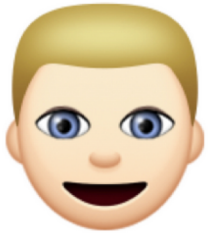
METIS

INTRODUCTION TO GIT BRANCHES

GIT BRANCHES

- Git is a great tool for version controlling your code.
- We’ve claimed Git is great for coding teams; let’s see how.

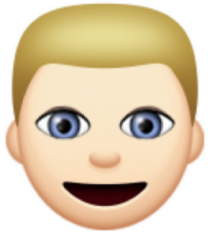


Alice and Brian are on the same team. Brian wants to use some of Alice’s code.

```
def get_data(query):
    """
    Gets 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(users.value_counts())

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



```
def get_data(query):
    """
    Gets 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(users.value_counts())

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

Alice makes a commit to the repo. Brian's code is now out of date.

```
def get_data(query):
    """
    Gets 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(users.value_counts(sort=False))

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

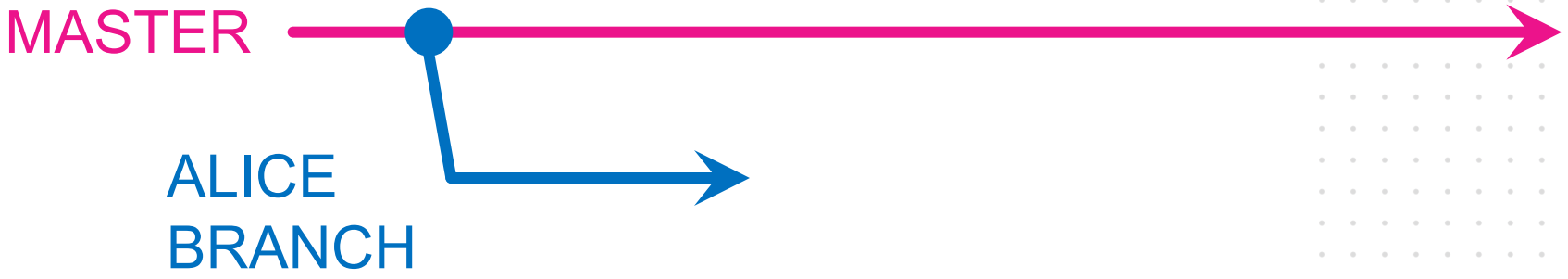
```
def get_data(query):
    """
    Gets the data from the database
    and returns it in dataframe format
    """
    table_data = sql.read(query)
    return pd.DataFrame(table_data)

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

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

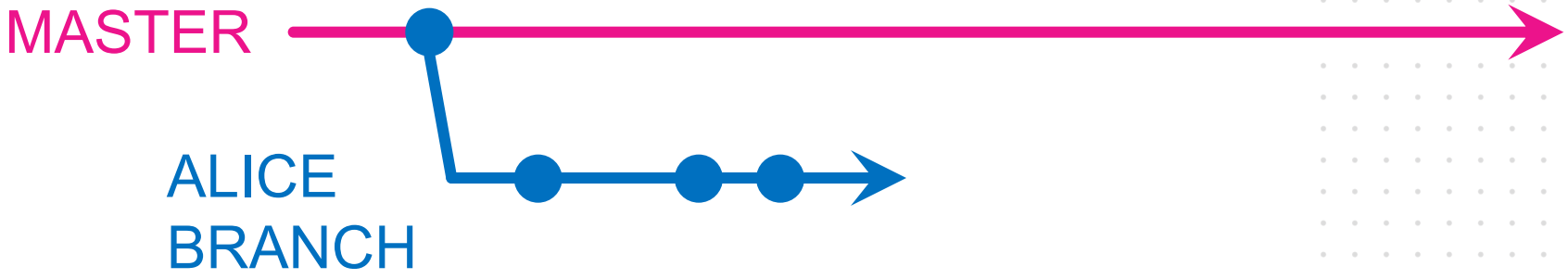
CODE BRANCHING

CODE BRANCHING



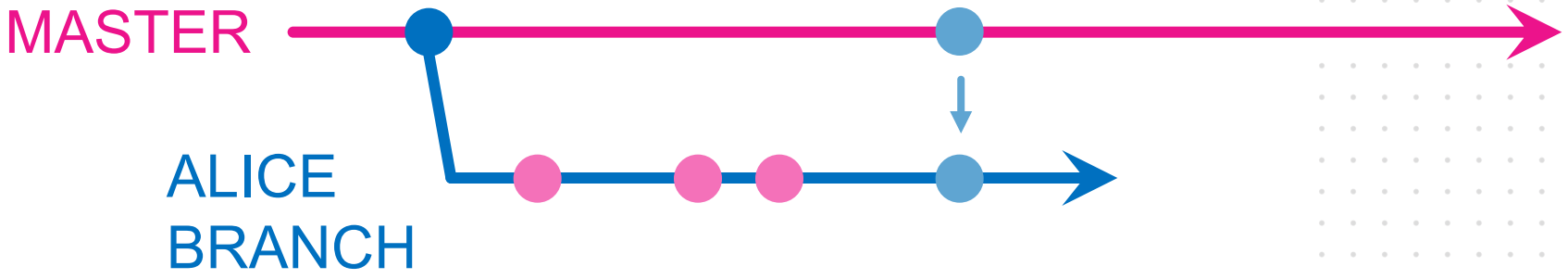
Branches allow users to edit code without destroying the Master.
Those edits only get added to the master when the user is ready.

CODE BRANCHING



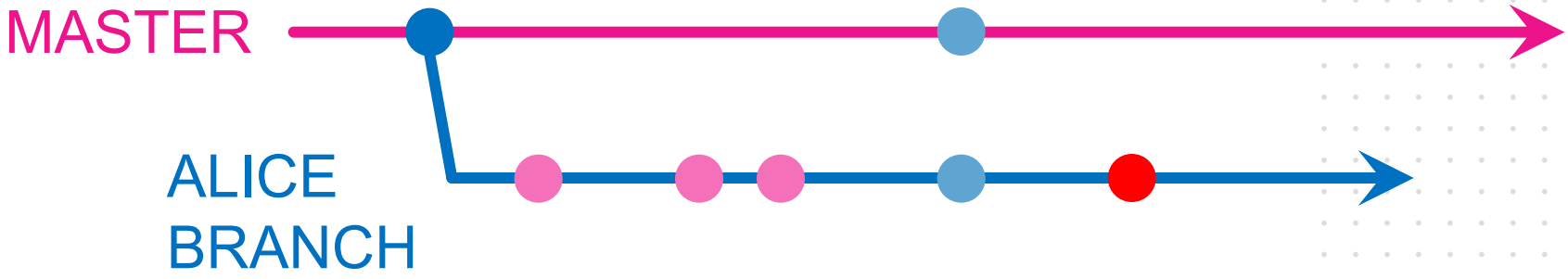
Alice can make whatever changes she wants on her own branch. She can make commits to her branch.

CODE BRANCHING



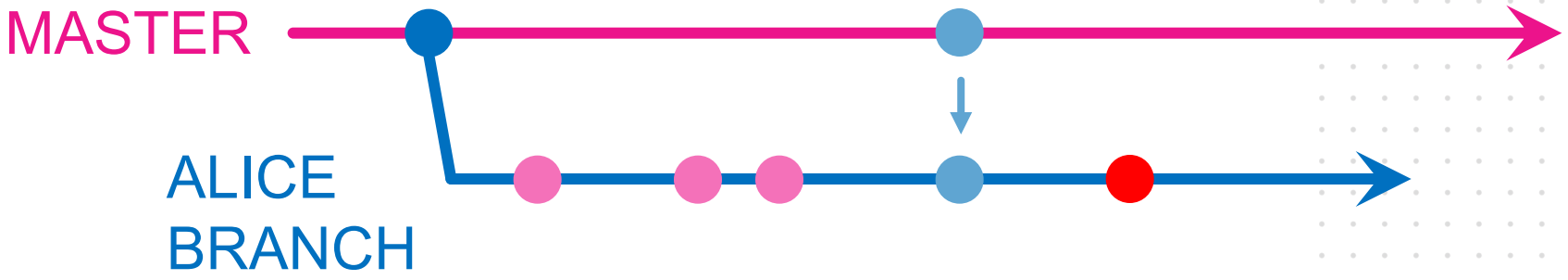
If someone changes the master she can keep up with other changes to the master branch using “**merges**”.

CODE BRANCHING



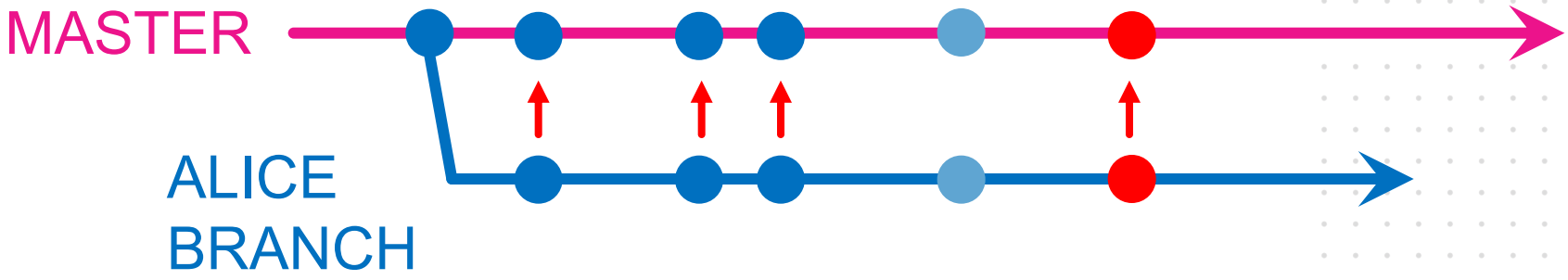
Then she can tell the Master branch, “hey, incorporate my code changes into you.” This is done via “Pull Request”

CODE BRANCHING



Note that before she makes a pull request, Alice got the most recent version of MASTER via a merge. This is best practice.

CODE BRANCHING

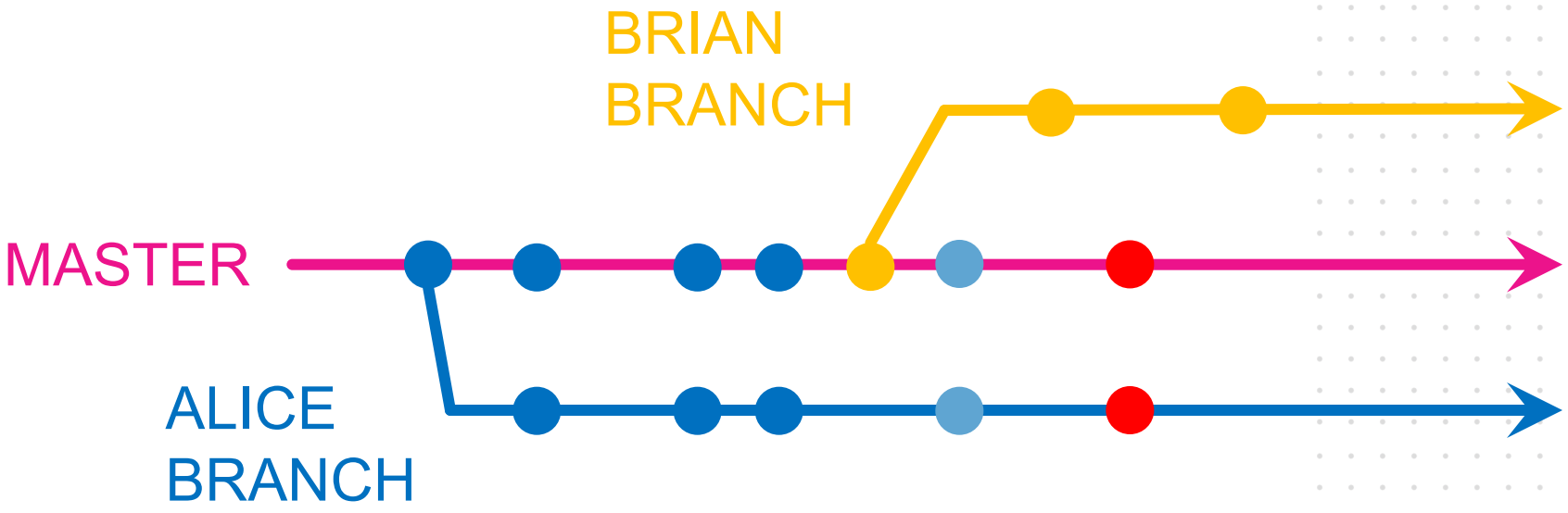


When someone approves the pull request, Alice's code gets added to the master branch, with all of the commit history via a merge.

What problems does this solve?

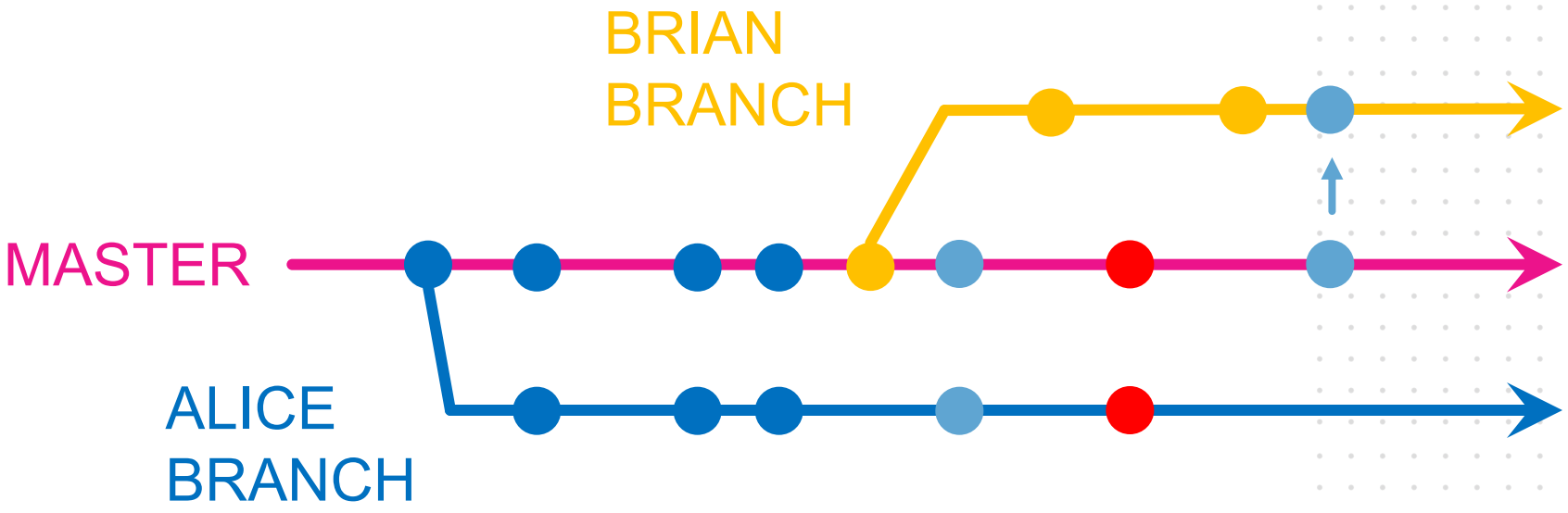
- Branches allow many members of the team to work on different code problems at once.
- The Master branch doesn’t break every time someone is testing new code.

CODE BRANCHING



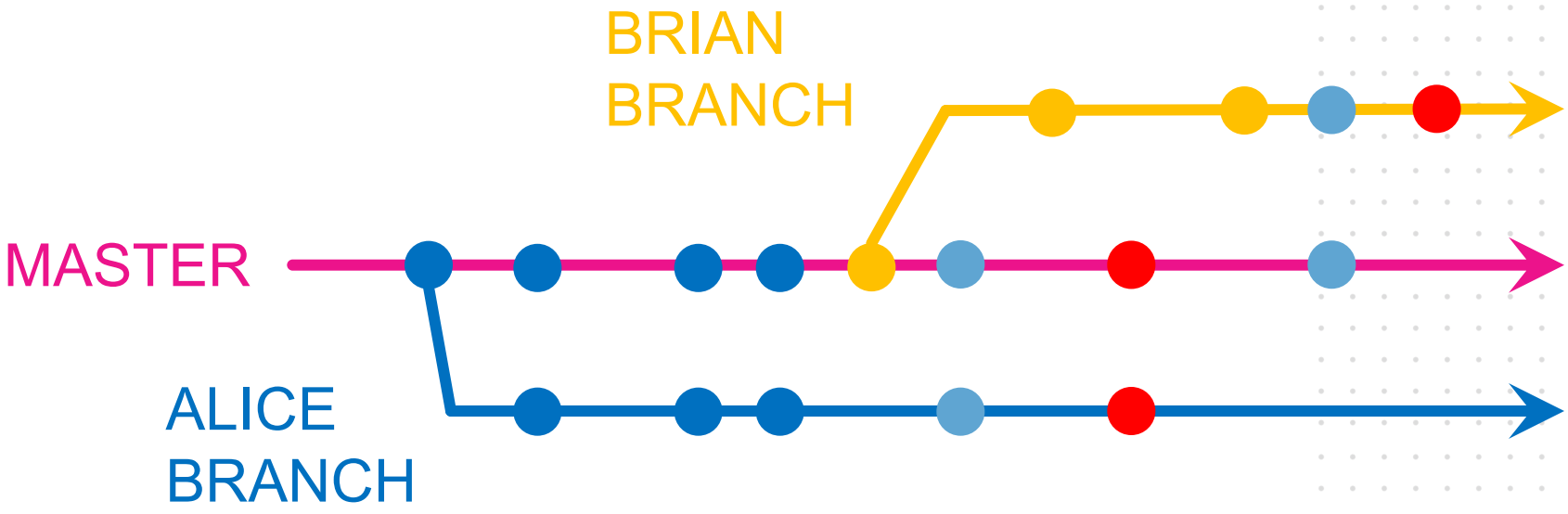
Brian can make his own branch, with his own changes.

CODE BRANCHING



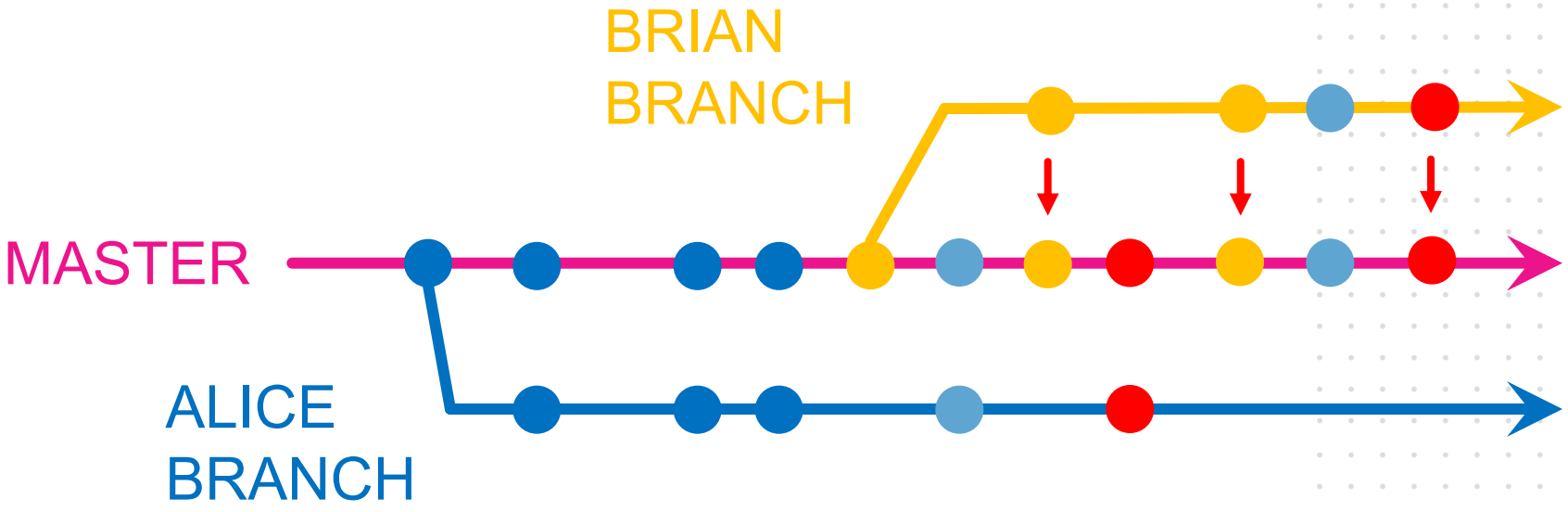
He merges the master into his branch first to make sure he gets all of Alice's code.

CODE BRANCHING



Brian can push back to Master too.

CODE BRANCHING



Brian can push back to Master too.

CODE BRANCHING

Alice's and Brian's changes made it back to the master branch. They were working on different pieces at the same time, but were able to contribute to the project in a cohesive way.

Exercise: Submitting to the class repository

- First, we need to make a local copy of the class repo for each of you. Navigate to the GitHub page for the class, and copy the link to the main page.
- Now in your terminal, navigate to wherever you want the Metis repo to live. I'm going to my Documents area:

```
Terminal:> cd ~/Documents
```

- Get all the files from the repo and connect it all together. We can make a copy using git's "clone" functionality.

```
Terminal:> git clone LINK_TO_GITHUB_REPO
```

- Move into the newly created directory. The directory will be of the format ”city##_ds##”.
- Check to see where your “remote” is pointing. It should be the Metis repo.

```
Terminal:> cd chi18_ds8
Terminal:> git remote -v
```

- Make a branch and add some files to the Metis repo.

```
Terminal:> git branch whatever_I_want_to_name_my_branch
```

This tells git we want to make a new branch.

- Switch over to your branch

```
Terminal:> git checkout whatever_I_want_to_name_my_branch
```

This tells git we want to switch branches.

- Check to make sure you're on your branch.

```
Terminal:> git branch
```

The one with the star next to it is your active branch. It should be your new branch now.

- Go to the test area and add some files.

```
Terminal:> cd student_submissions/test_area
```

```
Terminal:> echo “TEST” > my_user_name.txt
```

- Commit changes to the repo, then make sure we have the master branch updates.

```
Terminal:> git add my_user_name.txt
```

```
Terminal:> git commit –m “added a test file by user_name”
```

```
Terminal:> git merge master
```

- Push the new commits to the GitHub repo.

```
Terminal:> git push origin whatever_I_want_to_name_my_branch
```

Note that we’re not pushing to master!

- Push the new commits to the GitHub repo.

```
Terminal:> git push origin whatever_I_want_to_name_my_branch
```

Note 2: You might have trouble pushing if GitHub doesn't know who you are. If so, try using git config to set your username (google)

- After that push, GitHub now has your changes on your specific branch. So if we want it to become part of master, we need to make a Pull Request.

- Let’s do that together now.

GitHub Branches

- Branches allow us all to work on code concurrently and merge it into a single project.
- Pull Requests are how we tell the Master branch our code is ready to moved over to the main code.
- Real software teams use branches ALL the time. Any time a feature is being added, or a bug is being fixed, it gets a branch and the developer works on that branch until she solves the problem. Then she PR's it back to production.

QUESTIONS?

Pull Request Demo

thisismetis / chi18_ds7 Private

Unwatch 6 | Star 2 | Fork 1

Chicago 2018 Summer Bootcamp

Edit

Manage topics

917 commits | **26** branches | **0** releases | **15** contributors

Your recently pushed branches:

zwm_test (less than a minute ago) — Compare & pull request

Branch: **master** | New pull request | Create new file | Upload files | Find file | Clone or download

ZWMiller Merge branch 'master' of https://github.com/thisismetis/chi18_ds7 — Latest commit f9ca7f4 a day ago

administrative	first push	3 months ago
challenges	release all other challenges	a month ago
class_lectures	type in w10:d3 pair	20 days ago
images	add images	3 months ago
investigations	remerged with remote branch and corrected file location	2 months ago
optional_lectures	Update advanced_bash.md	14 days ago
projects	remove accidental push	3 months ago
resources	update review docs	a day ago
student_submissions	Merge pull request #296 from thisismetis/courtney_final_presentation	7 days ago
.gitignore	first push	3 months ago