



METIS

INTRODUCTION TO GIT & GITHUB



GIT != GITHUB



- Git and Github are two separate tools that accomplish two separate goals
- They interface with one another to make a nice pipeline for team coding that is less likely to catastrophically fail



Let's see how git
can help us out



```
def main():
    query = "SELECT * FROM customers"
    table = sql.read(query)
    customer_records = pd.DataFrame(table)
    users = customer_records['username']
    print(users.value_counts())

if __name__ == "__main__":
    main()
```

Let's imagine that Janice sets out to write some code to generate a report.



```
def main():
    query = "SELECT * FROM customers"
    table = sql.read(query)
    customer_records = pd.DataFrame(table)
    users = customer_records['username']
    print(users.value_counts())

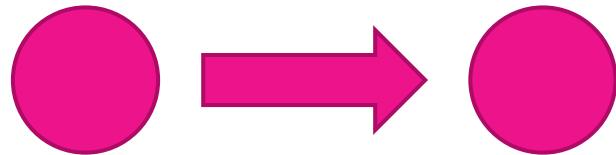
if __name__ == "__main__":
    main()
```



Hooray, Janice!
You did a thing!



CODE DEVELOPMENT TIMELINE

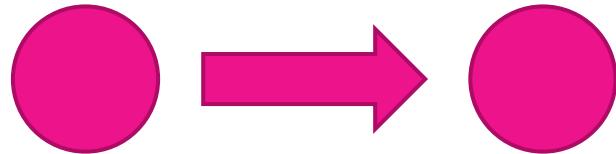


Janice writes
code.

Code put into
production. It's
ugly code.



CODE DEVELOPMENT TIMELINE



Janice writes
code.

Code put into
production. It's
ugly code.

Hey Janice.
Fix your ugly
code now.



```
def get_data(query):
    """
    Get's the data from the database
    and returns it in dataframe format
    """
    table_data = sql.read(query)
    return pd.DataFrame(table_data)

def print_user_report():
    query = "SELECT * FROM customers"
    customer_records = get_data(query)
    users = customer_records['username']
    print(customer_records.value_counts())

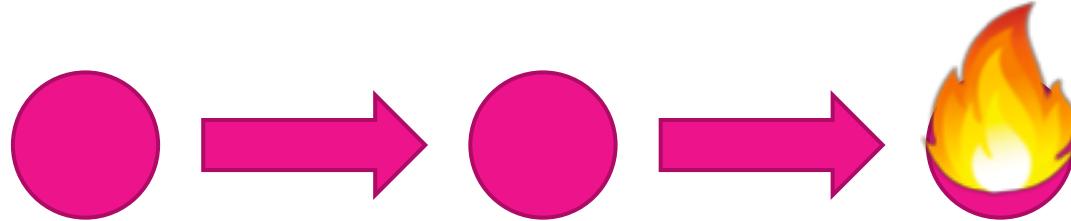
if __name__ == "__main__":
    print_user_report()
```

Janice now sets out to functionalize her code and make it reusable and clear.

Janice breaks things.



CODE DEVELOPMENT TIMELINE



Idea for what
the code
should do.

The code is ugly,
but it works.

Attempted
changes break
the code

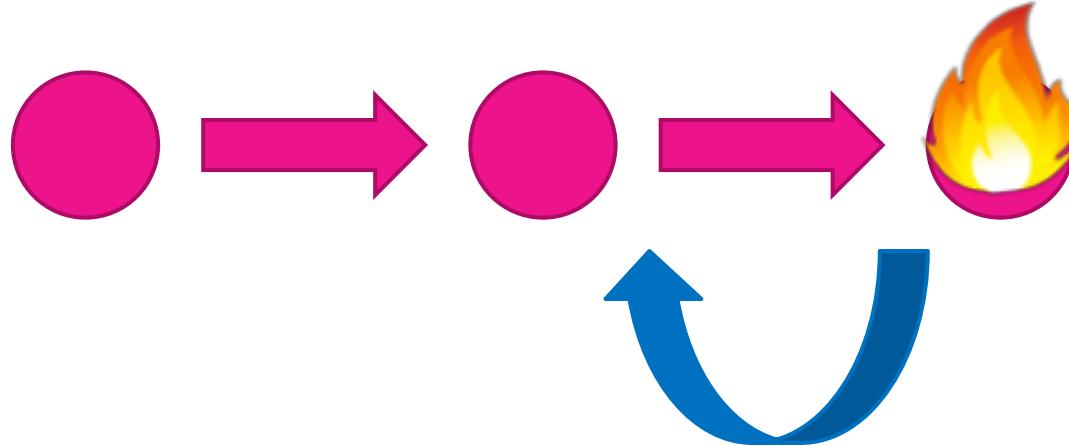


We broke our code, now what?

With the code broken, we have two options.

1. Figure out how to fix the code
2. Reset to the last time the code worked

CODE DEVELOPMENT TIMELINE



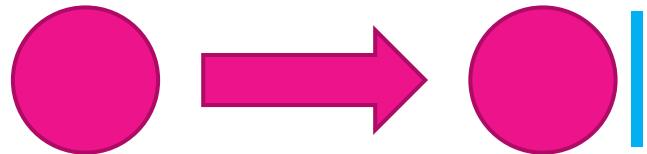
Doing this can be very time consuming if you haven't prepared for this eventuality. Let's use git to prepare ourselves.



What does Git do?

- Allows us to introduce a checkpointing system
- Keeps track of each file's evolution over time
- Keeps a history of those changes and allows us to return to a version of our code at any time
- Allows for multiple versions of the same code to exist simultaneously (we'll come back to this one)

CODE DEVELOPMENT TIMELINE



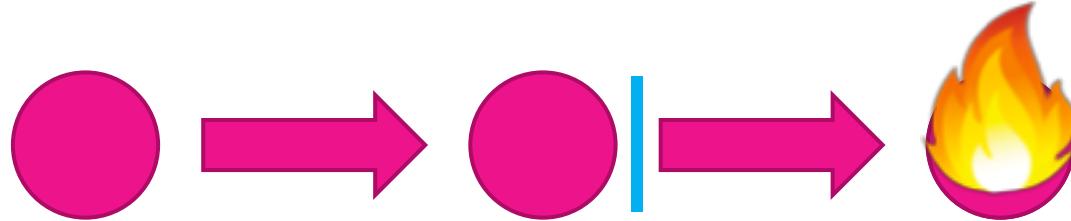
Checkpoint!



CODE DEVELOPMENT TIMELINE



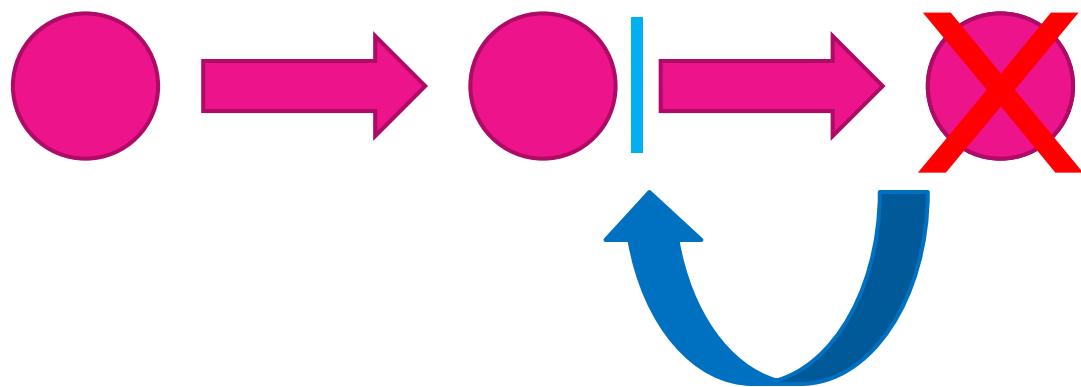
CODE DEVELOPMENT TIMELINE



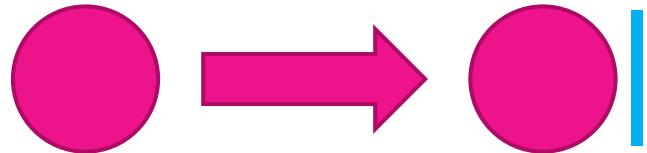
OOPS. REVERT.



CODE DEVELOPMENT TIMELINE



CODE DEVELOPMENT TIMELINE



Git Checkpointing



This checkpointing system is made up of things called “**commits**.” A **commit** is essentially a time stamp where git knows what all the files in the project look like. It remembers the code in each file so we can always get back to that version. Hence why git is called “version control.”

To actually make a **commit**, we need to understand the 3 phases of git.





Git is made up of three independent “phases” of code tracking. Let’s investigate each section.



Working Directory

The working directory area is our “normal” file system. This tracks the code as it exists on our computer right now. If we’ve made changes, then the working directory sees all of those changes. It doesn’t track any history. It just says, “this is how my code looks right now.”

Working
Directory

Staging Area

The staging area is how we prepare for a **commit**. Before making a commit, we'll tell git that we want certain files to be “staged” for the **commit** (in non-git terms, we're telling git that we want it to check this file to see if it's changed since the last checkpoint).



The repository is where all of our checkpoints live. Every time we make a **commit**, all of those changes are pushed into the repository. The repository is the holder of all knowledge about each version of a project. Until the code is **committed** to the repository, no checkpoints have been made.

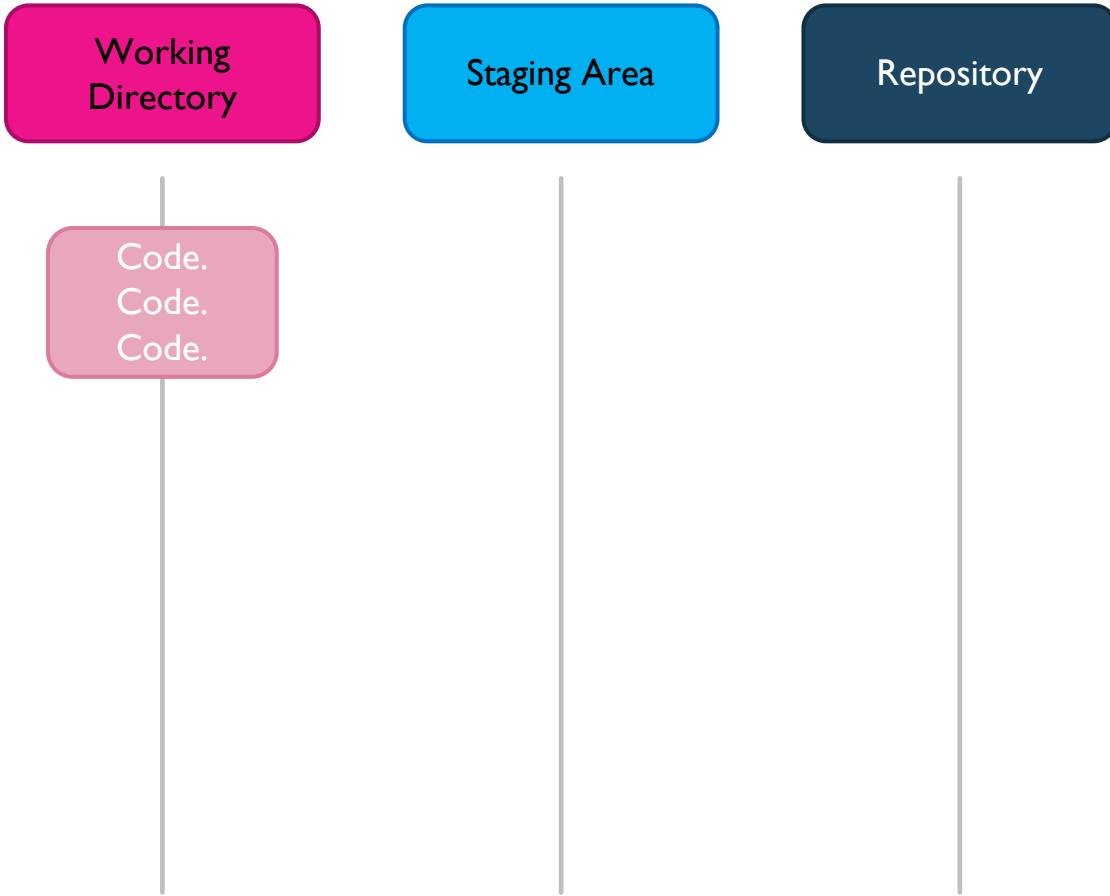
Repository





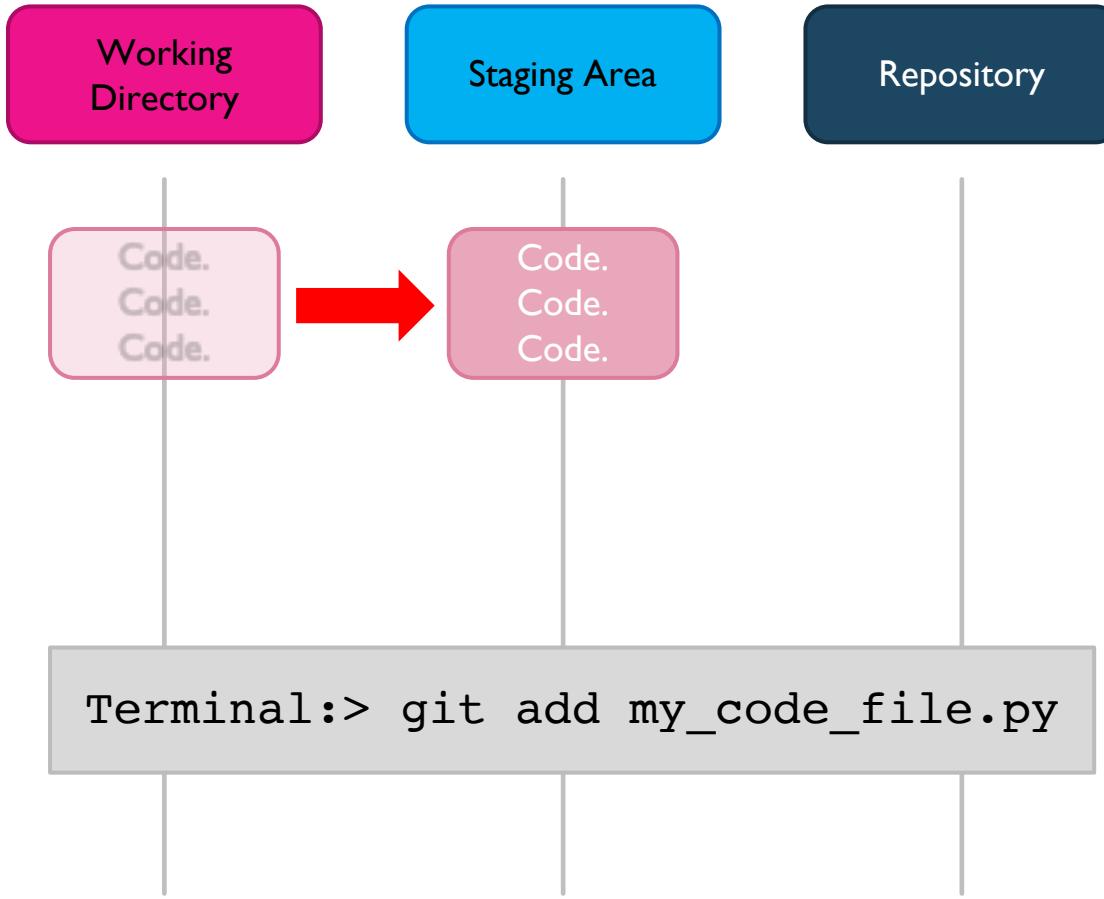
The flow of using Git



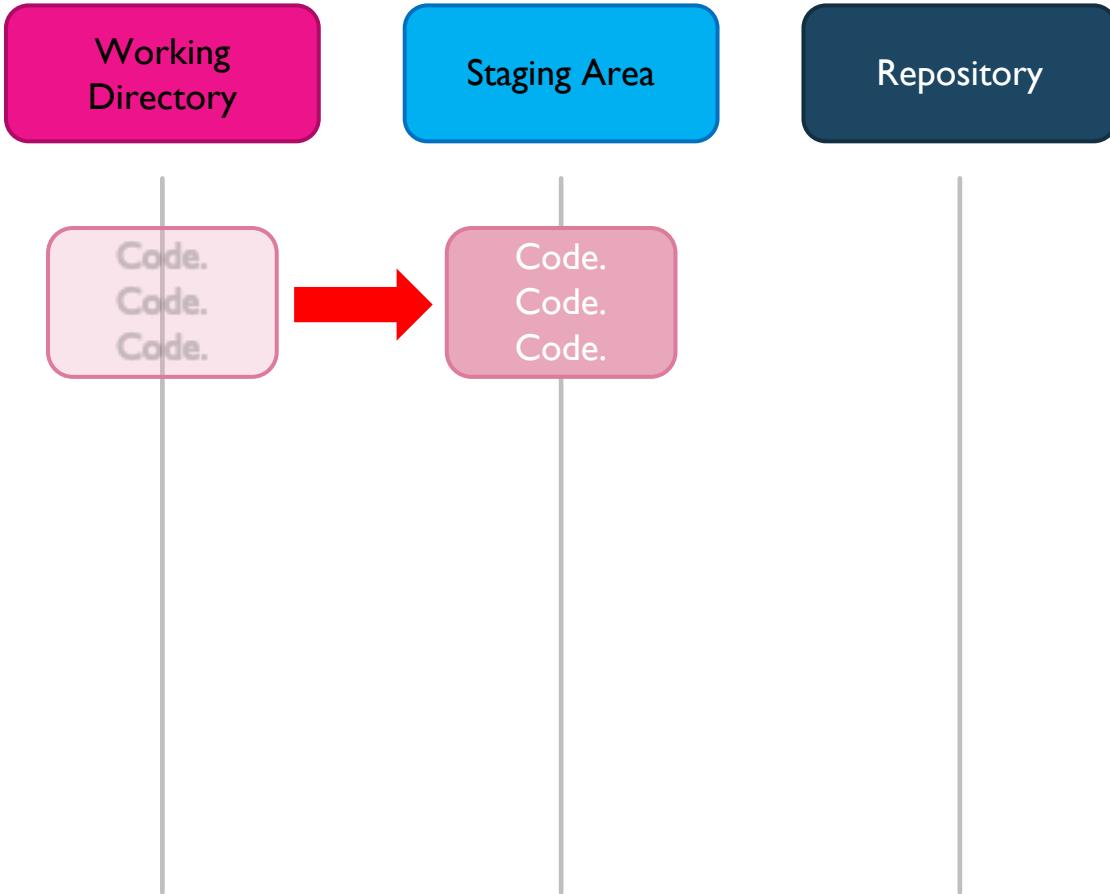


I work on my code in the working directory.

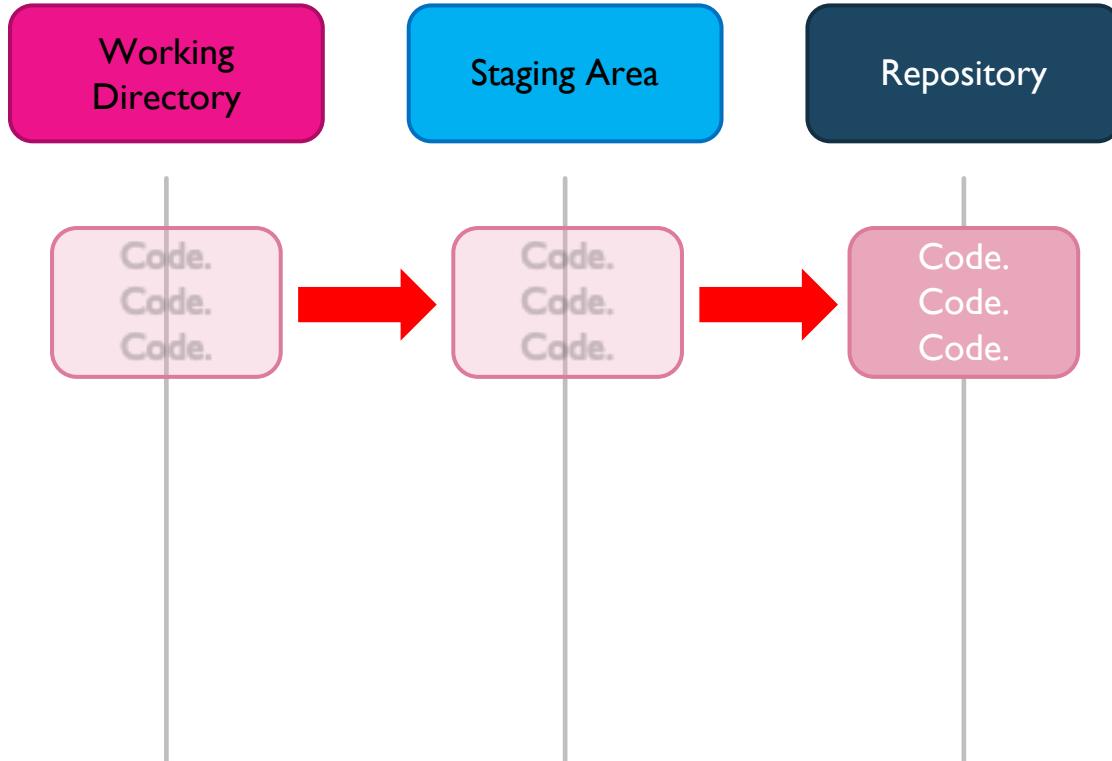




Now let's get it ready for a checkpoint. I need to tell git to 'add' it to the staging area.



At this point, I haven't made the checkpoint, I've just prepared for it. Now let's actually make the commit.



Now we've actually made a commit. All the things in the staging area are added to the repository and a special name is given to this version of the code to represent the checkpoint.

```
Terminal:> git commit -m "making my first checkpoint"
```





Working
Directory

Staging Area

Repository



This is called a “commit message” and it’s vitally important that you take it seriously. The special name that git gives to each checkpoint is just a series of characters like “d9rgy6431abql”. If you ever need to go back in time to a certain bit of code, you need to have a message that clearly identifies the commit’s purpose.

```
Terminal:> git commit -m "making my first checkpoint"
```



Commit Message Guide



Commit messages should:

- Clearly state what's changed since the last commit
- Tell us what bugs were fixed, and what new code was added

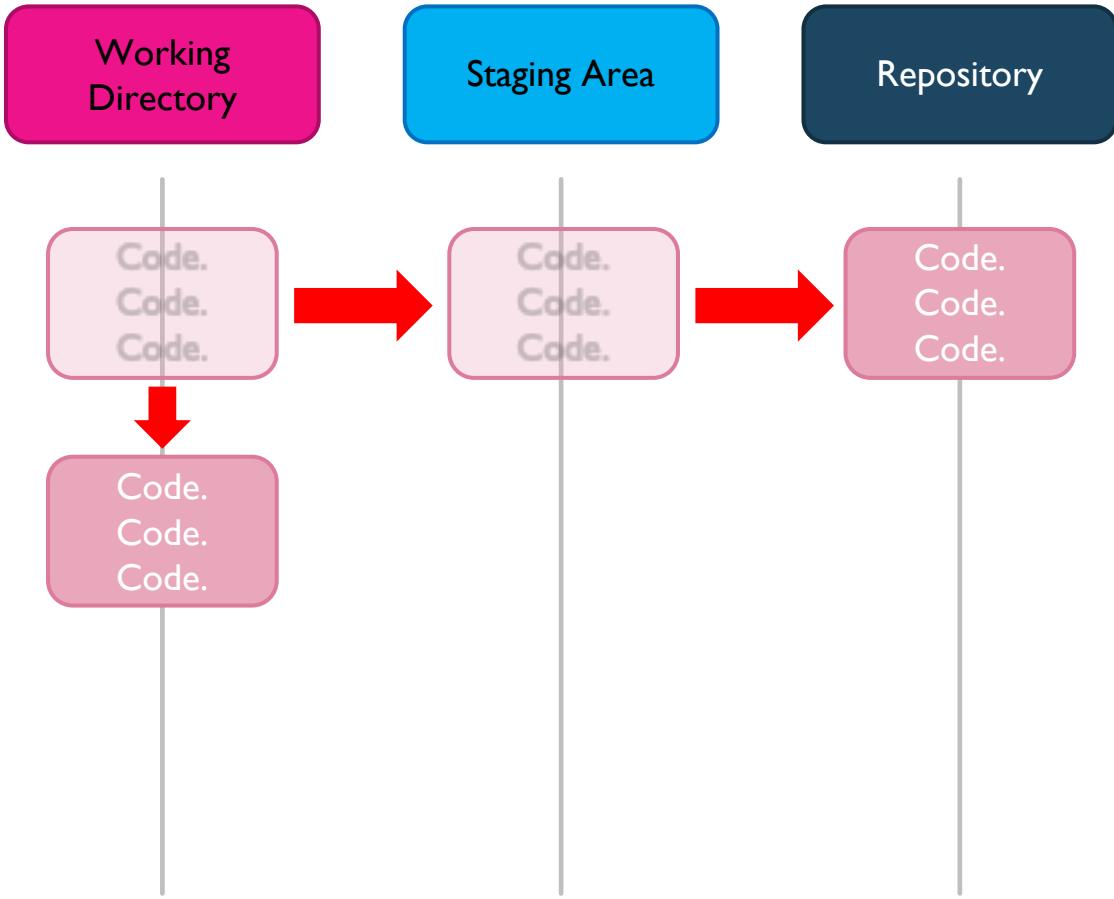
If we remove the `-m "message here"` part from the previous example, we will be taken to a text editor to make our commit messages instead and can make it more specific.



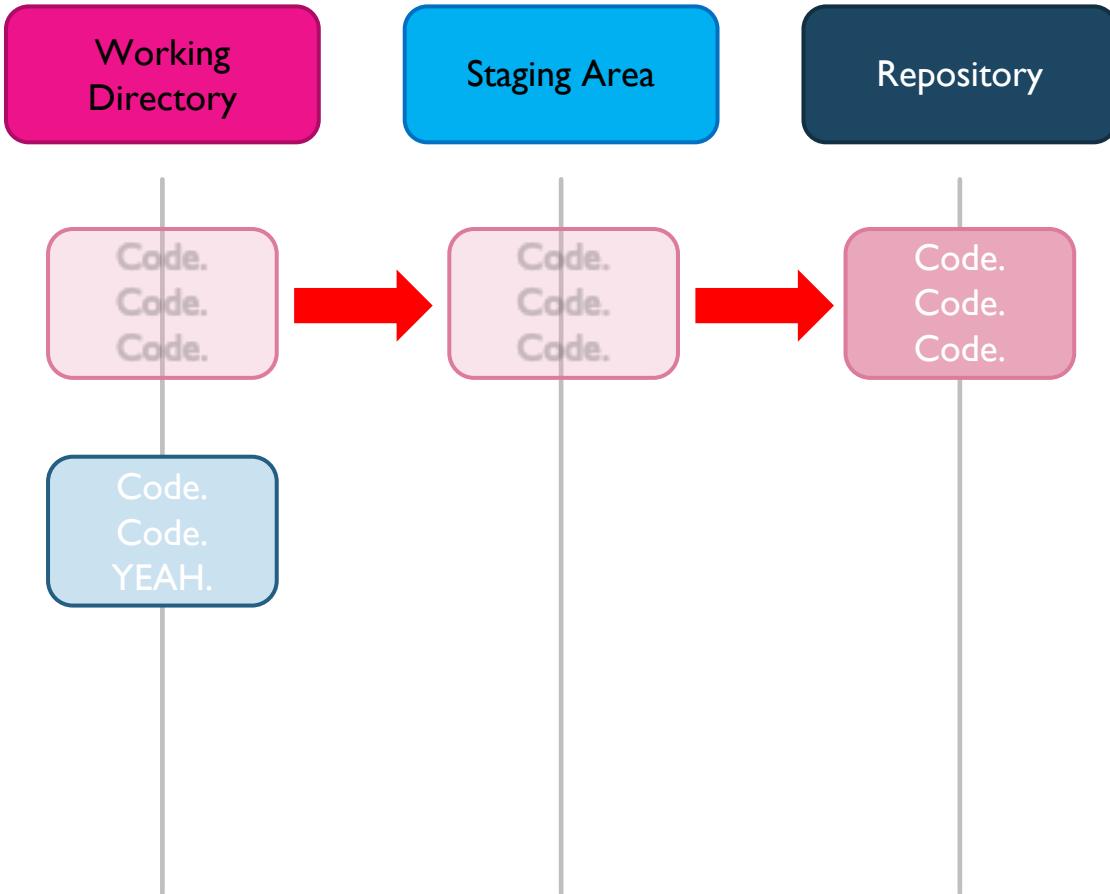
Commit Message Guide

People opine at length about good commit messages. Here are some great blogs on the topic:

- <https://hackernoon.com/what-makes-a-good-commit-message-995d23687ad>
- <https://chris.beams.io/posts/git-commit/>

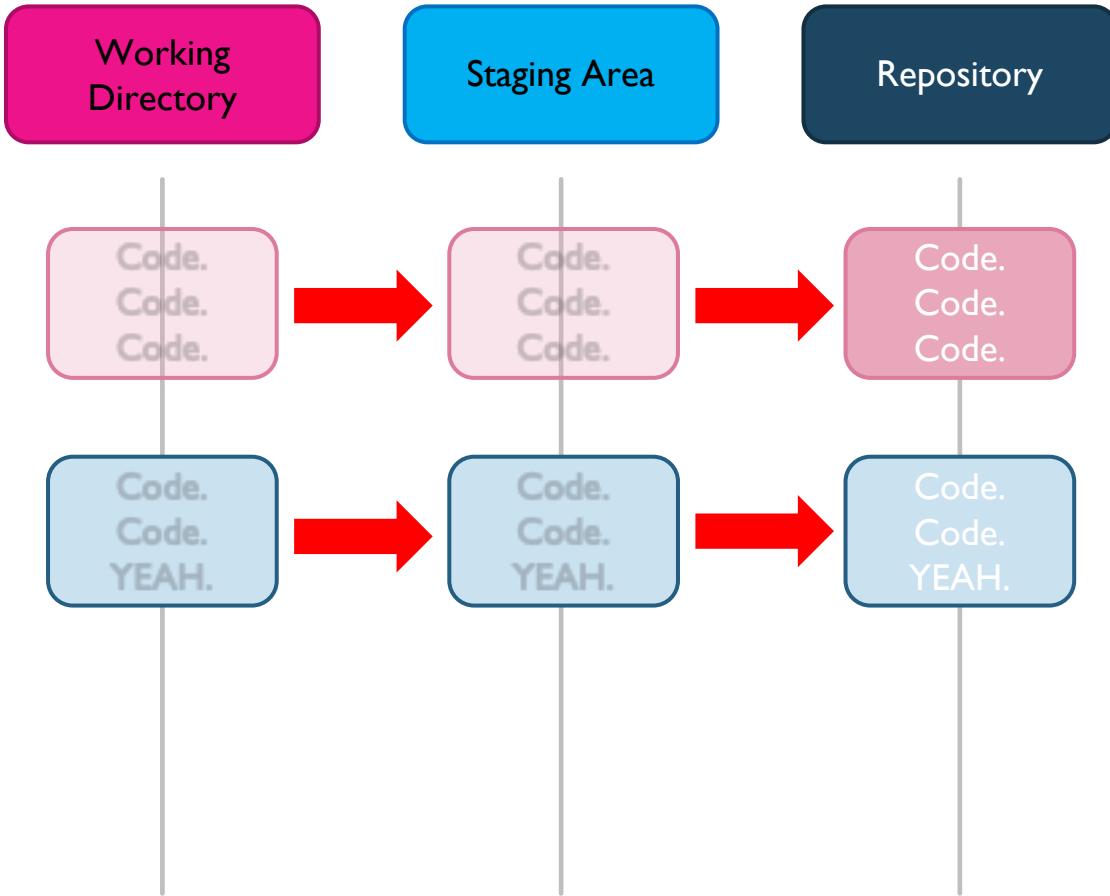


At this point our repository and our working directory match.



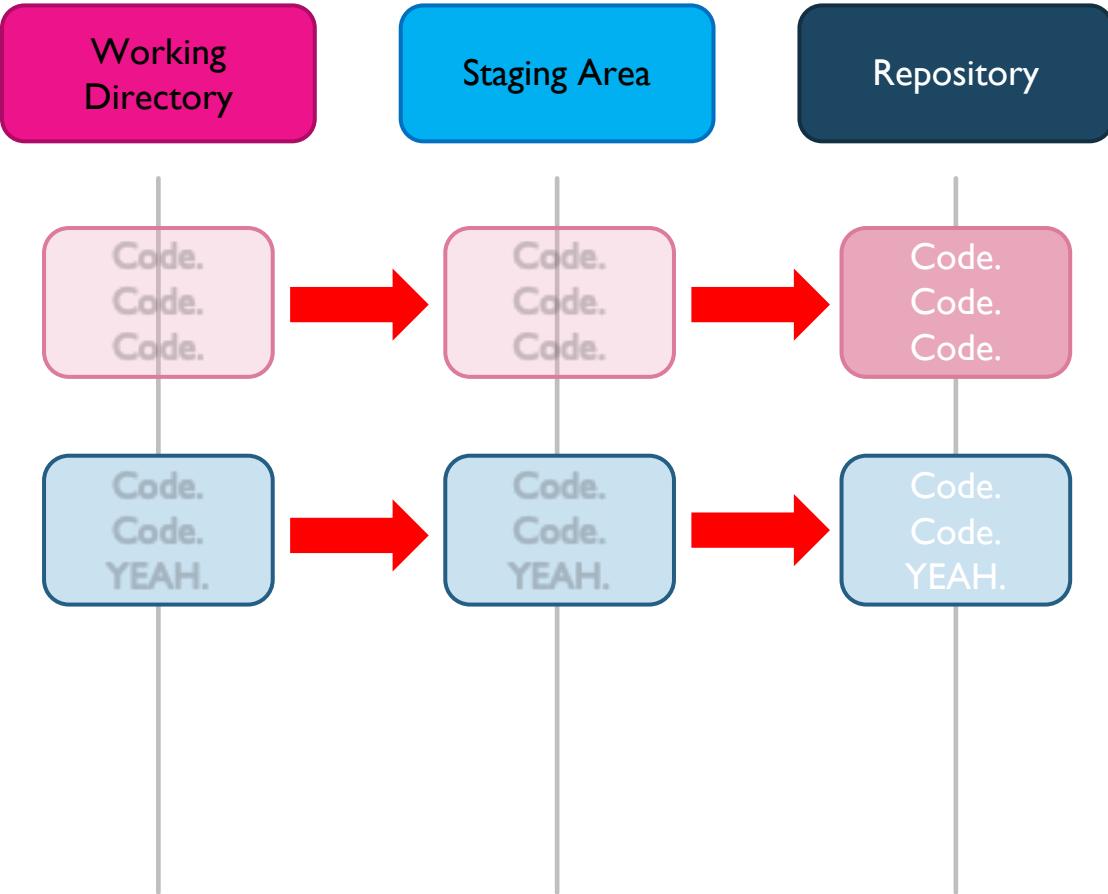
If we go through the process again, we can see that we'll add a new commit that also lives in the repository.





If we go through the process again, we can see that we'll add a new commit that also lives in the repository.





What if I want to revert to
the original commit?



Git Log

- There's a command underneath git called **git log**. It displays a list of all the previous commits and information about them.

```
zachariahmiller@metis-zachariahmiller~/Documents/PythonProjects$ git log
commit 8861cc02c62df5946d760a0618d349f6b4481f64 (HEAD -> master, origin/master)
Author: zwmiller <zachariahmiller.com>
Date:   Sun Sep 2 11:29:57 2018 -0500

    breakout network analysis to own repo

commit 128fc8819d8433971302cc94f5f3db7af08d4c51
Author: zwmiller <zachariahmiller.com>
Date:   Sun Sep 2 11:23:46 2018 -0500

    refactor code, fix sorting bug, redraw demo, add testing and test data

commit f27dbf83405f876643ffef270f11a1a21ab672a9
Author: zwmiller <zachariahmiller.com>
Date:   Fri Aug 31 14:30:39 2018 -0500

    add docstrings

commit 1dcaad09c98c3f8db2732db134b2e2a7a65ec681
Author: zwmiller <zachariahmiller.com>
Date:   Fri Aug 31 14:22:11 2018 -0500

    refactor plot making code
```

```
zachariahmiller@metis-zachariahmiller~/Documents/PythonProjects$ git log  
commit 8861cc02c62df5946d760a0618d349f6b4481f64 (HEAD -> master, origin/master)  
Author: zwmiller <z[REDACTED].com>  
Date: Sun Sep 2 11:29:57 2018 -0500  
  
    breakout network analysis to own repo
```

```
commit 128fc8819d8433971302cc94f5f3db7af08d4c51  
Author: zwmiller <z[REDACTED].com>  
Date: Sun Sep 2 11:23:46 2018 -0500  
  
    refactor code, fix sorting bug, redraw demo, add testing and test data
```

```
commit f27dbf83405f876643ffef270f11a1a21ab672a9  
Author: zwmiller <z[REDACTED].com>  
Date: Fri Aug 31 14:30:39 2018 -0500  
  
    add docstrings
```

```
commit 1dcaad09c98c3f8db2732db134b2e2a7a65ec681  
Author: zwmiller <z[REDACTED].com>  
Date: Fri Aug 31 14:22:11 2018 -0500  
  
    refactor plot making code
```

Commit IDs

```
zachariahmiller@metis-zachariahmiller~/Documents/PythonProjects$ git log  
commit 8861cc02c62df5946d760a0618d349f6b4481f64 (HEAD -> master, origin/master)  
Author: zwmiller <z[REDACTED].com>  
Date: Sun Sep 2 11:29:57 2018 -0500
```

breakout network analysis to own repo

```
commit 128fc8819d8433971302cc94f5f3db7af08d4c51  
Author: zwmiller <z[REDACTED].com>  
Date: Sun Sep 2 11:23:46 2018 -0500
```

refactor code, fix sorting bug, redraw demo, add testing and test data

```
commit f27dbf83405f876643ffef270f11a1a21ab672a9  
Author: zwmiller <z[REDACTED].com>  
Date: Fri Aug 31 14:30:39 2018 -0500
```

add docstrings

```
commit 1dcaad09c98c3f8db2732db134b2e2a7a65ec681  
Author: zwmiller <z[REDACTED].com>  
Date: Fri Aug 31 14:22:11 2018 -0500
```

refactor plot making code

Commit Messages

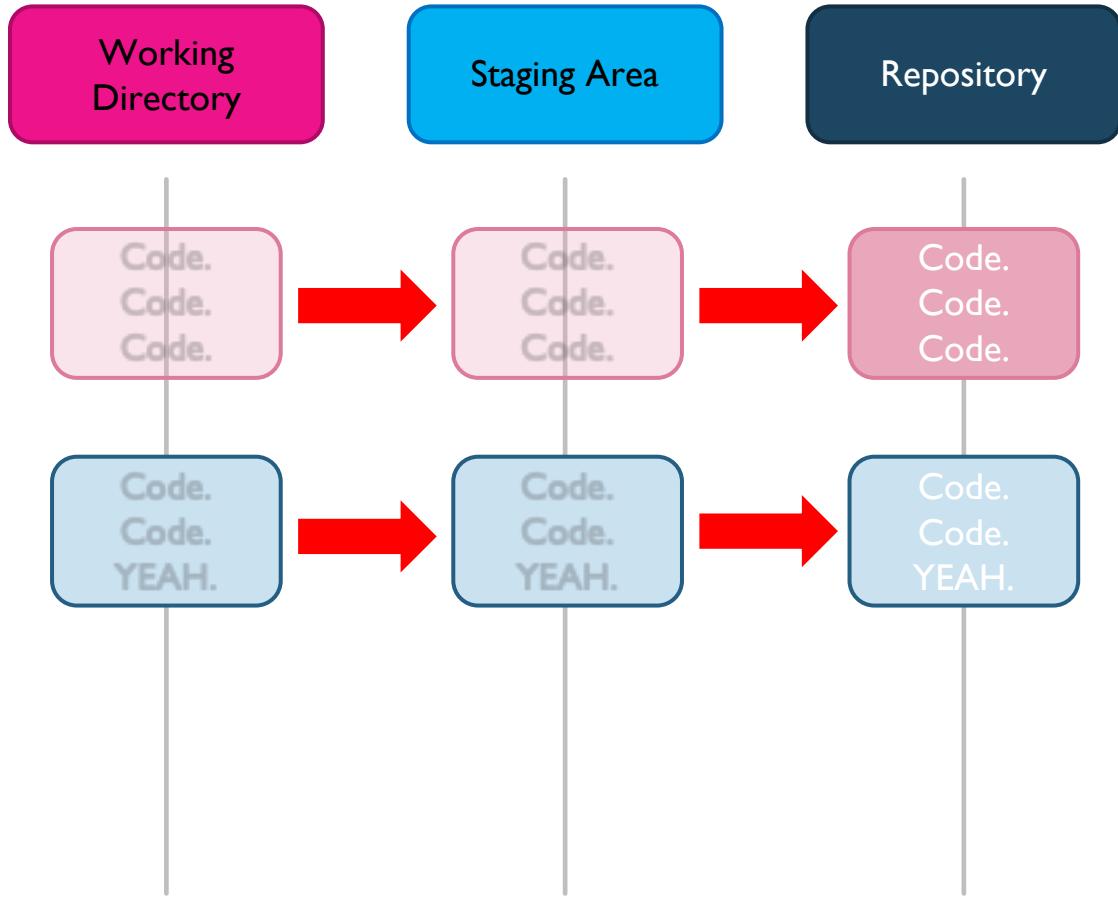
Git Reset



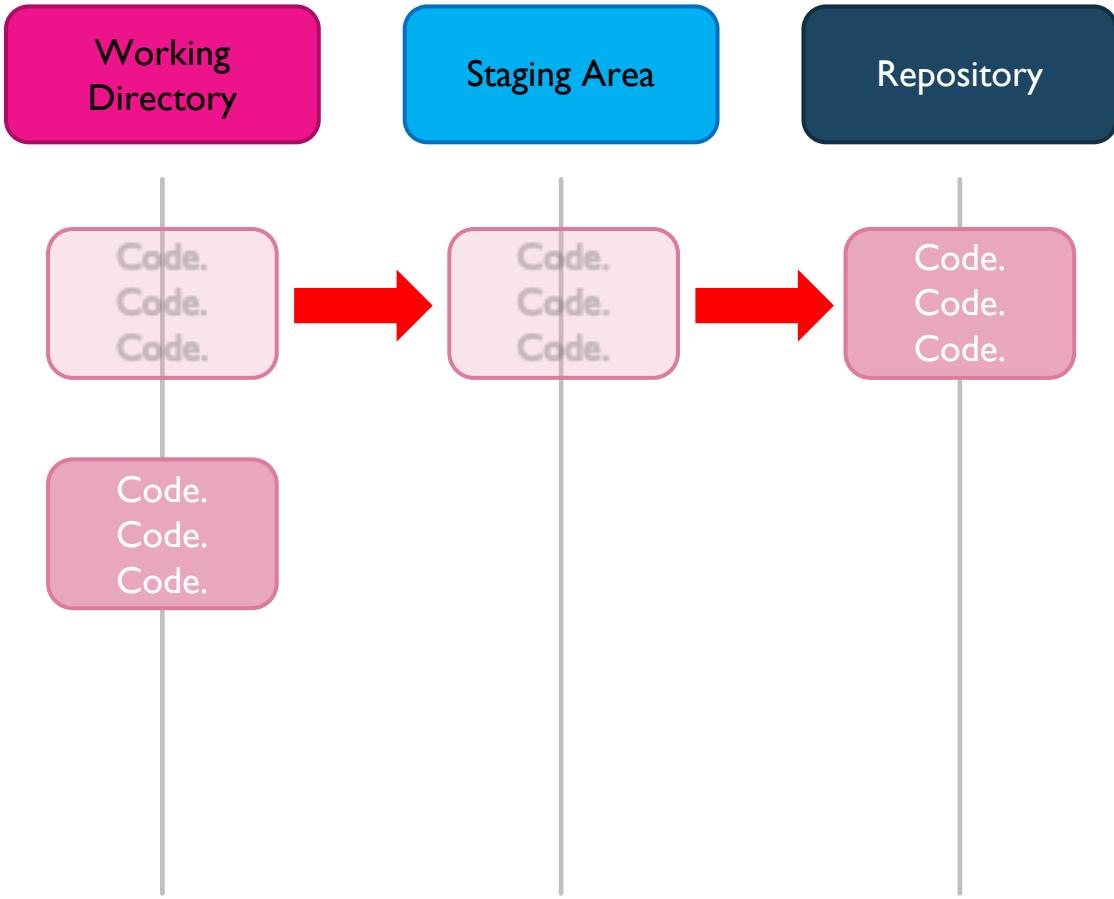
- So if I want to return to a previous state of code, I can just find the correct **commit id** then tell git to reset to that state.

```
Terminal:> git reset 128fc8819d8433971302cc94f5f3db7af08d4c51
```





Before reset.



Git: A quick review



- Git gives us the ability to track our code over time
- It gives us the history of our code, and the ability to reset to a previous status
- It provides a flexible interface for making checkpoints (called commits)



Exercise: Building your first repo



1.bash

python3.6 X bash X bash X3

zachariahmiller@metis-zachariahmiller~\$

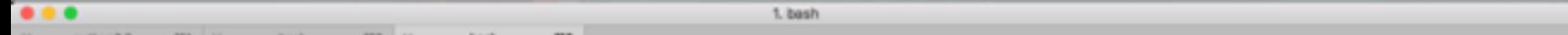
1.bash

```
zachariahmiller@metis-zachariahmiller~$ mkdir my_first_repo
```

1.bash

```
zachariahmiller@metis-zachariahmiller~$ mkdir my_first_repo  
zachariahmiller@metis-zachariahmiller~$ cd my_first_repo/
```

```
python3.6 301 bash 302 bash 303 1.bash
zachariahmiller@metis-zachariahmiller~$ mkdir my_first_repo
zachariahmiller@metis-zachariahmiller~$ cd my_first_repo/
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git init
```



```
zachariahmiller@metis-zachariahmiller~$ mkdir my_first_repo
zachariahmiller@metis-zachariahmiller~$ cd my_first_repo/
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git init
Initialized empty Git repository in /Users/zachariahmiller/my_first_repo/.git/
zachariahmiller@metis-zachariahmiller~/my_first_repo$ 
```

iTerm2 Shell Edit View Session Scripts Profiles Toolbar Window Help 100% Wed Sep 26 9:24 AM Zachariah Miller

python3.6 bash bash 1.bash

```
zachariahmiller@metis-zachariahmiller~$ mkdir my_first_repo
zachariahmiller@metis-zachariahmiller~$ cd my_first_repo/
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git init
Initialized empty Git repository in /Users/zachariahmiller/my_first_repo/.git/
zachariahmiller@metis-zachariahmiller~/my_first_repo$ ls -a
. .. .git
zachariahmiller@metis-zachariahmiller~/my_first_repos
```

iTerm2 Shell Edit View Session Scripts Profiles Toolbar Window Help 100% Wed Sep 26 9:24 AM Zachariah Miller

python3.6 X bash X bash X 1.bash

```
zachariahmiller@metis-zachariahmiller~$ mkdir my_first_repo
zachariahmiller@metis-zachariahmiller~$ cd my_first_repo/
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git init
Initialized empty Git repository in /Users/zachariahmiller/my_first_repo/.git/
zachariahmiller@metis-zachariahmiller~/my_first_repo$ ls -a
. .. .git
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git log
fatal: your current branch 'master' does not have any commits yet
zachariahmiller@metis-zachariahmiller~/my_first_repo$
```

```
zachariahmiller@metis-zachariahmiller~$ mkdir my_first_repo
zachariahmiller@metis-zachariahmiller~$ cd my_first_repo/
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git init
Initialized empty Git repository in /Users/zachariahmiller/my_first_repo/.git/
zachariahmiller@metis-zachariahmiller~/my_first_repo$ ls -a
.  ..  .git
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git log
fatal: your current branch 'master' does not have any commits yet
zachariahmiller@metis-zachariahmiller~/my_first_repo$ echo "YADA YADA" > my_file.txt
zachariahmiller@metis-zachariahmiller~/my_first_repo$
```

```
zachariahmiller@metis-zachariahmiller~$ mkdir my_first_repo
zachariahmiller@metis-zachariahmiller~$ cd my_first_repo/
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git init
Initialized empty Git repository in /Users/zachariahmiller/my_first_repo/.git/
zachariahmiller@metis-zachariahmiller~/my_first_repo$ ls -a
.  ..  .git
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git log
fatal: your current branch 'master' does not have any commits yet
zachariahmiller@metis-zachariahmiller~/my_first_repo$ echo "YADA YADA" > my_file.txt
zachariahmiller@metis-zachariahmiller~/my_first_repo$ ls -a
.  ..  .git  my_file.txt
zachariahmiller@metis-zachariahmiller~/my_first_repo$
```

```
python3.6      X  bash      X1  bash      X2  bash      X3  1.bash
zachariahmiller@metis-zachariahmiller~$ mkdir my_first_repo
zachariahmiller@metis-zachariahmiller~$ cd my_first_repo/
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git init
Initialized empty Git repository in /Users/zachariahmiller/my_first_repo/.git/
zachariahmiller@metis-zachariahmiller~/my_first_repo$ ls -a
.  ..  .git
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git log
fatal: your current branch 'master' does not have any commits yet
zachariahmiller@metis-zachariahmiller~/my_first_repo$ echo "YADA YADA" > my_file.txt
zachariahmiller@metis-zachariahmiller~/my_first_repo$ ls -a
.  ..  .git  my_file.txt
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git add my_file.txt ||
```

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbar Window Help 100% Wed Sep 26 9:25 AM Zachariah Miller Search
python3.6 301 bash 302 bash 303 bash 1.bash

zachariahmiller@metis-zachariahmiller~$ mkdir my_first_repo
zachariahmiller@metis-zachariahmiller~$ cd my_first_repo/
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git init
Initialized empty Git repository in /Users/zachariahmiller/my_first_repo/.git/
zachariahmiller@metis-zachariahmiller~/my_first_repo$ ls -a
. .. .git
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git log
fatal: your current branch 'master' does not have any commits yet
zachariahmiller@metis-zachariahmiller~/my_first_repo$ echo "YADA YADA" > my_file.txt
zachariahmiller@metis-zachariahmiller~/my_first_repo$ ls -a
. .. .git my_file.txt
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git add my_file.txt
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git status
On branch master

No commits yet

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

    new file:  my_file.txt

zachariahmiller@metis-zachariahmiller~/my_first_repo$
```

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbar Window Help M1 100% Wed Sep 26 9:25 AM Zachariah Miller Search
```

python3.6 X bash X2 X bash X3 X 1.bash

```
zachariahmiller@metis-zachariahmiller~$ mkdir my_first_repo
zachariahmiller@metis-zachariahmiller~$ cd my_first_repo/
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git init
Initialized empty Git repository in /Users/zachariahmiller/my_first_repo/.git/
zachariahmiller@metis-zachariahmiller~/my_first_repo$ ls -a
. .. .git
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git log
fatal: your current branch 'master' does not have any commits yet
zachariahmiller@metis-zachariahmiller~/my_first_repo$ echo "YADA YADA" > my_file.txt
zachariahmiller@metis-zachariahmiller~/my_first_repo$ ls -a
. .. .git my_file.txt
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git add my_file.txt
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git status
On branch master

No commits yet

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

    new file:  my_file.txt

zachariahmiller@metis-zachariahmiller~/my_first_repo$ git log
fatal: your current branch 'master' does not have any commits yet
zachariahmiller@metis-zachariahmiller~/my_first_repo$
```

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbar Window Help 100% Wed Sep 26 9:26 AM Zachariah Miller Search
python3.6 301 bash 302 bash 303 bash 1.bash

zachariahmiller@metis-zachariahmiller~$ mkdir my_first_repo
zachariahmiller@metis-zachariahmiller~$ cd my_first_repo/
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git init
Initialized empty Git repository in /Users/zachariahmiller/my_first_repo/.git/
zachariahmiller@metis-zachariahmiller~/my_first_repo$ ls -a
. .. .git
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git log
fatal: your current branch 'master' does not have any commits yet
zachariahmiller@metis-zachariahmiller~/my_first_repo$ echo "YADA YADA" > my_file.txt
zachariahmiller@metis-zachariahmiller~/my_first_repo$ ls -a
. .. .git my_file.txt
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git add my_file.txt
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git status
On branch master

No commits yet

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

    new file:  my_file.txt

zachariahmiller@metis-zachariahmiller~/my_first_repo$ git log
fatal: your current branch 'master' does not have any commits yet
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git commit -m "added my_file.txt to the repo"
```

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbar Window Help M1 100% Wed Sep 26 9:26 AM Zachariah Miller Search
```

python3.6 bash bash 1.bash

```
zachariahmiller@metis-zachariahmiller~$ mkdir my_first_repo
zachariahmiller@metis-zachariahmiller~$ cd my_first_repo/
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git init
Initialized empty Git repository in /Users/zachariahmiller/my_first_repo/.git/
zachariahmiller@metis-zachariahmiller~/my_first_repo$ ls -a
. .. .git
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git log
fatal: your current branch 'master' does not have any commits yet
zachariahmiller@metis-zachariahmiller~/my_first_repo$ echo "YADA YADA" > my_file.txt
zachariahmiller@metis-zachariahmiller~/my_first_repo$ ls -a
. .. .git my_file.txt
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git add my_file.txt
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git status
On branch master

No commits yet

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

    new file:  my_file.txt

zachariahmiller@metis-zachariahmiller~/my_first_repo$ git log
fatal: your current branch 'master' does not have any commits yet
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git commit -m "added my_file.txt to the repo"
[master (root-commit) 2c90153] added my_file.txt to the repo
 1 file changed, 1 insertion(+)
 create mode 100644 my_file.txt
zachariahmiller@metis-zachariahmiller~/my_first_repo$
```

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbar Window Help 100% Wed Sep 26 9:26 AM Zachariah Miller Search
python3.6  bash  bash  bash  1.bash

zachariahmiller@metis-zachariahmiller~$ mkdir my_first_repo
zachariahmiller@metis-zachariahmiller~$ cd my_first_repo/
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git init
Initialized empty Git repository in /Users/zachariahmiller/my_first_repo/.git/
zachariahmiller@metis-zachariahmiller~/my_first_repo$ ls -a
.  ..  .git
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git log
fatal: your current branch 'master' does not have any commits yet
zachariahmiller@metis-zachariahmiller~/my_first_repo$ echo "YADA YADA" > my_file.txt
zachariahmiller@metis-zachariahmiller~/my_first_repo$ ls -a
.  ..  .git  my_file.txt
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git add my_file.txt
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git status
On branch master

No commits yet

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

    new file:  my_file.txt

zachariahmiller@metis-zachariahmiller~/my_first_repo$ git log
fatal: your current branch 'master' does not have any commits yet
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git commit -m "added my_file.txt to the repo"
[master (root-commit) 2c90153] added my_file.txt to the repo
 1 file changed, 1 insertion(+)
 create mode 100644 my_file.txt
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git log
commit 2c9015399839d375254f22a68db539ff3c05b485 (HEAD -> master)
Author: zmiller <zagliamir@gmail.com>
Date:   Wed Sep 26 09:26:04 2018 -0500

    added my_file.txt to the repo
zachariahmiller@metis-zachariahmiller~/my_first_repo$
```

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbar Window Help M1 100% Wed Sep 26 9:26 AM Zachariah Miller Search
```

python3.6 bash bash 1.bash

```
zachariahmiller@metis-zachariahmiller~$ mkdir my_first_repo
zachariahmiller@metis-zachariahmiller~$ cd my_first_repo/
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git init
Initialized empty Git repository in /Users/zachariahmiller/my_first_repo/.git/
zachariahmiller@metis-zachariahmiller~/my_first_repo$ ls -a
. .. .git
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git log
fatal: your current branch 'master' does not have any commits yet
zachariahmiller@metis-zachariahmiller~/my_first_repo$ echo "YADA YADA" > my_file.txt
zachariahmiller@metis-zachariahmiller~/my_first_repo$ ls -a
. .. .git my_file.txt
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git add my_file.txt
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git status
On branch master

No commits yet

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

    new file:  my_file.txt

zachariahmiller@metis-zachariahmiller~/my_first_repo$ git log
fatal: your current branch 'master' does not have any commits yet
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git commit -m "added my_file.txt to the repo"
[master (root-commit) 2c90153] added my_file.txt to the repo
1 file changed, 1 insertion(+)
 create mode 100644 my_file.txt
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git log
commit 2c9015399839d375254f22a68db539ff3c05b485 (HEAD -> master)
Author: zmiller <zagliamir@gmail.com>
Date:   Wed Sep 26 09:26:04 2018 -0500

    added my_file.txt to the repo
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git status
On branch master
nothing to commit, working tree clean
zachariahmiller@metis-zachariahmiller~/my_first_repo$
```

1.bash

```
python3.6      X1      bash      X2      bash      X3      1.bash
zachariahmiller@metis-zachariahmiller~/my_first_repo$ echo "HOORAY" >> my_file.txt
zachariahmiller@metis-zachariahmiller~/my_first_repo$ cat my_file.txt
YADA YADA
HOORAY
zachariahmiller@metis-zachariahmiller~/my_first_repos ]]
```

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbar Window Help 100% Wed Sep 26 9:27 AM Zachariah Miller Search
python3.6 301 bash 302 bash 303 1.bash
zachariahmiller@metis-zachariahmiller~/my_first_repo$ echo "HOORAY" >> my_file.txt
zachariahmiller@metis-zachariahmiller~/my_first_repo$ cat my_file.txt
YADA YADA
HOORAY
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git add my_file.txt
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   my_file.txt

zachariahmiller@metis-zachariahmiller~/my_first_repo$ git commit -m "update my_file.txt with HOORAY"
[master 4f72269] update my_file.txt with HOORAY
 1 file changed, 1 insertion(+)
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git log
commit 4f722698c22bf3404fda07943688f66471 (HEAD -> master)
Author: zwmiller <zaglamir@gmail.com>
Date:   Wed Sep 26 09:27:49 2018 -0500

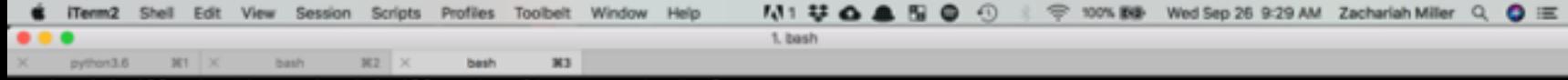
  update my_file.txt with HOORAY

commit 2c9015399839d375254f22a68db539ff3c05b485
Author: zwmiller <zaglamir@gmail.com>
Date:   Wed Sep 26 09:26:04 2018 -0500

  added my_file.txt to the repo
zachariahmiller@metis-zachariahmiller~/my_first_repo$
```

Reverting to a previous point



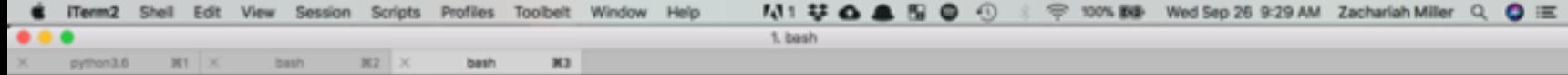


```
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git log  
commit 4f722698c22bf3404fda8a9dba07943688f66471 (HEAD -> master)  
Author: zwmiller <zaglamir@gmail.com>  
Date:   Wed Sep 26 09:27:49 2018 -0500
```

```
    update my_file.txt with MOORAY
```

```
commit 2c9015399839d375254f22a68db539ff3c05b485  
Author: zwmiller <zaglamir@gmail.com>  
Date:   Wed Sep 26 09:26:04 2018 -0500
```

```
    added my_file.txt to the repo
```

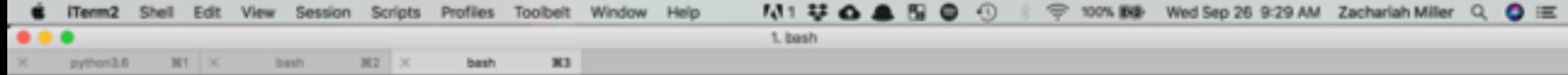


```
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git log
commit 4f722698c22bf3404fda8a9dba07943688f66471 (HEAD -> master)
Author: zwmiller <zaglamir@gmail.com>
Date:   Wed Sep 26 09:27:49 2018 -0500

    update my_file.txt with HOORAY

commit 2c9015399839d375254f22a68db539ff3c05b485
Author: zwmiller <zaglamir@gmail.com>
Date:   Wed Sep 26 09:26:04 2018 -0500

    added my_file.txt to the repo
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git reset 2c9015399839d37
Unstaged changes after reset:
M      my_file.txt
zachariahmiller@metis-zachariahmiller~/my_first_repo$ cat my_file.txt
YADA YADA
HOORAY
```



```
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git log
commit 4f722698c22bf3404fda8a9dba07943688f66471 (HEAD -> master)
Author: zwmiller <zaglamir@gmail.com>
Date:   Wed Sep 26 09:27:49 2018 -0500

    update my_file.txt with HOORAY

commit 2c9015399839d375254f22a68db539ff3c05b485
Author: zwmiller <zaglamir@gmail.com>
Date:   Wed Sep 26 09:26:04 2018 -0500

    added my_file.txt to the repo
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git reset 2c9015399839d37
Unstaged changes after reset:
M      my_file.txt
zachariahmiller@metis-zachariahmiller~/my_first_repo$ cat my_file.txt
YADA YADA
HOORAY
```

```
iTerm2 Shell Edit View Session Scripts Profiles Toolbar Window Help 100% Wed Sep 26 9:29 AM Zachariah Miller Search
python3.6 301 bash 302 bash 303 1.bash
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git log
commit 4f722698c22bf3404fda8a9dba07943688f66471 (HEAD -> master)
Author: zwmiller <zaglamir@gmail.com>
Date:   Wed Sep 26 09:27:49 2018 -0500

    update my_file.txt with HOORAY

commit 2c9015399839d375254f22a68db539ff3c05b485
Author: zwmiller <zaglamir@gmail.com>
Date:   Wed Sep 26 09:26:04 2018 -0500

    added my_file.txt to the repo
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git reset 2c9015399839d37
Unstaged changes after reset:
M      my_file.txt
zachariahmiller@metis-zachariahmiller~/my_first_repo$ cat my_file.txt
YADA YADA
HOORAY
zachariahmiller@metis-zachariahmiller~/my_first_repo$ git reset --hard 2c9015399839d37
HEAD is now at 2c90153 added my_file.txt to the repo
zachariahmiller@metis-zachariahmiller~/my_first_repo$ cat my_file.txt
YADA YADA
zachariahmiller@metis-zachariahmiller~/my_first_repo$
```

GitHub: Managing Coding Teams



GitHub



GitHub looks at what git does and asks the question, “what if I want lots of people to be sharing code?”

- It adds an online component of code sharing
- It adds a new phase to our “3 phases of Git”
- It allows the repo to live somewhere other than your computer, so your backups are backed up.





Search or jump to...

Pull requests Issues Marketplace Explore



scikit-learn / scikit-learn

Watch 2,194

Star 30,731

Fork 15,156

Code

Issues 1,092

Pull requests 641

Projects 5

Wiki

Insights

scikit-learn: machine learning in Python <http://scikit-learn.org>

machine-learning python statistics data-science data-analysis

23,273 commits

18 branches

89 releases

1,164 contributors

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

egriseli Dedicate the release to Raghav

Latest commit 86931ad 4 hours ago

ci/circleci	[MRG] Fix FutureWarnings in logistic regression examples (#12114)	2 days ago
benchmarks	FIX: enforce backward compatibility of decision function in Iforest (#...	2 months ago
build_tools	[MRG] MNT Re-enable PyPy CI (#12039)	8 days ago
doc	Dedicate the release to Raghav	4 hours ago
examples	MNT Unused import in plot_gpr_co2.py	7 hours ago
sklearn	[MRG] Crash when using SGDClassifier with early stopping in a paralle...	5 hours ago
.codecov.yml	Turn off codecov comments (#10146)	11 months ago
.coveragerc	coverall added	5 years ago
.gitattributes	MAINT remove .c files from .gitattributes	2 years ago
.gitignore	MAINT Complete 0.20 deprecations (#9570)	3 months ago
.landscape.yml	make landscape.io much more useful	4 years ago
.mailmap	Fix mailmap format (#9620)	a year ago
.travis.yml	MNT Only checks warnings on latest dependencies versions in CI (#12048)	9 days ago
CONTRIBUTING.md	DOC Link to dev doc in CONTRIBUTING.md	6 months ago



Search or jump to...

Pull requests Issues Marketplace Explore



scikit-learn / scikit-learn

Watch 2,194

Star 30,731

Fork 15,156

Code

Issues 1,092

Pull requests 641

Projects 5

Wiki

Insights

scikit-learn: machine learning in Python <http://scikit-learn.org>

machine-learning python statistics data-science data-analysis

23,273 commits

18 branches

89 releases

1,164 contributors

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

e grisel Dedicate the release to Raghav

Latest commit 86931ad 4 hours ago

.circleci [MRG] Fix FutureWarnings in logistic regression examples (#12114) 2 days ago

benchmarks FIX: enforce backward compatibility of decision function in iforest (#... 2 months ago

build_tools [MRG] MNT Re-enable PyPy CI (#12039) 8 days ago

doc Dedicate the release to Raghav 4 hours ago

examples MNT Unused import in plot_gpr_co2.py 7 hours ago

sklearn [MRG] Crash when using SGDClassifier with early stopping in a paralle... 5 hours ago

.codecov.yml Turn off codecov comments (#10146) 11 months ago

.coveragerc coverall added 5 years ago

.gitattributes MAINT remove .c files from .gitattributes 2 years ago

.gitignore MAINT Complete 0.20 deprecations (#9570) 3 months ago

.landscape.yml make landscape.io much more useful 4 years ago

.mailmap Fix mailmap format (#9620) a year ago

.travis.yml MNT Only checks warnings on latest dependencies versions in CI (#12048) 9 days ago

CONTRIBUTING.md DOC Link to dev doc in CONTRIBUTING.md 6 months ago

1000+ Coders





Search or jump to...

Pull requests Issues Marketplace Explore



scikit-learn / scikit-learn

Watch 2,194

Star 30,731

Fork 15,156

Code

Issues 1,092

Pull requests 641

Projects 5

Wiki

Insights

scikit-learn: machine learning in Python <http://scikit-learn.org>

machine-learning python statistics data-science data-analysis

23,273 commits

18 branches

89 releases

1,164 contributors

23,000+ Commits

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

 egriseli	Dedicate the release to Raghav	Latest commit 86931ad 4 hours ago
 .circleci	[MRG] Fix FutureWarnings in logistic regression examples (#12114)	2 days ago
 benchmarks	FIX: enforce backward compatibility of decision function in Iforest (#...	2 months ago
 build_tools	[MRG] MNT Re-enable PyPy CI (#12039)	8 days ago
 doc	Dedicate the release to Raghav	4 hours ago
 examples	MNT Unused import in plot_gpr_co2.py	7 hours ago
 sklearn	[MRG] Crash when using SGDClassifier with early stopping in a paralle...	5 hours ago
 .codecov.yml	Turn off codecov comments (#10146)	11 months ago
 .coveragerc	coverall added	5 years ago
 .gitattributes	MAINT remove .c files from .gitattributes	2 years ago
 .gitignore	MAINT Complete 0.20 deprecations (#9570)	3 months ago
 .landscape.yml	make landscape.io much more useful	4 years ago
 .mailmap	Fix mailmap format (#9620)	a year ago
 .travis.yml	MNT Only checks warnings on latest dependencies versions in CI (#12048)	9 days ago
 CONTRIBUTING.md	DOC Link to dev doc in CONTRIBUTING.md	6 months ago



Search or jump to...

Pull requests Issues Marketplace Explore



scikit-learn / scikit-learn

Watch 2,194

Star 30,731

Fork 15,156

Code

Issues 1,092

Pull requests 641

Projects 5

Wiki

Insights

scikit-learn: machine learning in Python <http://scikit-learn.org>

machine-learning python statistics data-science data-analysis

23,273 commits

18 branches

89 releases

1,164 contributors

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

egriseli Dedicate the release to Raghav

Latest commit 86931ad 4 hours ago

.circleci

[MRG] Fix FutureWarnings in logistic regression examples (#12114)

2 days ago

benchmarks

FIX: enforce backward compatibility of decision function in Iforest (#...

2 months ago

build_tools

[MRG] MNT Re-enable PyPy CI (#12039)

8 days ago

doc

Dedicate the release to Raghav

4 hours ago

examples

MNT Unused import in plot_gpr_co2.py

7 hours ago

sklearn

[MRG] Crash when using SGDClassifier with early stopping in a paralle...

5 hours ago

.codecov.yml

Turn off codecov comments (#10146)

11 months ago

.coveragerc

coverall added

5 years ago

.gitattributes

MAINT remove .c files from .gitattributes

2 years ago

.gitignore

MAINT Complete 0.20 deprecations (#9570)

3 months ago

.landscape.yml

make landscape.io much more useful

4 years ago

.mailmap

Fix mailmap format (#9620)

a year ago

.travis.yml

MNT Only checks warnings on latest dependencies versions in CI (#12048)

9 days ago

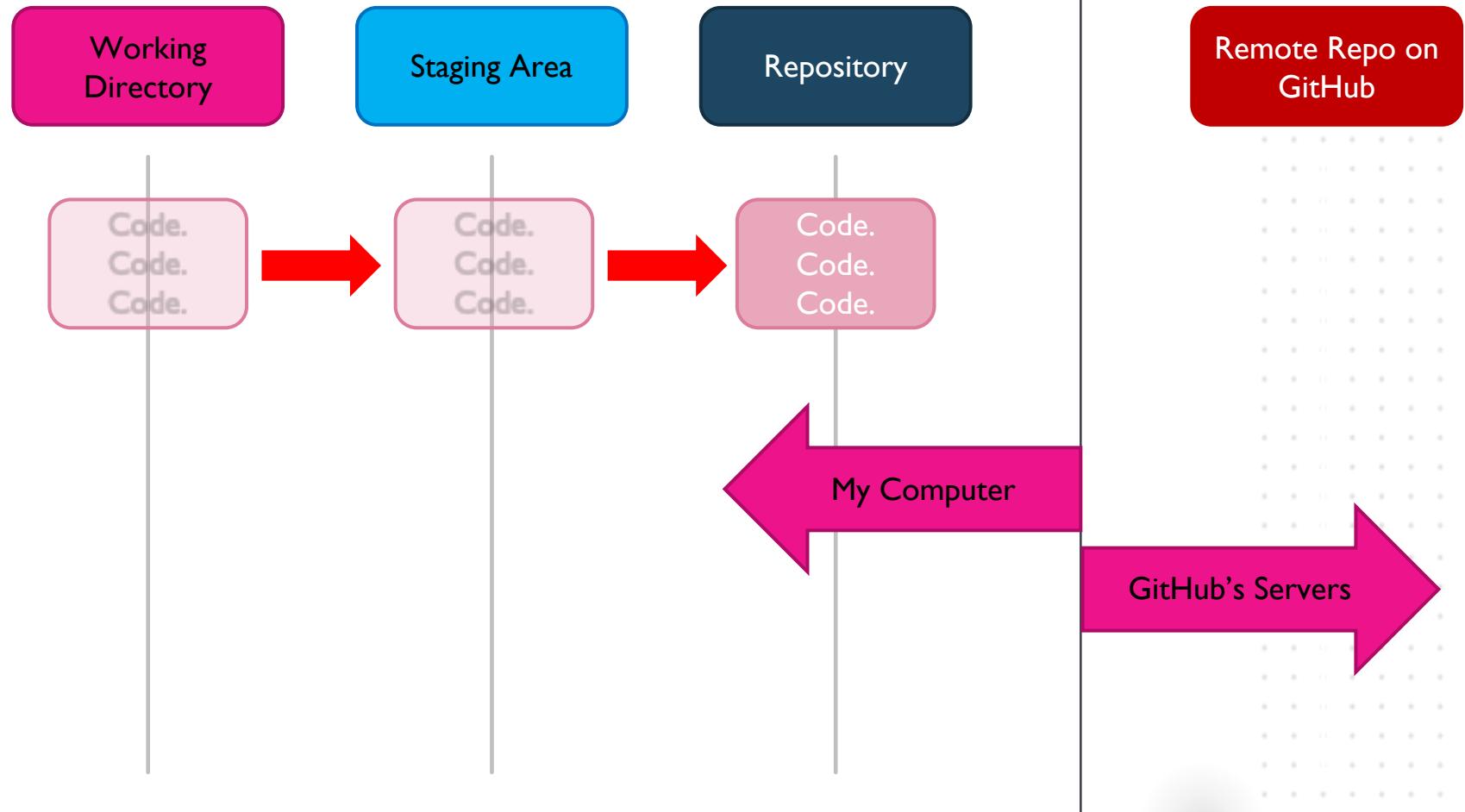
CONTRIBUTING.md

DOC Link to dev doc in CONTRIBUTING.md

6 months ago

All of Scikit Learn



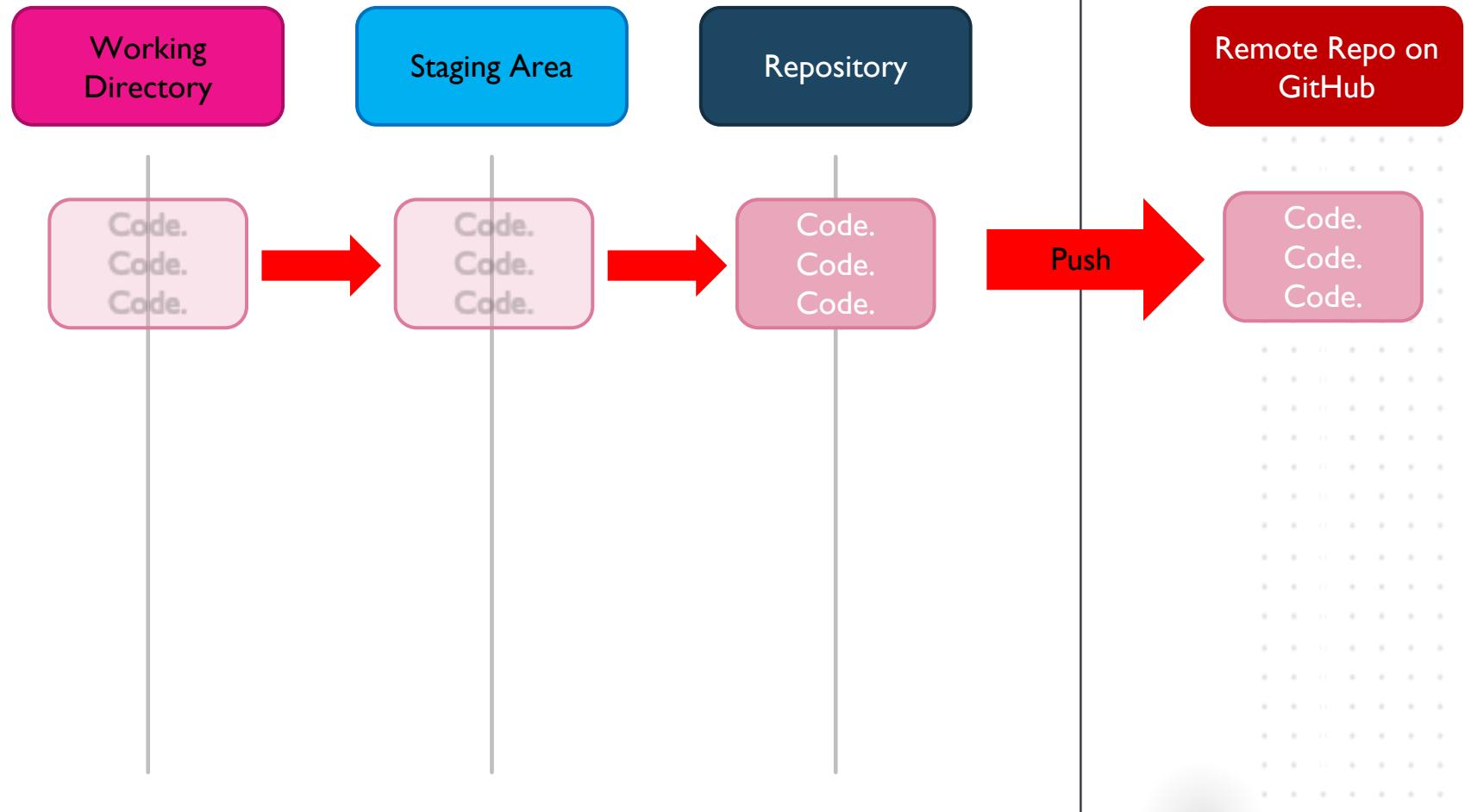


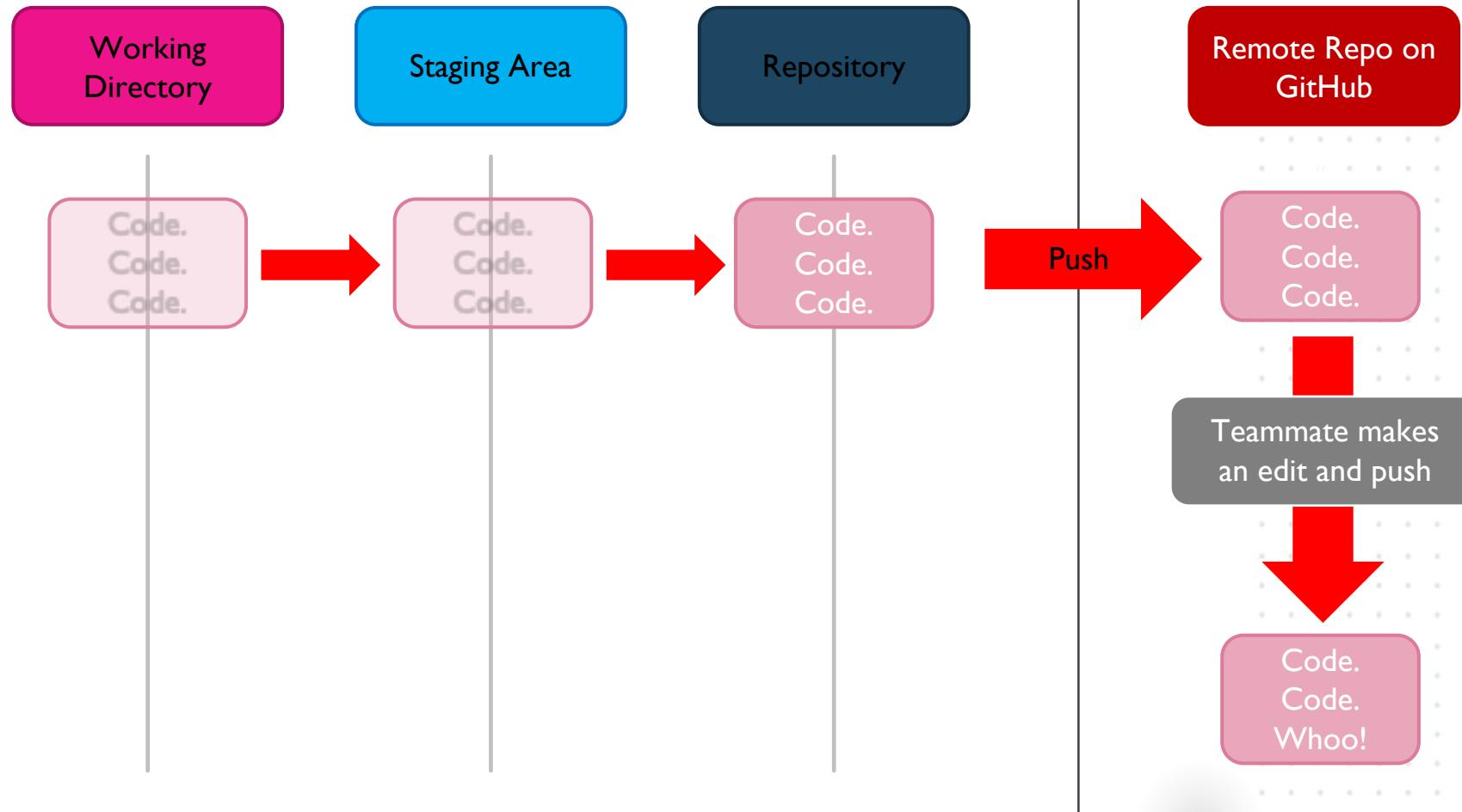
GitHub: Remote vs Local

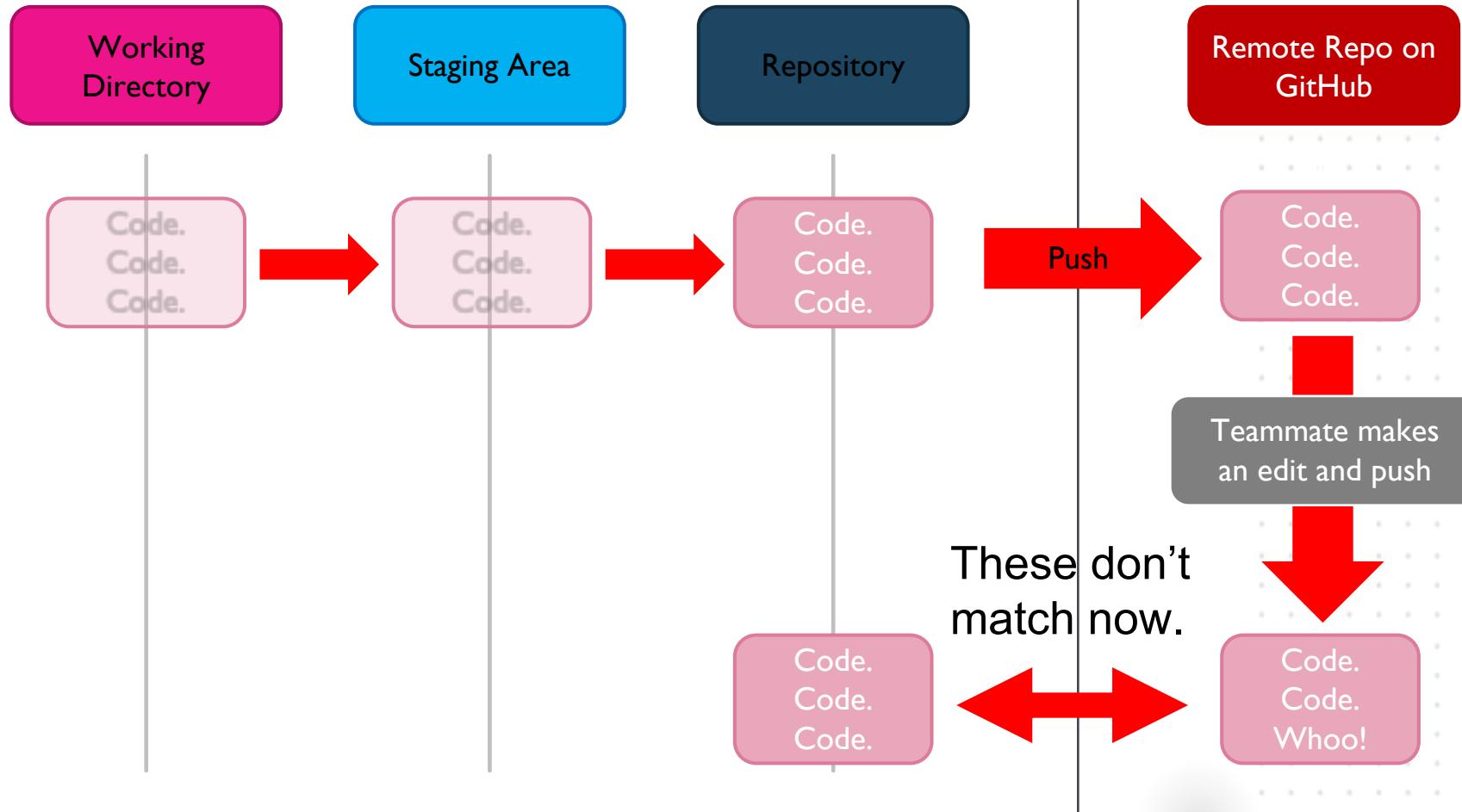


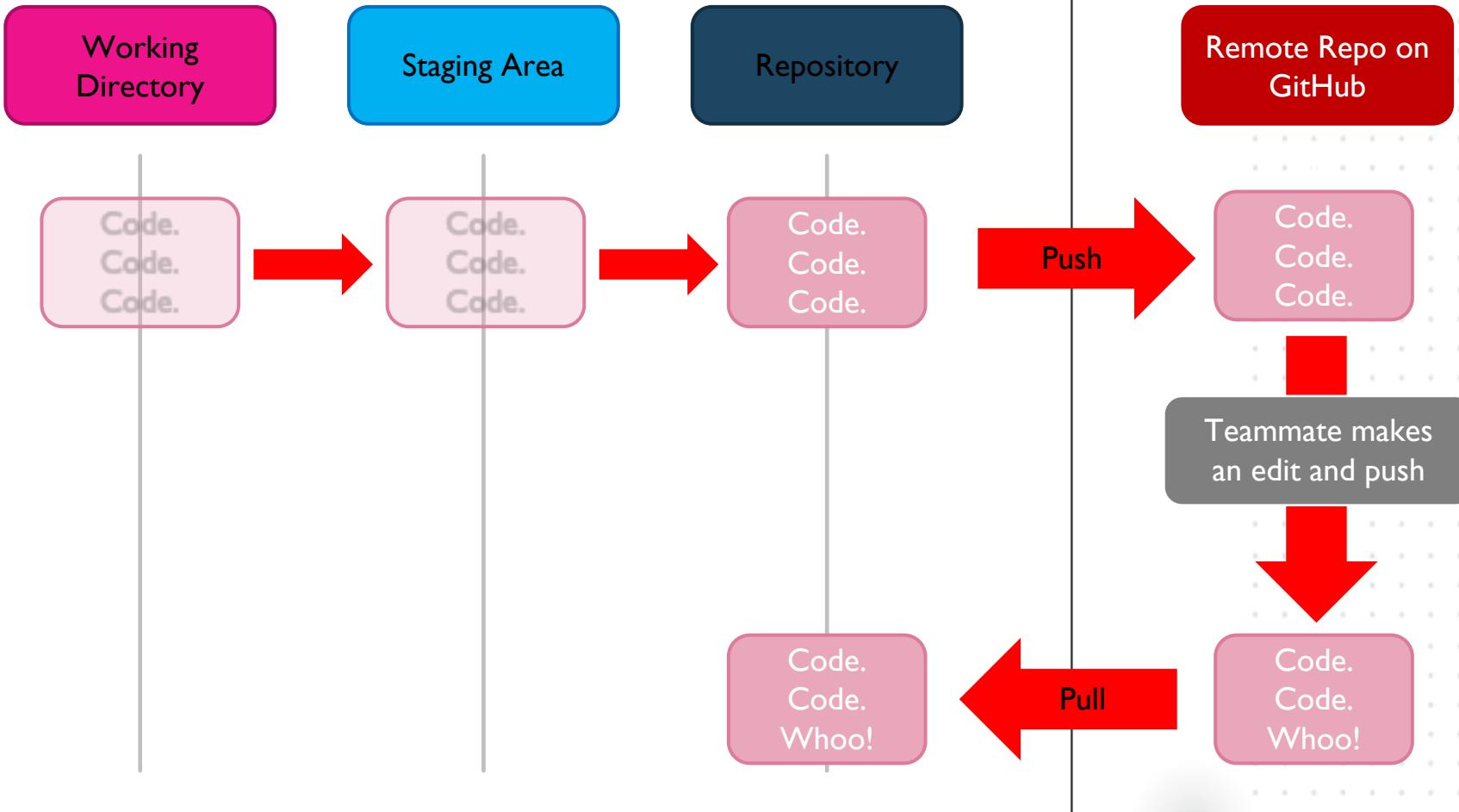
- We use pushes and pulls to move the current version of our repo between our computer and a “remote” version of the repo.
 - A push says, “make the remote version look like my code” and in particular it updates any files that changed in the last commits.
 - A pull says, “bring in any changes that have occurred on the remote version of the repo and make my code match those”











GitHub: Remote vs Local

- One subtlety: GitHub only modifies files that have CHANGED since the last time you asked Git to track them. So if I never asked Git to track a certain file (by making an add-commit), even if I push my code, it won't go to the remote version).

Exercise: Making your repo remote



**Let's start by
creating a repo on
GitHub called
“Test”**



Connecting GitHub to our Local Repo



- We now have a remote repo (on GitHub) and the local one where we put our test files. Let's link those together.

First, in the terminal, move to the repo we started earlier. You need to be in the folder where we did the `git init` earlier. (or any directory below that)



Connecting GitHub to our Local Repo



- We now have a remote repo (on GitHub) and the local one where we put our test files. Let's link those together.

```
Terminal:> git remote add origin URL_FROM_GITHUB
```



This tells git are about to tell it about the non-local version of the repo

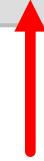


Connecting GitHub to our Local Repo



- We now have a remote repo (on GitHub) and the local one where we put our test files. Let's link those together.

```
Terminal:> git remote add origin URL_FROM_GITHUB
```



This tells git we're adding a new remote place to interface with



Connecting GitHub to our Local Repo



- We now have a remote repo (on GitHub) and the local one where we put our test files. Let's link those together.

```
Terminal:> git remote add origin URL_FROM_GITHUB
```



This tells git that we're going to call this remote place "origin" (it's convention)



Connecting GitHub to our Local Repo



- We now have a remote repo (on GitHub) and the local one where we put our test files. Let's link those together.

```
Terminal:> git remote add origin URL_FROM_GITHUB
```



This is specific to your repo and tells git where the remote repo lives.



Connecting GitHub to our Local Repo



- Make sure it worked by testing your remote locations using:

```
Terminal:> git remote -v
```

I see an output like this:

```
origin https://github.com/ZWMiller/test (fetch)
origin https://github.com/ZWMiller/test (push)
```



Connecting GitHub to our Local Repo



- Now let's move our code to GitHub

```
Terminal:> git push origin master
```

- When we go back to our GitHub and refresh, we should see our files there!



Connecting GitHub to our Local Repo



- Now let's move our code to GitHub

```
Terminal:> git push origin master
```



This told git we wanted to move our commit history up to GitHub



Connecting GitHub to our Local Repo



- Now let's move our code to GitHub

```
Terminal:> git push origin master
```



This told git to use the remote place we just created called “origin”



Connecting GitHub to our Local Repo



- Now let's move our code to GitHub

```
Terminal:> git push origin master
```



This told git to put those changes on the “master” branch. We’ll talk about branches later, but master is the most important one.



Summary



- Git and GitHub are powerful tools for maintaining good code.
- They allow us to checkpoint via commits, and give us the power to move back in time if we break our code.
- GitHub allows us to share our code with others and allows multiple people to make changes to the code.



Best Practice



- A smart person would keep a GitHub repo for all of their projects. They would create one right after this lecture for project 1. That way, if they happen to spill coffee on their laptop, their code is backed up.
- It's a good idea to commit/push often. I usually do so after every completed “point” in my code. Finished a few functions that now work together? Commit. Updated comments in the code? Commit.



QUESTIONS?

