



Pulling and Merging with Git

Data Boot Camp
Lesson 7.2



Class Objectives

By the end of today's class you will...



Pull a branch from GitHub



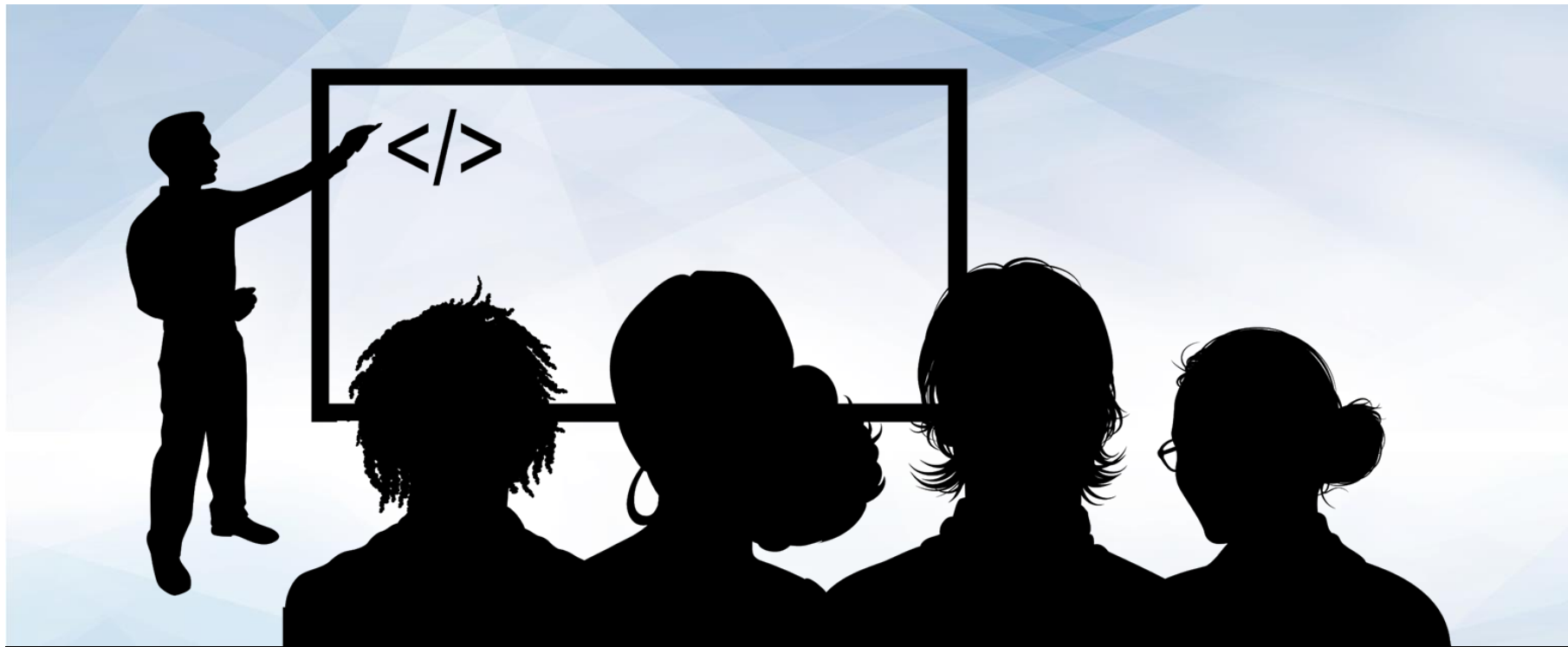
Merge branches with Git



Open, review, and merge PRs with GitHub



Continue work on Project 1



Instructor Demonstration

Merging on Git and GitHub



What is a **branch** in Git?



A branch is a timeline and history of changes.

There are many benefits to developing on a **separate branch!**



Your
commits,
at the top
of the logs

Checkout
old commits
without
affecting
master

Organize
your
development
efforts

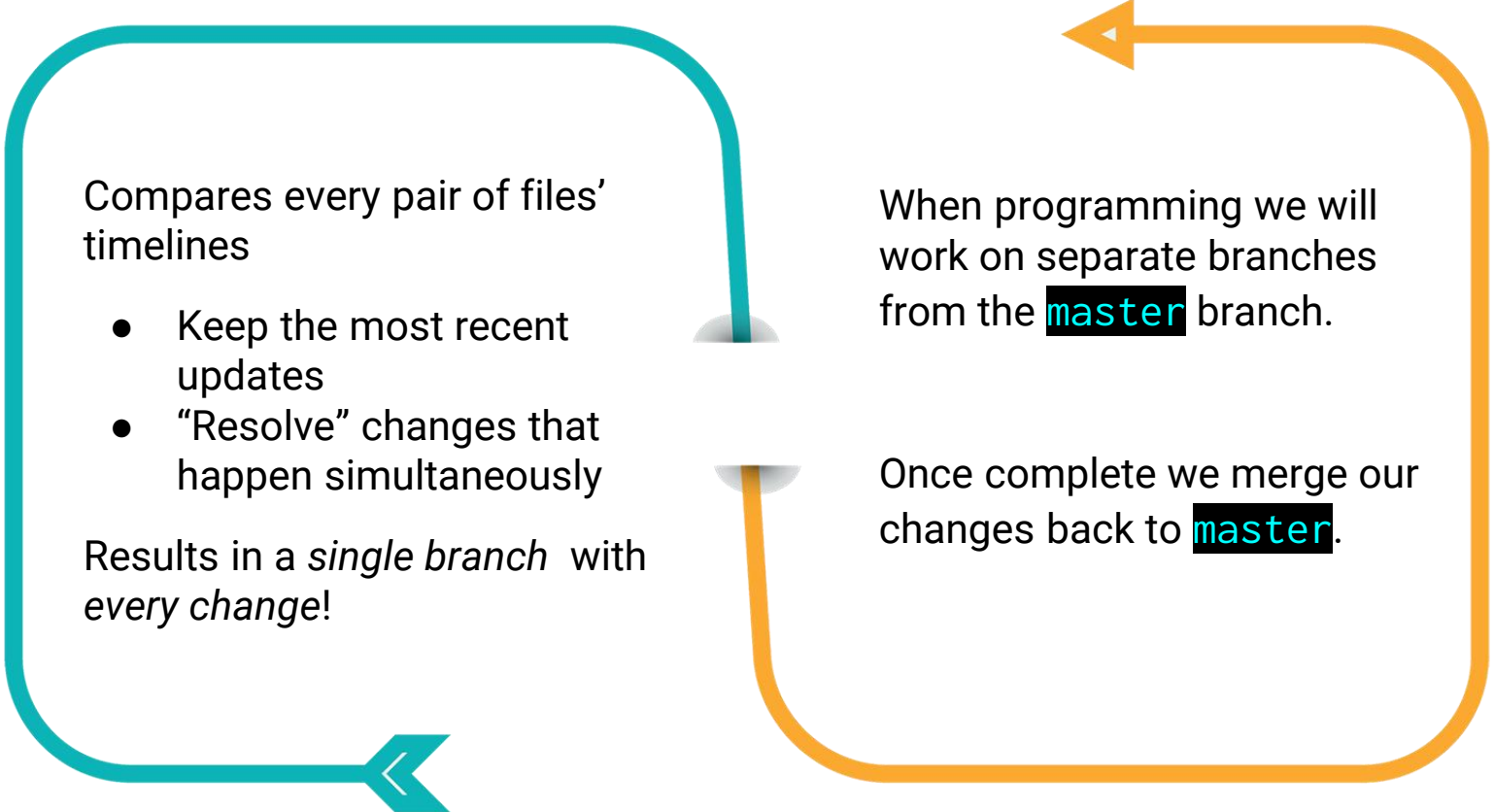


What does it mean to
merge a branch?



A merge **combines** two
branches together.

Merging Combines Two Branches Together



Compares every pair of files' timelines

- Keep the most recent updates
- “Resolve” changes that happen simultaneously

Results in a *single branch* with *every change*!

When programming we will work on separate branches from the **master** branch.

Once complete we merge our changes back to **master**.

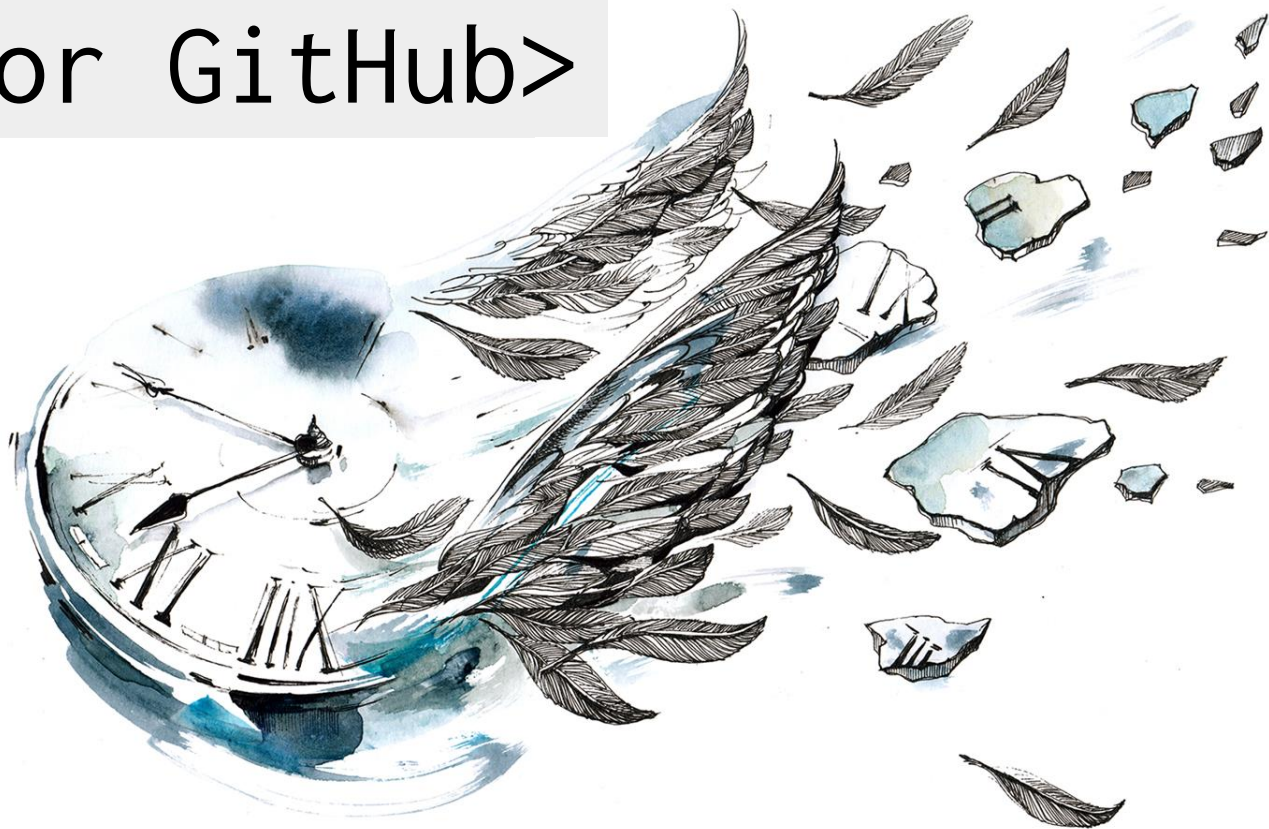


We can merge branches
using our **local repository**
or **GitHub!**

<Time for Git>



<Time for GitHub>



Before You Merge Branches Together

Ensure that the local branch is up-to-date with the remote branch.



Use `git pull origin <branch name>`



Then `git checkout <merge branch>` and `git merge`

Does the remote branch have new and exciting content, bug fixes or improvements? Are you not ready to merge branches?

There is no need to wait! Use `git pull origin <branch name>` to pull down changes at any time.



Take a Break!



For the remainder of
class... project work!

