 4, 2019

- updated README to say that VNC is optional
dmoracze committed on Oct 4 [b51a4e8](#)
- support for multiple tunnels for connect, fixed typo for bids
dmoracze committed on Oct 4 [4581b81](#)

Commits on Sep 23, 2019

- fixed typos
dmoracze committed on Sep 23 [69a0f20](#)
- updating README to include new ports and slurm environment
dmoracze committed on Sep 23 [958c4af](#)
- added multiple port options and slurm environment
dmoracze committed on Sep 23 [ec4d36c](#)

Commits on Sep 16, 2019

- removed scripts from top dir
dmoracze committed on Sep 16 [434119a](#)
- moved scripts to bin
dmoracze committed on Sep 16 [1bc2603](#)

GitHub

support for multiple tunnels for connect, fixed typo for bids

master

dmoracze committed on Oct 4 1 parent 69a0f20 commit 4581b814b384efc384cb30e57f57d06022bbfe73

Showing 2 changed files with 45 additions and 10 deletions. Unified Split

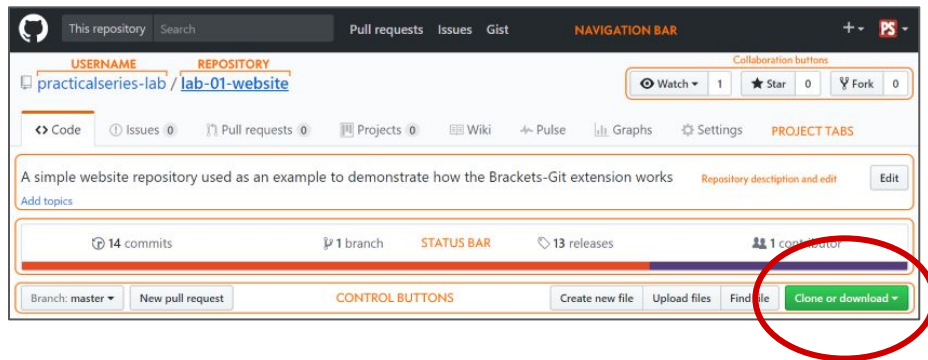
bin/is_bids.sh

```
@@ -48,7 +48,7 @@ for i in $@; do :
48 48         echo
49 49         echo "ERROR: Unknown option: $1"
50 50         usage
51 -         echo 1
51 +         exit 1
52 52     esac
53 53 done
54 54
```

bin/spersist-connect.sh

```
@@ -4,7 +4,14 @@
4 4     ssh ${USER}@biowulf.nih.gov "test -e ~/.spersist"
5 5
6 6     # download remote .spersist and connect
7 - if [ $? -eq 0 ]; then
7 + if [ $? -eq 1 ]
8 + then
9 +     echo
10 +     echo "ERROR: Session information not found in biowulf.nih.gov:~/.spersist"
11 +     echo "Please run 'spersist-store.sh' within spersist session to record session information"
12 +     echo
13 +     exit 1
14 + else
8 15     echo
9 16     echo "Updating .spersist..."
10 17     echo
```

Git commands: clone

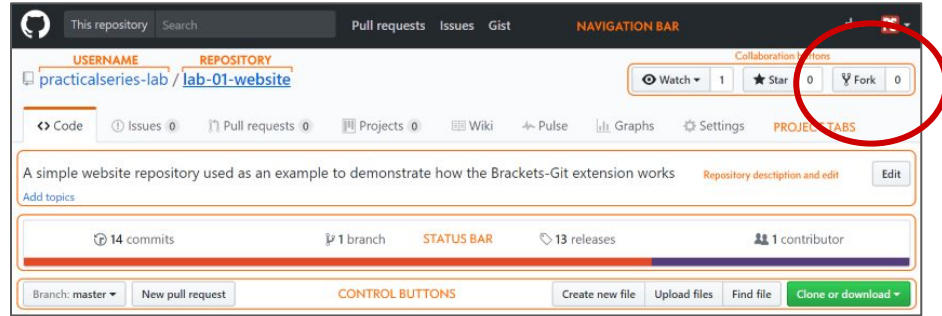


Download an existing remote repository to local

- Good practice to clone repo into a directory of the same name
- Changes will be pushed to the origin of the clone
- This should look familiar to last class's exercise

```
git clone <repo_url>
```

GitHub function: fork

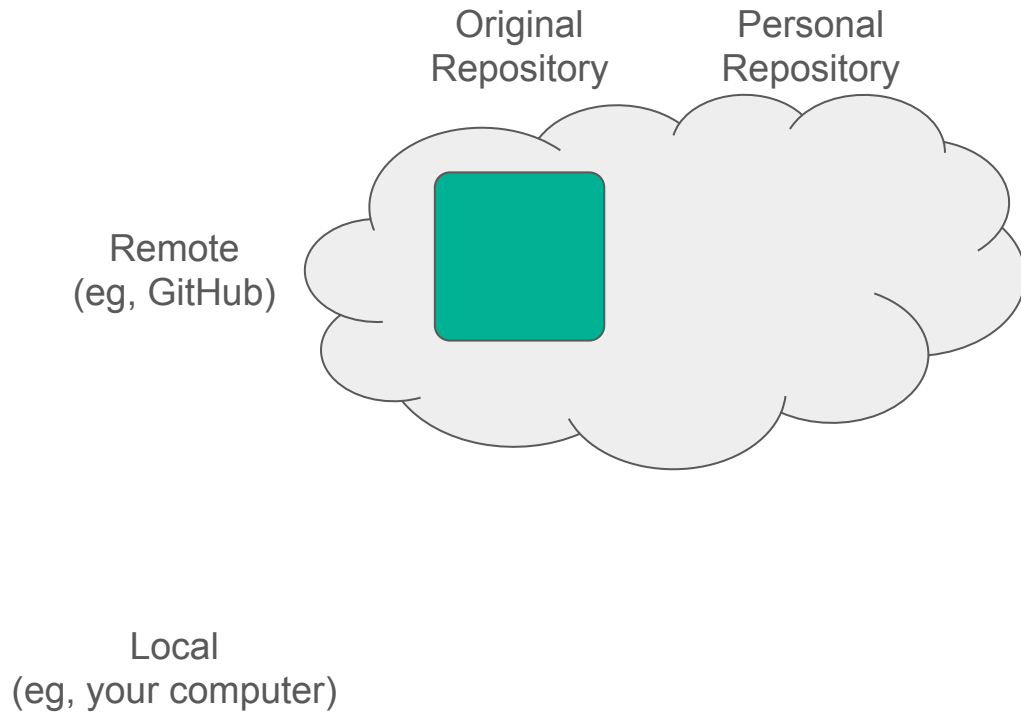


What if you want your own copy?

- Forking creates a personal copy of a repo, with a link back to the original
- You can then `clone` your personal copy to your local computer

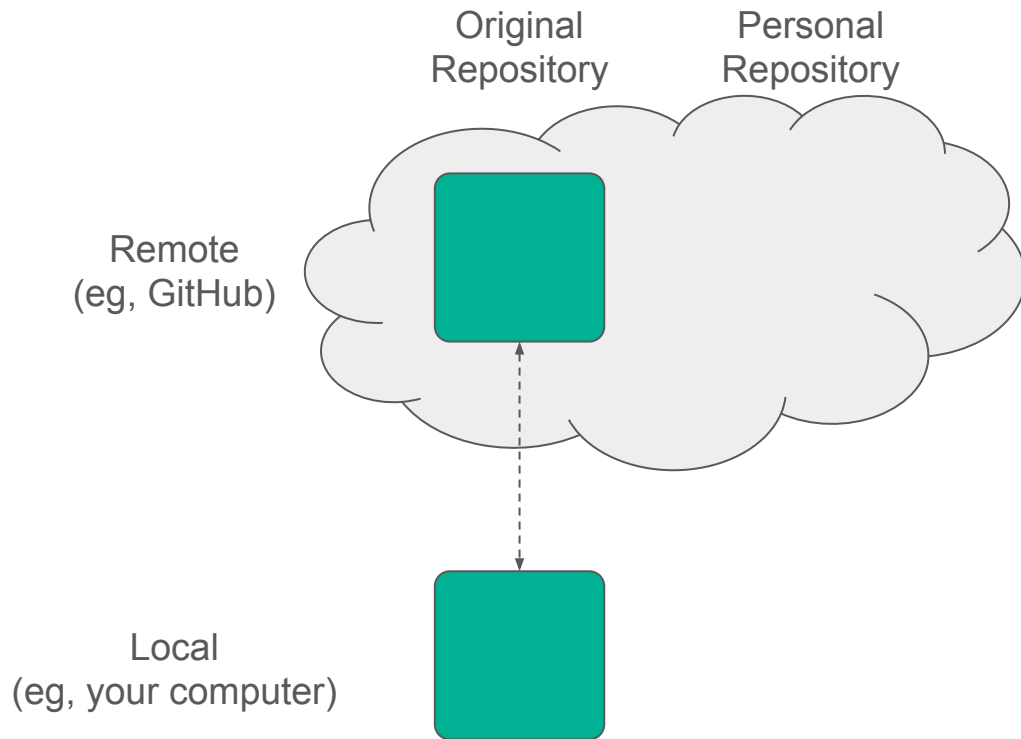
Click 'Fork' button (no command line)

Clone versus Fork



Git operation

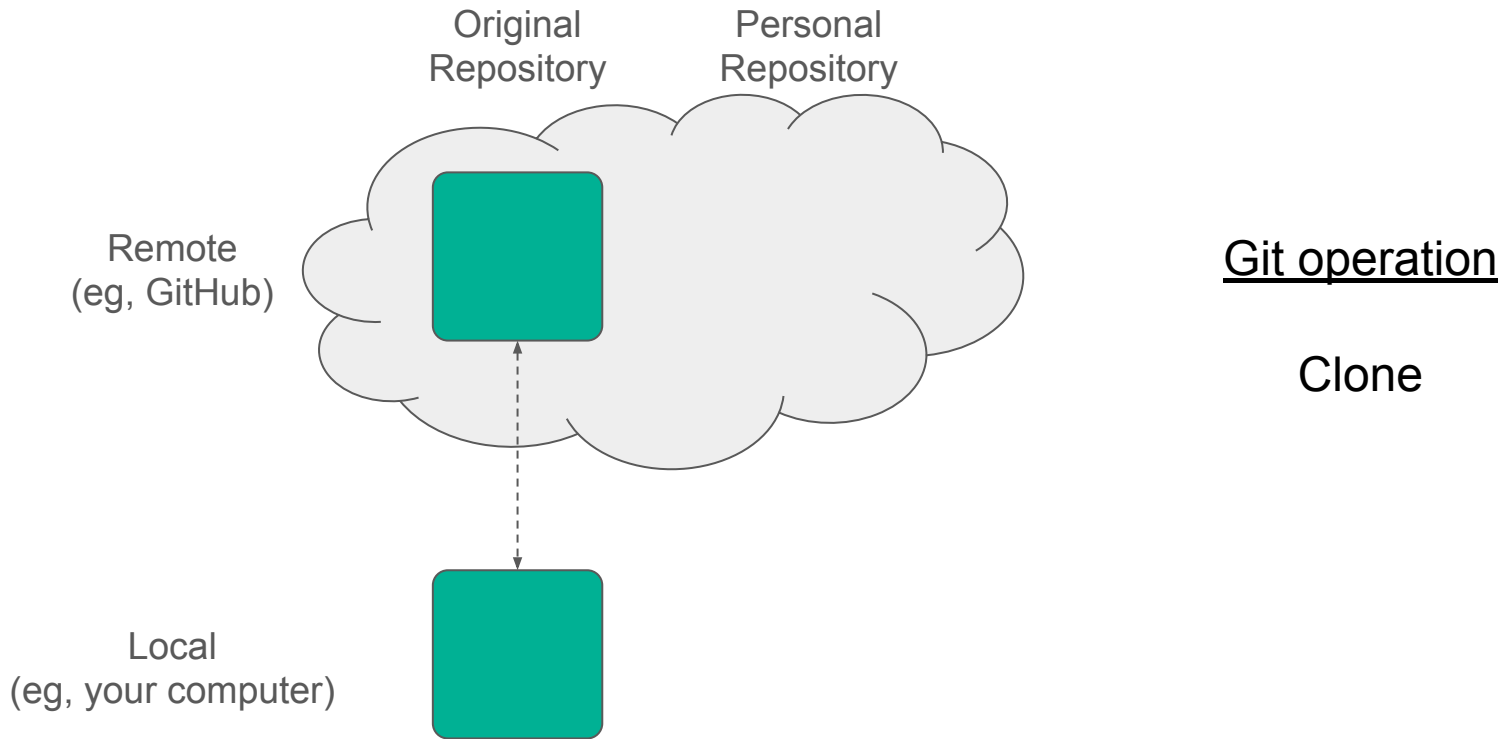
Clone versus Fork



Git operation

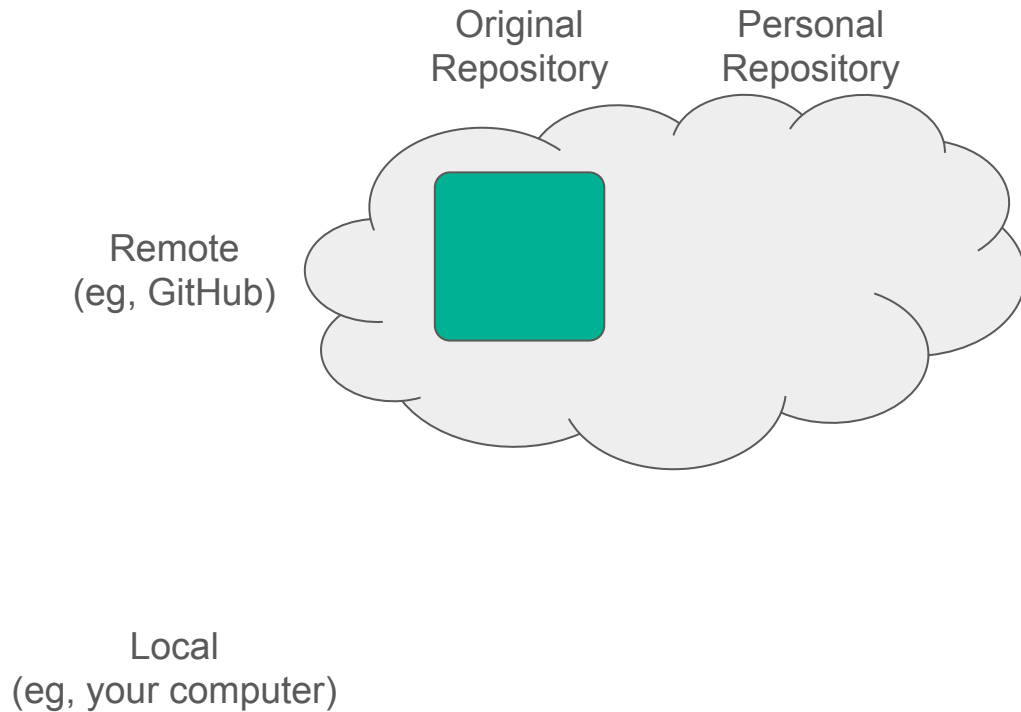
Clone

Clone versus Fork



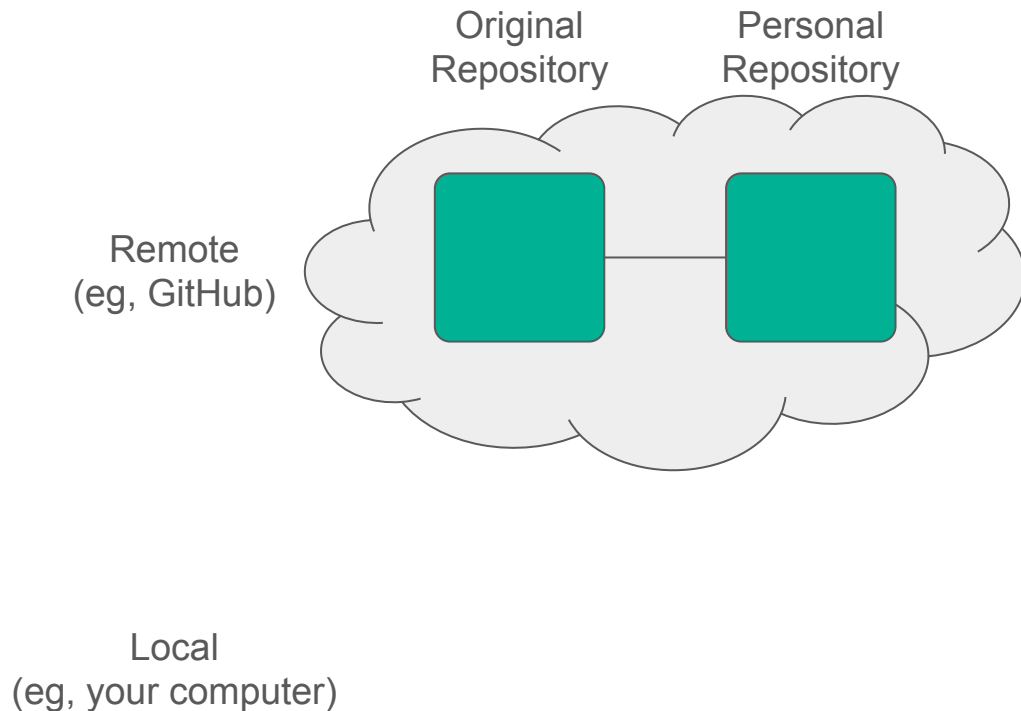
What if you want to push changes to remote?

Clone versus Fork



Git operation

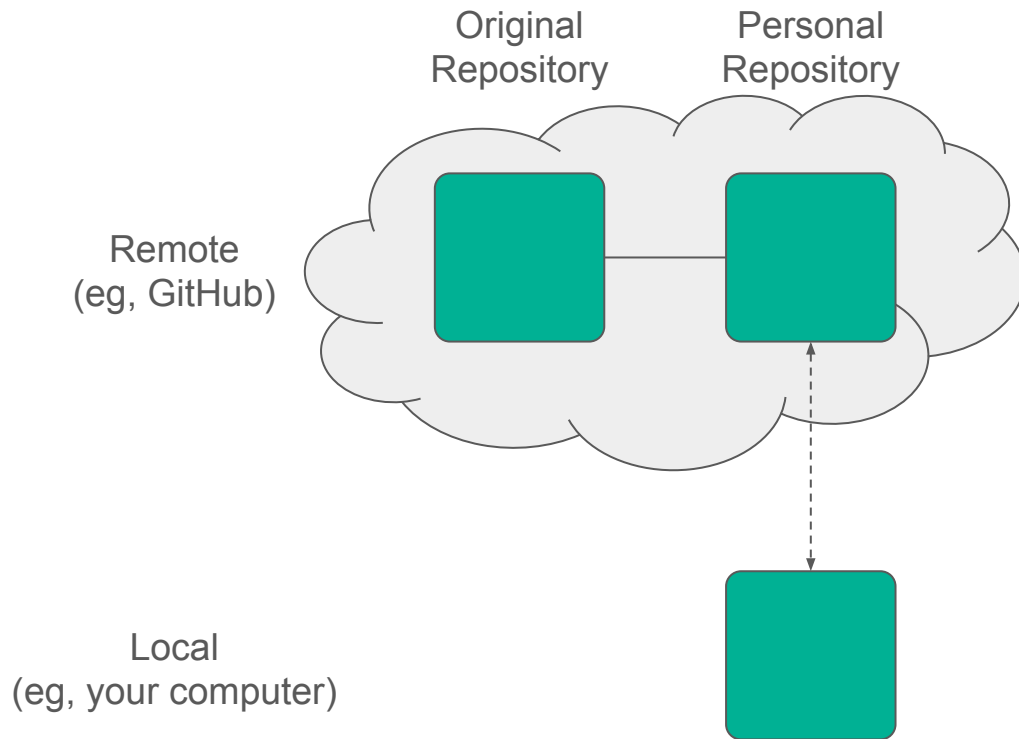
Clone versus Fork



Git operation

Fork

Clone versus Fork

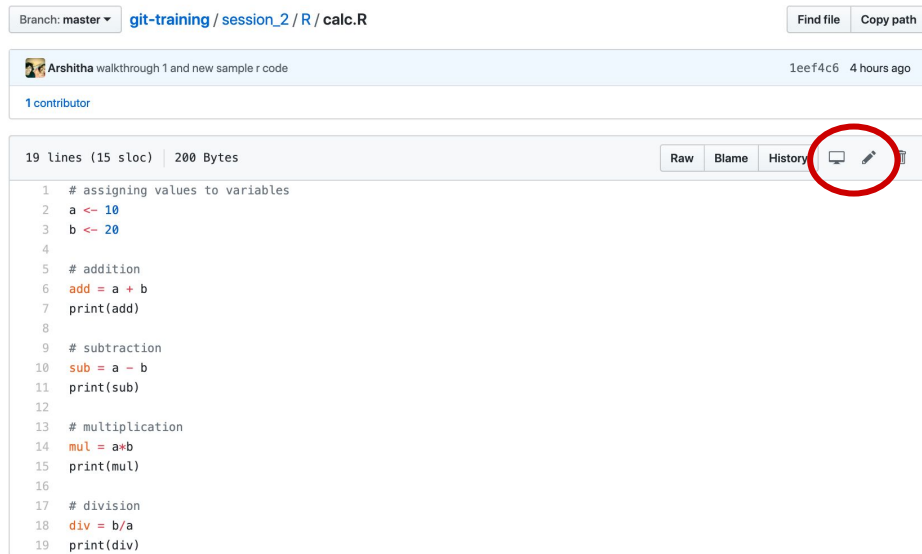


Git operation

Fork
Clone

Editing files on GitHub

- All text files (eg, code, markdown, etc...) can be edited remotely
- To save changes, you'll need to add a commit
- What do you think happens to a local repo when a remote commit is made?



The screenshot shows the GitHub interface for editing a file named `calc.R` in the `git-training / session_2 / R /` directory. The file is 19 lines long (15 sloc) and 200 Bytes. The code is as follows:

```
1 # assigning values to variables
2 a <- 10
3 b <- 20
4
5 # addition
6 add = a + b
7 print(add)
8
9 # subtraction
10 sub = a - b
11 print(sub)
12
13 # multiplication
14 mul = a*b
15 print(mul)
16
17 # division
18 div = b/a
19 print(div)
```

At the top right, there are buttons for "Find file" and "Copy path". Below the file name, it says "1 contributor". On the right side of the code editor, there are tabs for "Raw", "Blame", and "History". A red circle highlights the "Edit" icon (a pencil) next to the "History" tab.



Commit changes

Update calc.R

Add an optional extended description...

- ☒ Commit directly to the `master` branch.
- ☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Commit changes

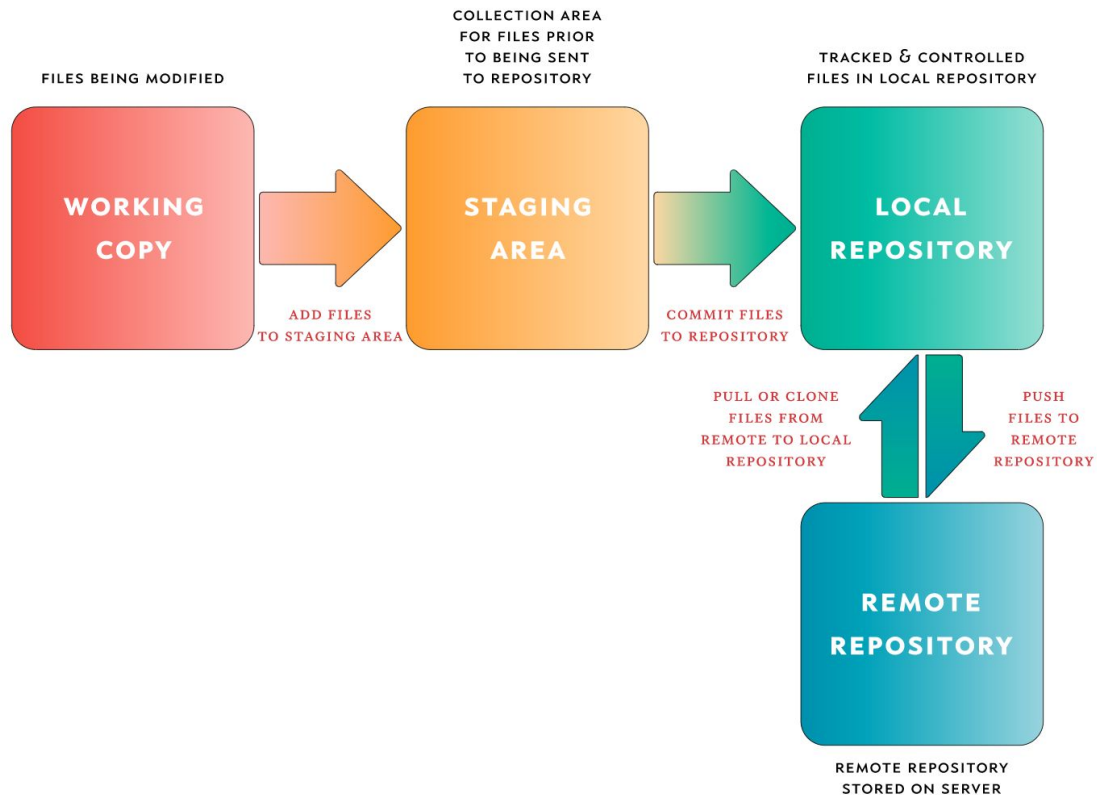
Cancel

Git commands: pull

- Pull changes from a remote repository to your local copy
- Simplest example, one branch cloned from origin: `git pull`
- Could create conflicts (more next time)

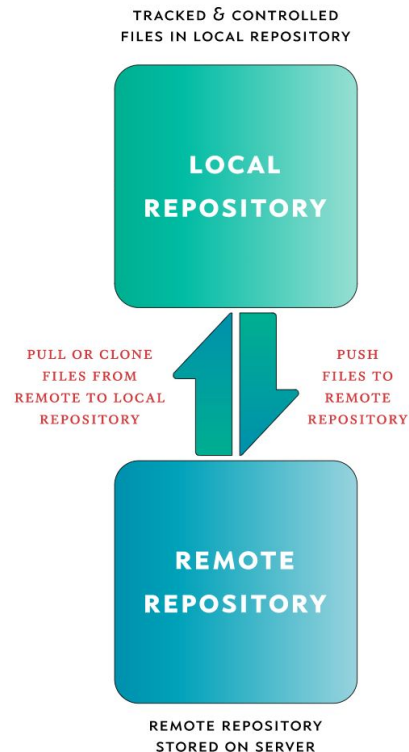
```
git pull <remote> <branch>
```

Review: git workflow



Best practices: remote

- Use the same same for local and remote repositories
- Update local repository frequently
 - Store local changes (`commit`)
 - Update local repo (`pull`)
- Update and resolve conflicts before pushing changes
 - Store local changes (`commit`)
 - Update local (`pull`)
 - Update remote (`push`)



Documentation

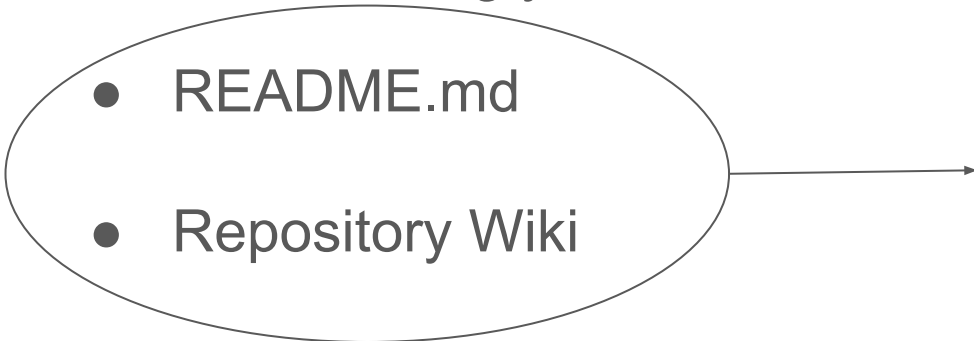
3 Levels

- Commenting your code (which you should be doing already, right?)
- README.md
- Repository Wiki

Documentation

3 Levels


- ~~Commenting your code~~ (which you should be doing already, right?)

- 
- A diagram consisting of an oval containing two bullet points, with an arrow pointing from the right side of the oval to the text 'Both styles are repository-level documentation'.
- README.md
 - Repository Wiki

Both styles are
repository-level
documentation

Documentation: README.md

```
--
27  ### creating local repository
28  ```shell
29  # creating a directory on the desktop
30  cd ~/Desktop
31  mkdir sample
32  cd sample
33  ls -a
34  ```
35  * Right now this is just another directory on my desktop. How do I make this a repo?
36
37  ```shell
38  git init
39  ls -a
40  ```
41
42  * .git sub-directory within the directory is what makes it a repository
43
44  ### creating a new file and tracking it with git
45  ```shell
46  touch myNewFile.txt
47  echo 'Hello, world!' > myNewFile.txt
48
49  # checking which files are tracked and which aren't
50  git status
51
52  # adding a file to the staging area
53  git add myNewFile.txt
54
55  git status
56
57  # committing the file
58  git commit -m "test file commit"
59
60  git status
61
62  vi myNewFile.txt
63  # add 2 new lines of text
64
65  # differences made after the last commit
66  git diff
67
68  git status
69
70  git add myNewFile.txt
71
72  git commit -m "adding two new lines"
73  ```
```



creating local repository

```
# creating a directory on the desktop
cd ~/Desktop
mkdir sample
cd sample
ls -a
```

- Right now this is just another directory on my desktop. How do I make this a repo?

```
git init
ls -a
```

- .git sub-directory within the directory is what makes it a repository

creating a new file and tracking it with git

```
touch myNewFile.txt
echo 'Hello, world!' > myNewFile.txt

# checking which files are tracked and which aren't
git status

# adding a file to the staging area
git add myNewFile.txt

git status

# committing the file
git commit -m "test file commit"

git status

vi myNewFile.txt
# add 2 new lines of text

# differences made after the last commit
git diff

git status

git add myNewFile.txt

git commit -m "adding two new lines"
```

Documentation: Wiki

[bids-standard](#) / [bids-starter-kit](#)

[Watch](#) 25 [Star](#) 97 [Fork](#) 49

[Code](#) [Issues 18](#) [Pull requests 2](#) [Actions](#) [Projects 1](#) [Wiki](#) [Security](#) [Insights](#)

Editing Home

[Page History](#) [New Page](#)

Home

Write Preview

[h1](#) [h2](#) [h3](#) [Link](#) [Image](#) [B](#) [i](#) [Code](#) [List](#) [Table](#) [Quote](#) [Media](#) [Help](#) Edit mode: [Markdown](#)

```
<p align="center">
  
</p>
```


</p>

```
<h3> Welcome to the home page of the wiki! </h3>
```

This is a community edited source of information and collection of links to help you get started with BIDS. Some resources include a description of BIDS-compatible folder structures, a glossary to help with some of the technical terms, and some tips and tricks to get you started.

The various pages in this wiki can be found by clicking "pages" at the top right side of this screen 📄

We also have a list of the most common places to start in the sidebar (also on the right 📄)

Please note that this wiki is actively maintained and still growing, so please help us add anything that you feel would be useful! You can read more information about [how to contribute in our guidelines] (<https://github.com/INCF/BIDS-Starter-Kit/blob/master/CONTRIBUTING.md>).

Edit message

Write a small message here explaining this change. (Optional)

[Save Page](#)

[bids-standard](#) / [bids-starter-kit](#)

[Watch](#) 25 [Star](#) 97 [Fork](#) 49

[Code](#) [Issues 18](#) [Pull requests 2](#) [Actions](#) [Projects 1](#) [Wiki](#) [Security](#) [Insights](#)

Home

Kirstie Whitaker edited this page on Oct 3, 2018 - 15 revisions

[Edit](#) [New Page](#)

BIDS STARTER KIT WIKI
Simple tips and references for an easy start to BIDS

Welcome to the home page of the wiki!

This is a community edited source of information and collection of links to help you get started with BIDS. Some resources include a description of BIDS-compatible folder structures, a glossary to help with some of the technical terms, and some tips and tricks to get you started.

The various pages in this wiki can be found by clicking "pages" at the top right side of this screen 📄

We also have a list of the most common places to start in the sidebar (also on the right 📄)

Please note that this wiki is actively maintained and still growing, so please help us add anything that you feel would be useful! You can read more information about [how to contribute in our guidelines](#).

Want to add to this wiki? Check out our [contributing guidelines](#)

For any questions, please [contact us](#)

Pages 16

[Home](#)[BIDS Apps Help](#)[BIDS Extensions](#)[BIDS Validator info](#)[Community Resources](#)[Contact](#)[Creating a BIDS compatible iEEG dataset](#)[Dataset Examples](#)[FAQ](#)[Glossary](#)[Metadata file formats](#)[Publications](#)[The BIDS folder hierarchy](#)[Tutorials](#)[Tutorials FAQ](#)[Useful Links](#)[Home](#)[Frequently Asked Questions](#)

Documentation: Which one?

README.md

- 'Front page' of project
- Markdown syntax
- Files stored in repo
- Editing requires changes to repo

Wiki

- In depth documentation
- Markdown syntax
- Files stored remotely
- Edits can be made freely

Really, it comes down to preference...

Example

Forking, cloning, remote,
and local changes

[https://github.com/nih-fmrif/git-training/tree/master/
session_2#example-2](https://github.com/nih-fmrif/git-training/tree/master/session_2#example-2)

Exercise

Putting it all together

[https://github.com/nih-fmrif/git-training/tree/master/
session_2#exercise](https://github.com/nih-fmrif/git-training/tree/master/session_2#exercise)