



# GitHub Developer Training

Presented by GitHub Professional Services

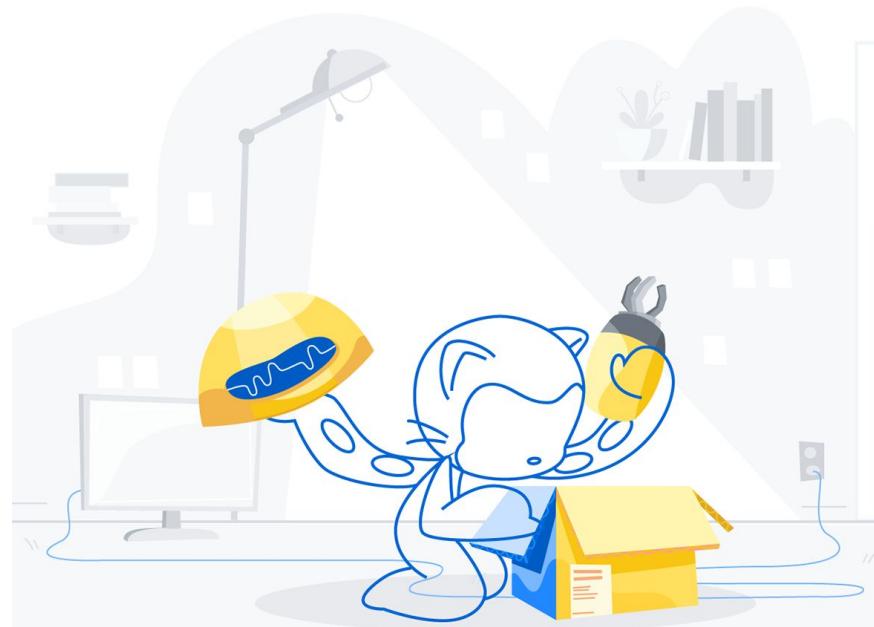
# Objectives



- Have a hands-on training in how to **collaborate** effectively using the GitHub's toolset
- Understand how to **navigate** around GitHub
- Maximise **Git** and **GitHub** usage to streamline your development lifecycle
- Implement the **GitHub flow** in building software
- Resolve various types of **merge conflict** as they arise
- **Protect branches** with rules that work for your team using the **CODEOWNERS** file
- Mark moments in time within your repo using **tags** and **releases**
- **Rewrite the history** of your project in a way that suits you

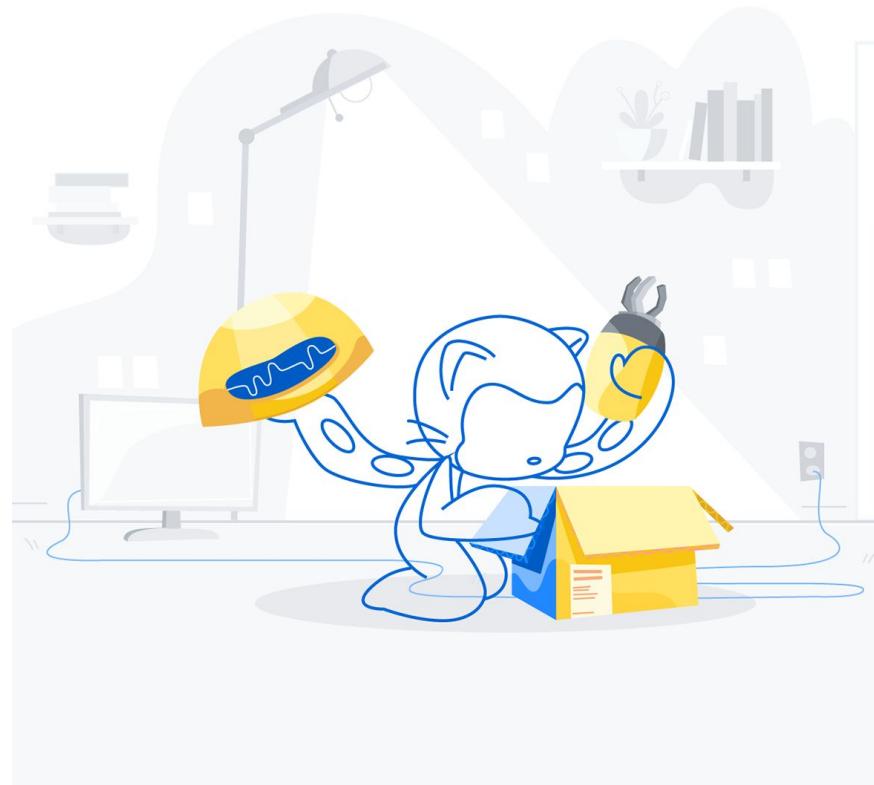
# Agenda: Part 1

- **Getting started**
  - Getting ready for class
  - Getting started
  - GitHub flow
- **Project 1: caption this**
  - Branching with Git
  - Local Git Configs
  - Working locally
  - Collaborating on code
  - Editing on GitHub
  - Merging Pull Requests
  - Local History
  - Streamline workflow with aliases
- **Project 2: Merge Conflicts**
  - Defining a merge conflict
  - Resolving a merge conflict



# Agenda: Part 2

- **Project 3: GitHub Games**
  - Workflow review
  - Protected branches & CODEOWNERS
  - Git bisect
  - Reverting commits
  - Helpful Git commands
  - Viewing local changes
  - Tags and releases
  - Workflow discussion
- **Project 4: Local repository**
  - Create a local repository
  - Fixing commit mistakes
  - Git reset
  - Cherry picking
  - Merge strategies



# Getting ready for class

User account

● Install Git

● Cloning and proxy configuration

Personal Access Tokens



# Setting up a GitHub.com user account

1. Access GitHub.com and click Sign up.
2. Choose the free account.
3. You will receive a verification email at the address provided.
4. Click the link to complete the verification process.



## Install Git

- `git --version`
- If you don't already have Git installed, you can download Git at [www.git-scm.com](http://www.git-scm.com).



# Cloning and proxy configuration

- `git clone`  
<https://github.com/githubschool/scratch>

**Proxy configuration if no authentication required**

- `git config --global http.proxy`  
`https://YOUR.PROXY.SERVER:8080`

**Proxy configuration if authentication required**

- `git config --global http.proxy`  
`https://YOUR_PROXY_USERNAME:YOUR_PROXY_P`  
`ASSWORD@YOUR.PROXY.SERVER:8080`

# Personal Access Tokens

Settings / Developer settings

GitHub Apps
OAuth Apps
Personal access tokens

## New personal access token

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

### Note

repo-access

What's this token for?

### Expiration \*

30 days

The token will expire on Thu, Oct 7 2021

### Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

<input type="checkbox"/> <b>repo</b>	Full control of private repositories
<input type="checkbox"/> repo:status	Access commit status
<input type="checkbox"/> repo_deployment	Access deployment status
<input type="checkbox"/> public_repo	Access public repositories
<input type="checkbox"/> repo:invite	Access repository invitations
<input type="checkbox"/> security_events	Read and write security events
<input type="checkbox"/> workflow	Update GitHub Action workflows
<input type="checkbox"/> write:packages	Upload packages to GitHub Package Registry
<input type="checkbox"/> read:packages	Download packages from GitHub Package Registry
<input type="checkbox"/> delete:packages	Delete packages from GitHub Package Registry
<input type="checkbox"/> admin:org	Full control of orgs and teams, read and write org projects
<input type="checkbox"/> write:org	Read and write org and team membership, read and write org projects
<input type="checkbox"/> read:org	Read org and team membership, read org projects
<input type="checkbox"/> admin:public_key	Full control of user public keys
<input type="checkbox"/> write:public_key	Write user public keys



Demo

# Getting started

Version control systems



Git



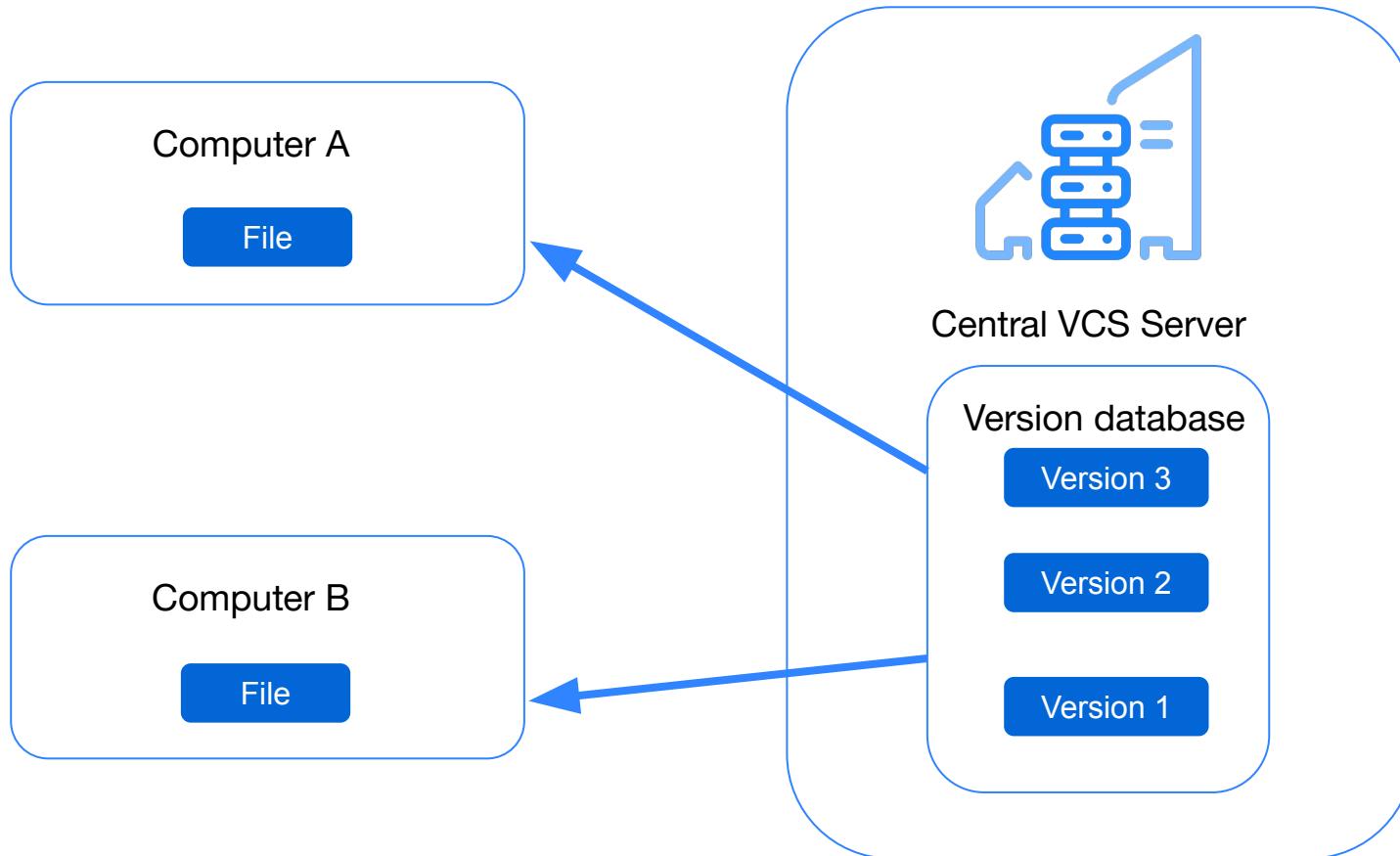
GitHub

# What is version control?

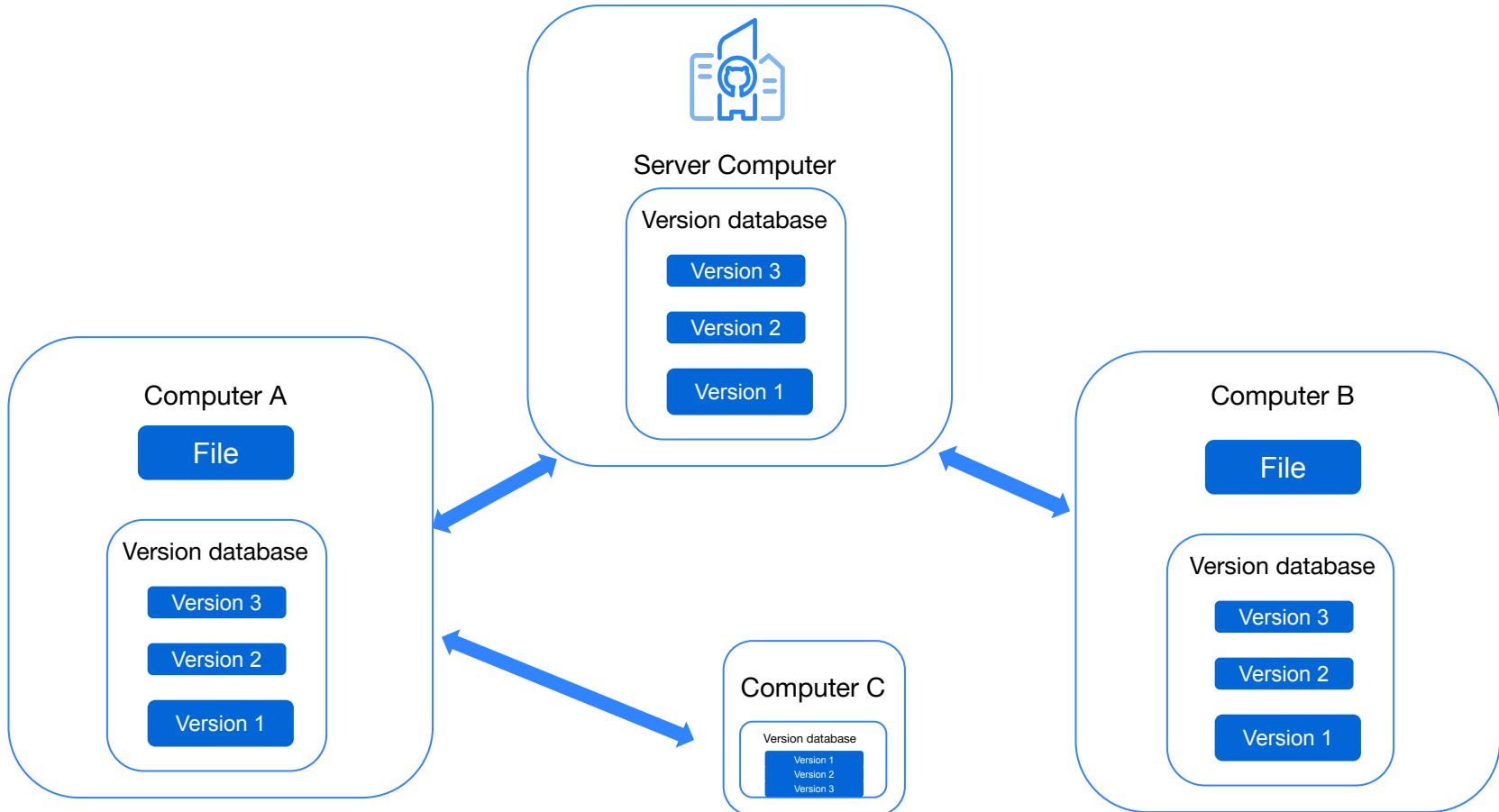
**“Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later.”**

<https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control>

# Centralized Version Control



# Distributed Version Control



# Getting started

Version control systems



Git



GitHub



- A **distributed** version control system (DVCS)
- **Free** and **open source**
- Designed to handle everything from **small** to very **large** projects with speed and efficiency
- Both small in size and **fast**
- Optimized for **local operations**
- **Branching**
- **Snapshots**, not deltas

# Snapshots vs deltas

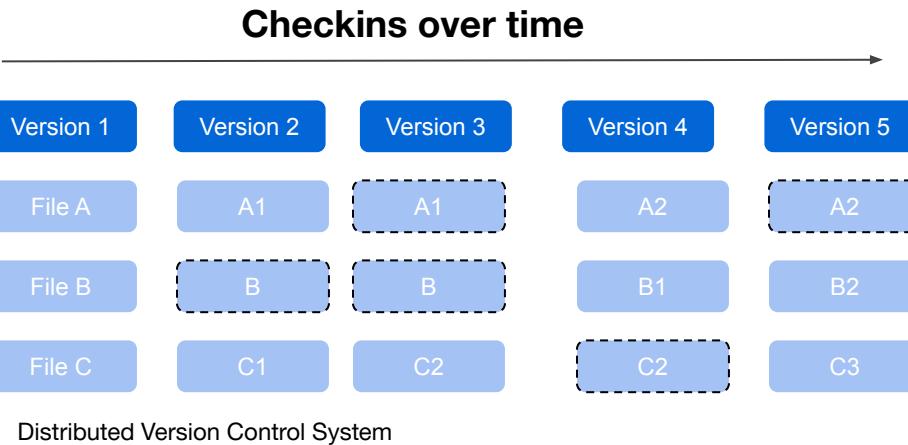
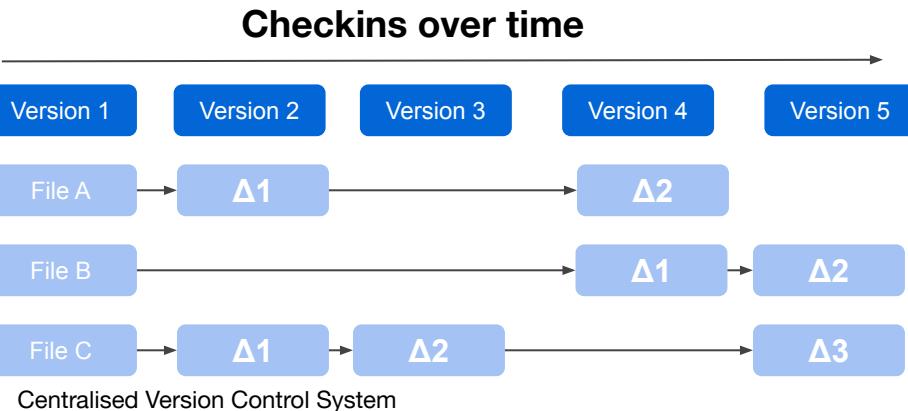
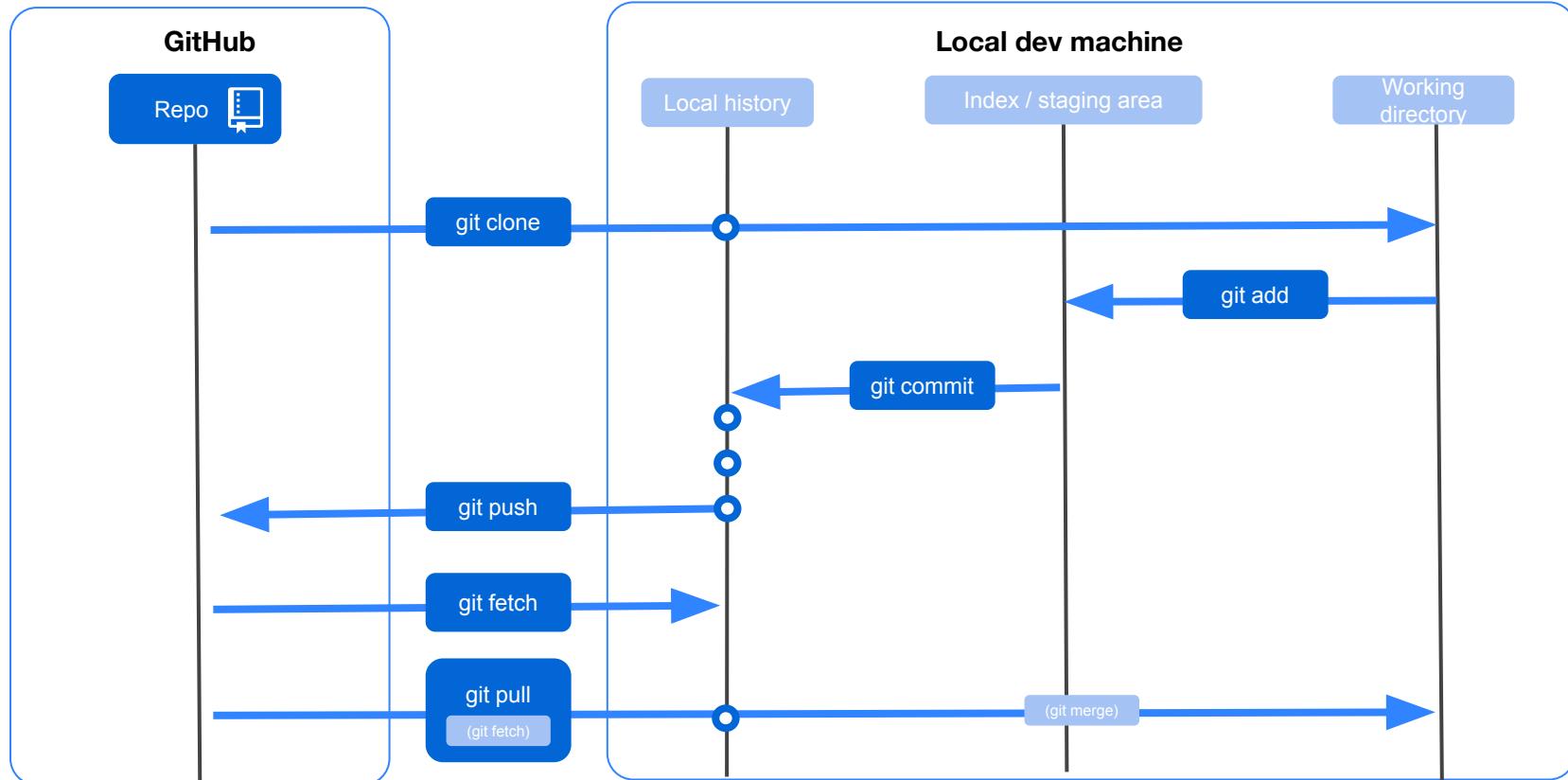


Image source: ProGit v2 by Scott Chacon



# How Git interacts with GitHub



# Getting started

Version control systems



Git



GitHub

- Ecosystem
- Repository navigation



- Technology platform built **on top** of **DVCS**
- **Social** side of Git
- **Code hosting** and **collaboration** platform
- **Community** of developers
- **Ecosystem** of integrations
- **Team** collaboration through:
  - **Issues**
  - **Pull requests**
  - **Organisation** and **teams**
  - **Projects**

# Getting started

Version control systems



Git



GitHub

- Ecosystem
- Repository navigation

# The GitHub ecosystem

Pull requests Issues Marketplace Explore

## Extend GitHub

Add tools to help you build and grow

Explore apps

Types

Actions

Categories

API management

Chat

Code quality

Code review

Continuous integration

Dependency management

Deployment

IDEs

Search for apps and actions Sort: Best Match

**CodeFactor**  
By codefactor-io Recommended  
Automated code review for GitHub

**Datreo**  
By datreeio Policy enforcement solution for confident and compliant code  
↓ 1.8k installs

**Testspace.com**  
By testspace-com Integrated GitHub test solution for CI Results Dashboard, Reporting, Manual Test Case Management, and Exploratory  
↓ 337 installs

**CircleCI**  
By circleci Automatically build, test, and deploy your project in minutes  
Recommended

**Ranger**  
By reporanger A sidekick for repository maintainers  
↓ 393 installs

**Layer CI**  
By webappio A staging server replacement and CI/CD tool for full-stack apps  
↓ 791 installs



# Getting started

Version control systems



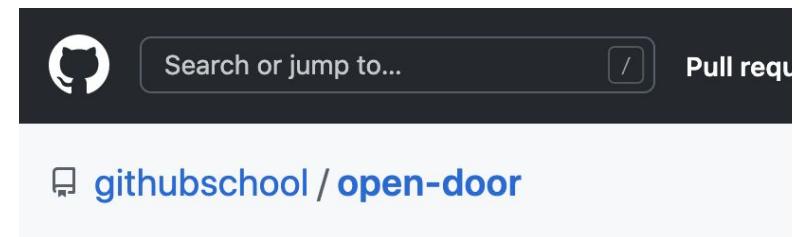
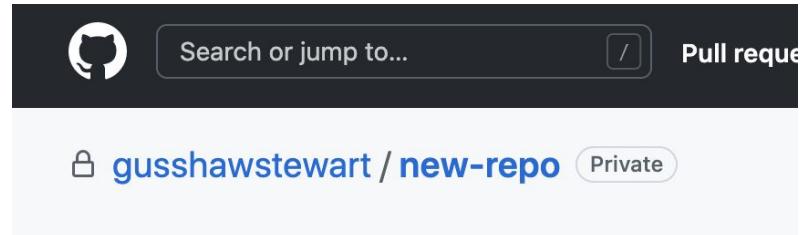
Git



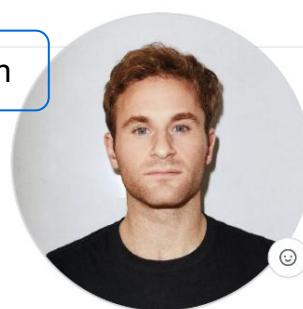
GitHub

- Ecosystem
- Repository navigation

# User accounts vs organization accounts



## Global navigation

**Gus**

gusshawstewart

[Edit profile](#)

9 followers · 12 following · 9

 GitHub STAFF

## Organizations



## Contributions

[Overview](#) [Repositories 76](#) [Projects](#) [Packages](#)

## Navigation tabs

Customize your pins

## Popular repositories

[wdi-fundamentals-rps](#)

Forked from fishermanswharf/wdi-fundamentals-rps-experimental

[WDI\\_20\\_Project\\_2](#)

My Second project from my WDI course at General Assembly

[WDI\\_2\\_\\_PROJECT\\_4](#)

ga project 4

[WDI\\_20\\_Project\\_1](#)

This is my first project from my immersive course at General Assembly.

[gusshawstewart.github.io](#)

My portfolio

[WDI\\_20\\_PROJECT\\_3\\_SOLO](#)

Group project - Alissa, Gus, Viktor

 Single sign-on to see contributions within the [githubcustomers](#) organization.

2021

2020

2019

2018

2017

 Single sign-on to see contributions within the [saml-sso-org](#) organization. Single sign-on to see contributions within the [gus-avocado-org](#) organization. Single sign-on to see contributions within the [snow-test-org](#) organization.

1,447 contributions in the last year

Contribution settings ▾

2016





Search or jump to...

Pull requests Issues Marketplace Explore



## GitHub School

The Interwebs <https://services.github.com/> [services@github.com](mailto:services@github.com)

[Overview](#) [Repositories 6.4k](#) [Packages](#) [People 123](#) [Teams 54](#) [Projects 1](#) [Insights](#) [Settings](#)

## Navigation tabs

Pinned

Customize your pins

People



[View all](#)

[Invite someone](#)

[Repositories](#)

Find a repository...

Type ▾

Language ▾

Sort ▾

[New](#)

[congenial-giggle](#)

Let's learn about Git and GitHub

HTML 0 stars MIT 0 forks 1 issues 0 updated 28 minutes ago

[github-games-deannalo68](#) Private

A fun way to learn about git troubleshooting.

HTML 0 stars MIT 0 forks 2 issues 1 updated 36 minutes ago

[github-games-joseatbunnings](#) Private

A fun way to learn about git troubleshooting.

HTML 0 stars MIT 0 forks 2 issues 1 updated 36 minutes ago

[github-games-JunyLaksmana](#) Private

A fun way to learn about git troubleshooting.

HTML 0 stars MIT 0 forks 2 issues 1 updated 37 minutes ago

[github-games-PowerPlant987](#) Private

A fun way to learn about git troubleshooting.

Top languages

HTML JavaScript Ruby CSS

Python

Most used topics

Manage

training git github tetris

javascript



Demo

< Code

Branches

main Branches Tags

Go to file Add file

Repo management

Files

gusshawstewart	Create main.yml	✓ 3188896 on 27 Jul	198 commits
· .github/workflows	Create main.yml	last month	
· _includes	Update script.html	16 months ago	
· _layouts	initial commit	4 years ago	
· _sass	initial commit	4 years ago	
· _slides	Merge pull request #15 from githubschool/charanya	last month	
· assets	initial commit	4 years ago	
· docs	their --> your	4 years ago	
· reveal.js @ 94d98ff	Update reveal.js	16 months ago	
· script	Update server script	7 months ago	
· .editorconfig	initial commit	4 years ago	
· .gitignore	Ignore vendor directory	10 months ago	
· .gitmodules	initial commit	4 years ago	
· Gemfile	fix: Gemfile & Gemfile.lock to reduce vulnerabilities	2 years ago	
· Gemfile.lock	Bump addressable from 2.7.0 to 2.8.0	2 months ago	
· LICENSE	initial commit	4 years ago	
· README.md	Update README.md	last month	
· _config.yml	update config with class information	last month	
· circle.yml	initial commit	4 years ago	
· index.html	posts -> slides	16 months ago	

README.md

GitHub for Developers!!!

- Class Date: 27 July, 2021
- Facilitator: @gusshawstewart

README

Talk to Us!!

About

Let's learn about Git and GitHub

githubschool.github.io/open-door/

Readme

MIT License

Releases

2 tags

Create a new release

Packages

No packages published  
Publish your first package

Contributors 34



+ 23 contributors

Environments 2

github-pages Active  
production Active

Languages



Metadata, releases, packages, contributors

Code

Issues 5

Pull requests

Discussions

Actions

Projects 1

Wiki

Security

Insights

Settings

## Label issues and pull requests for new contributors

Now, GitHub will help potential first-time contributors [discover issues](#) labeled with [good first issue](#)

Dismiss

Filters ▾

is:issue is:open

Labels 10

Milestones 2

New issue

Issues

  5 Open  11 Closed

Author ▾ Label ▾ Projects ▾ Milestones ▾ Assignee ▾ Sort ▾

  Parking Lot

#7 opened on 27 Jul by gusshawstewart

1

  Restyle the Slides

#6 opened on 27 Jul by gusshawstewart

  Add a Caption To An Existing Slide

#4 opened on 27 Jul by gusshawstewart

  Add A New Slide

#3 opened on 27 Jul by gusshawstewart

  Activity 1: Your First Caption

#2 opened on 27 Jul by gusshawstewart

26 tasks

⌚ ProTip! What's not been updated in a month: [updated:<2021-08-01](#).

New issue

Label issues and pull requests for new contributors

Now, GitHub will help potential first-time contributors [discover issues](#) labeled with [good first issue](#).

[Dismiss](#)

Filters  Labels 10 Milestones 2 [New pull request](#)

Author ▾ Label ▾ Projects ▾ Milestones ▾ Reviews ▾ Assignee ▾ Sort ▾

0 Open ✓ 26 Closed

There aren't any open pull requests.

You could search [all of GitHub](#) or try an [advanced search](#).

💡 ProTip! Updated in the last three days: updated:>2021-08-29.

Pull  
Requests



Search or jump to...

[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)

Log in + Create account

githubschool / open-door

[Watch](#) 1 [Star](#) 0 [Fork](#) 0

[Code](#) [Issues 5](#) [Pull requests](#)

[Discussions](#)

[Actions](#)

[Projects 1](#)

[Wiki](#)

[Security](#)

[Insights](#)

[Settings](#)



Search all discussions

New

Top: All ▾

Answered

Unanswered

Label ▾

New discussion

Categories



↑ 1

Welcome to open-door Discussions!

gusshawstewart announced now in Announcements



0

Discussions

∞ View all

Announcements

General

Ideas

Q&A

Show and tell

Most helpful

Be sure to mark someone's comment as an answer if it helps you resolve your question — they deserve the credit! ❤

[Community guidelines](#)

[githubschool.github.io/open-door](#)



Search or jump to...

Pull requests Issues Marketplace Explore

L + ⌂



githubschool / open-door

Watch 1 Star 0 Fork 0

Code

Issues 5

Pull requests

Discussions

Actions

Projects 1

Wiki Security

Insights

Settings

Workflows

New workflow

All workflows

Test action



## All workflows

Showing runs from all workflows

Filter workflow runs

2 workflow runs

Event ▾ Status ▾ Branch ▾ Actor ▾

Test action

Test action #2: Manually run by gusshawstewart

last month

15s

...

Create main.yml

Test action #1: Commit 3180896 pushed by gusshawstewart

main

last month

17s

...

GitHub  
Actions  
workflows

© 2021 GitHub, Inc. Terms Privacy Security Status Docs Site admin mode off



Contact GitHub

Pricing

API

Training Blog About

Q is:open

[New project](#)[1 Open](#) ✓ 0 Closed

Sort ▾

**Class Activity: Caption Contest**

🕒 Updated on 27 Jul

Let's see who can create the best captions for our collection of memes.

...

**Projects**© 2021 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Docs](#) [Site admin mode off](#)[Contact GitHub](#)[Pricing](#)[API](#)[Training](#)[Blog](#)[About](#)



## Welcome to the open-door wiki!

Wikis provide a place in your repository to lay out the roadmap of your project, show the current status, and document software better, together.

[Create the first page](#)

Wikis





Search or jump to...

[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)[githubschool / open-door](#)[Watch](#) 1 [Star](#) 0 [Fork](#) 0[Code](#) [Issues 5](#) [Pull requests](#) [Discussions](#) [Actions](#) [Projects 1](#) [Wiki](#)[Security](#)[Insights](#)[Settings](#)[Overview](#)[Security policy](#)[Security advisories](#)[Dependabot alerts](#)[Code scanning alerts](#)

## Security overview

### Security policy

Define how users should report security vulnerabilities for this repository

[Set up a security policy](#)

### Security advisories

View or disclose security advisories for this repository

[View security advisories](#)

### Dependabot alerts

Get notified when one of your dependencies has a vulnerability

[Enable Dependabot alerts](#)

### Code scanning alerts

Automatically detect common vulnerability and coding errors

[Set up code scanning](#)

Security  
policies and  
Dependabot

© 2021 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Docs](#) [Site admin mode off](#)

[Contact GitHub](#)[Pricing](#)[API](#)[Training](#)[Blog](#)[About](#)

Pulse

Contributors

Community

Traffic

Commits

Code frequency

Dependency graph

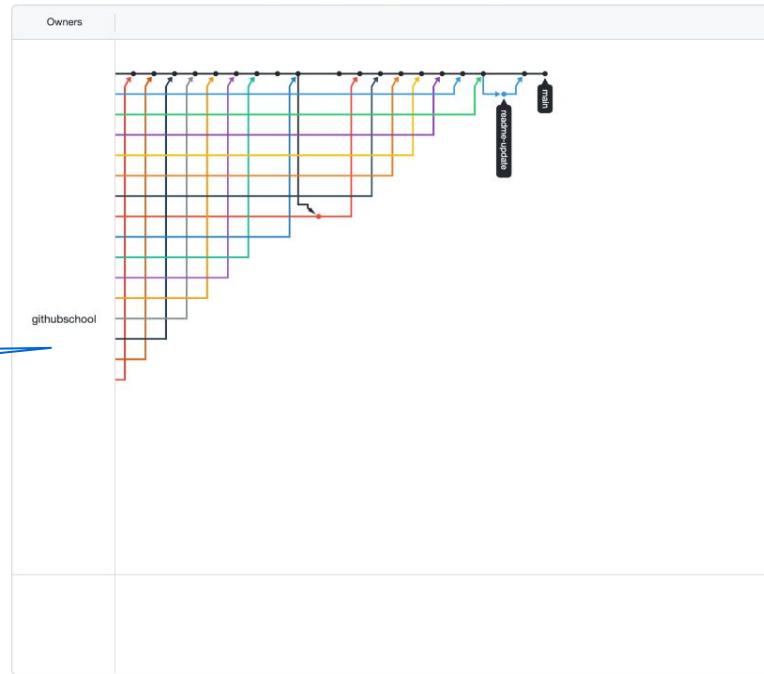
Network

Forks

People

## Network graph

Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.



Keyboard shortcuts available ⓘ



Search or jump to...

Pull requests Issues Marketplace Explore

Logout + Add user

githubschool / open-door

Watch 1 Star 0 Fork 0

Code Issues 5 Pull requests Discussions Actions Projects 1 Wiki Security Insights Settings

Settings

## Options

Manage access

Security &amp; analysis

Branches

Webhooks

Notifications

Integrations

Deploy keys

Autolink references

Actions

Environments

Secrets

Pages

Moderation settings

## Settings

## Repository name

open-door

Rename

 Template repositoryTemplate repositories let users generate new repositories with the same directory structure and files. [Learn more.](#)

## Social preview

Upload an image to customize your repository's social media preview.

Images should be at least 640x320px (1280x640px for best display).

[Download template](#)

GitHub Pages

[Edit](#)

Repo settings

## Features

 Wikis Restrict editing to users in teams with push access only

Public wikis will still be readable by everyone.

 Issues

Issues integrate lightweight task tracking into your repository. Keep projects on track with issue labels and milestones, and reference them in commit messages.

## Get organized with issue templates

Give contributors issue templates that help you cut through the noise and help them push your project forward.

Set up templates

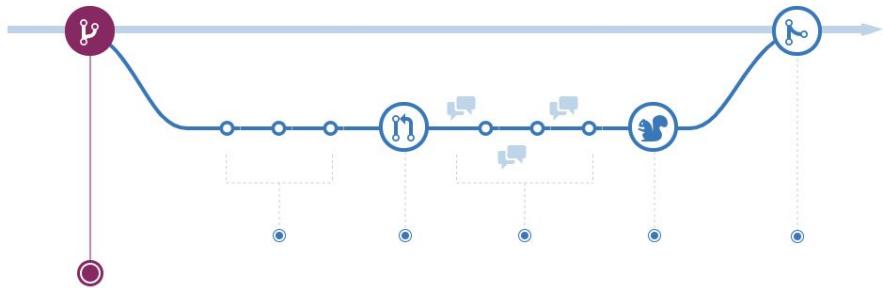
## **Activity: comment on the issue**

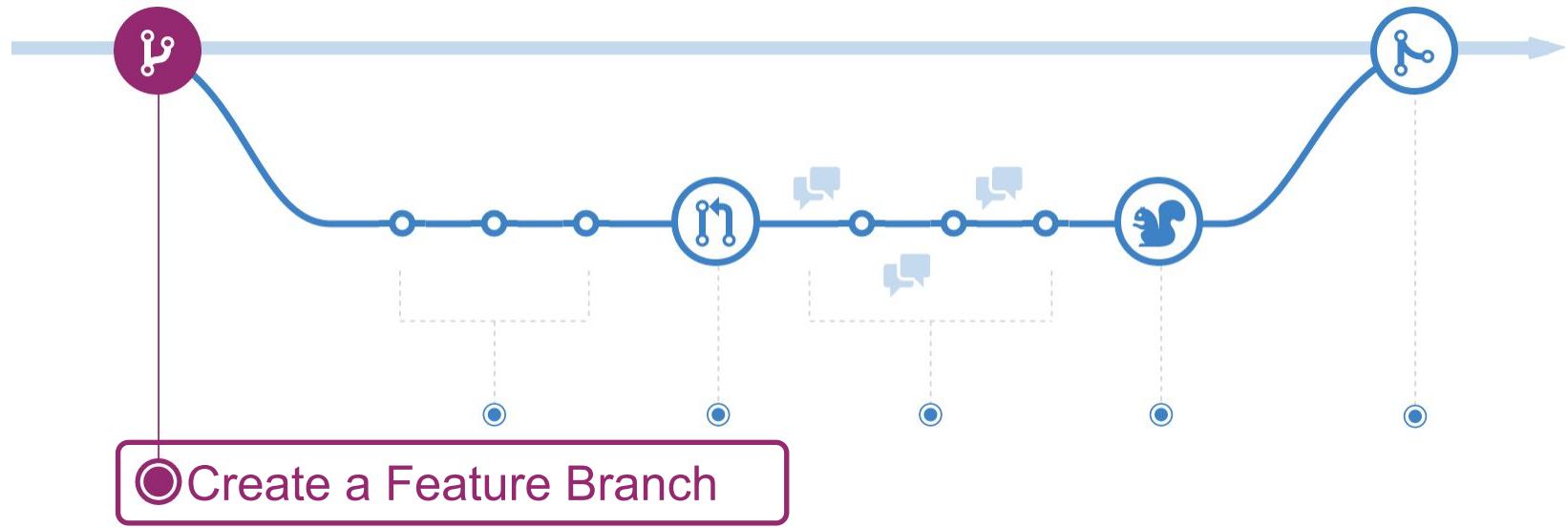
1. Navigate to the issues tab, and click on the issue entitled “Comment here to be added as a collaborator”
2. Leave a comment

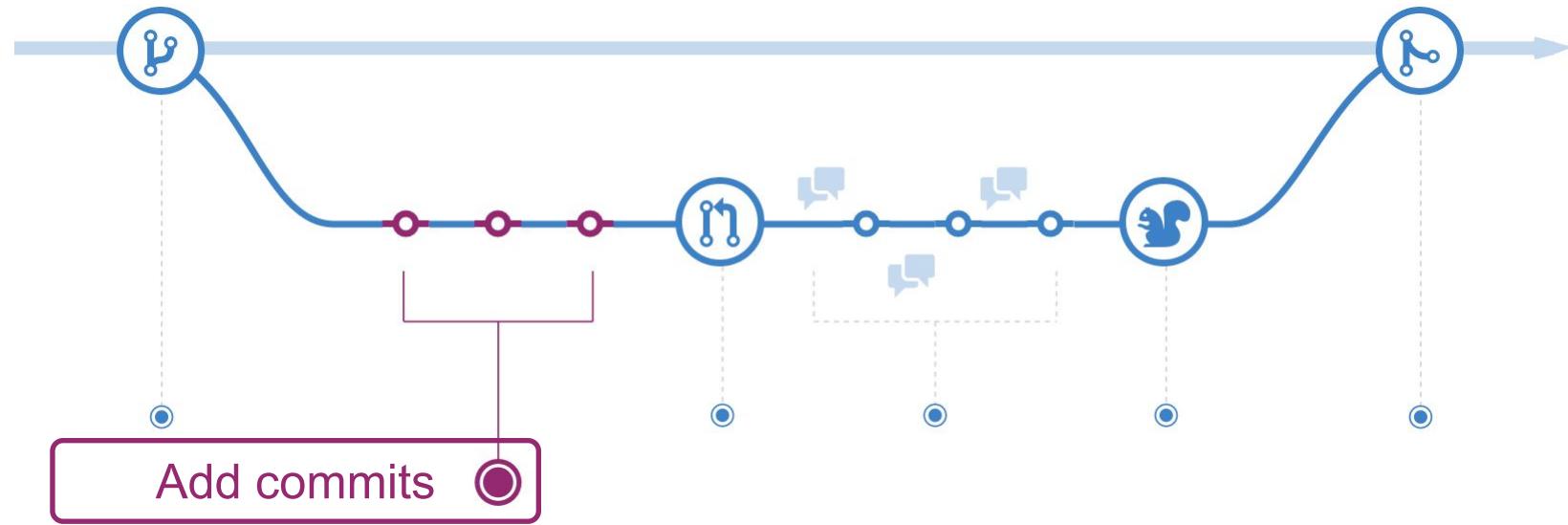
# Branching Strategy

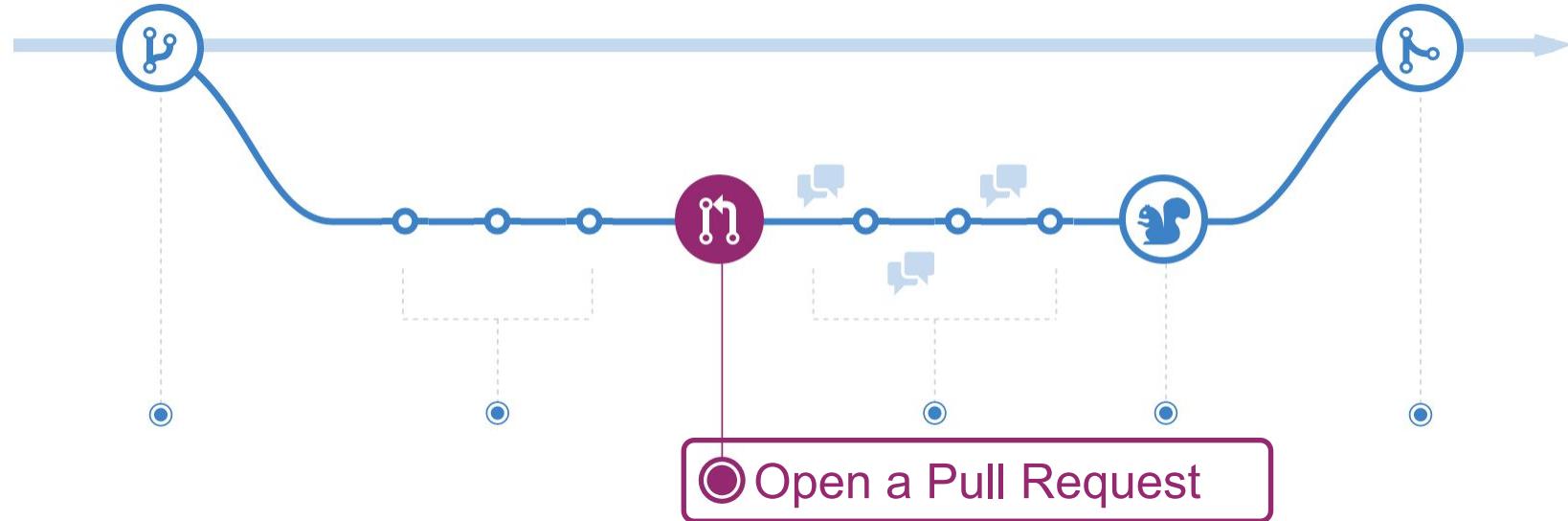
GitHub Flow

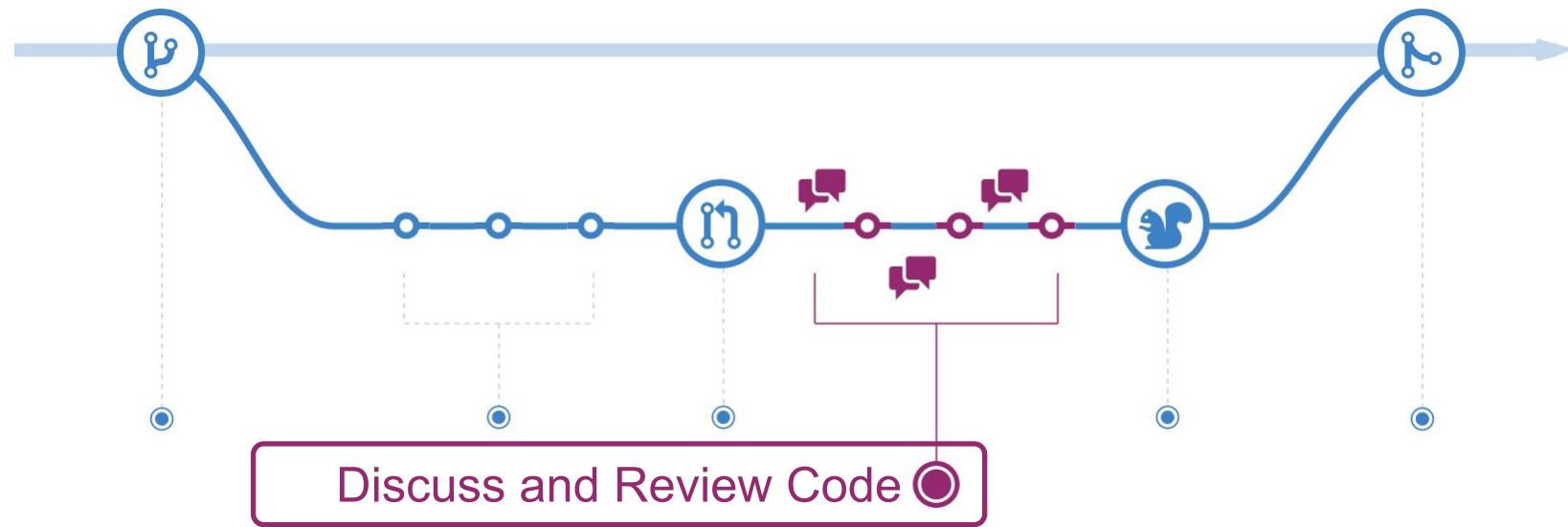
# The GitHub Flow

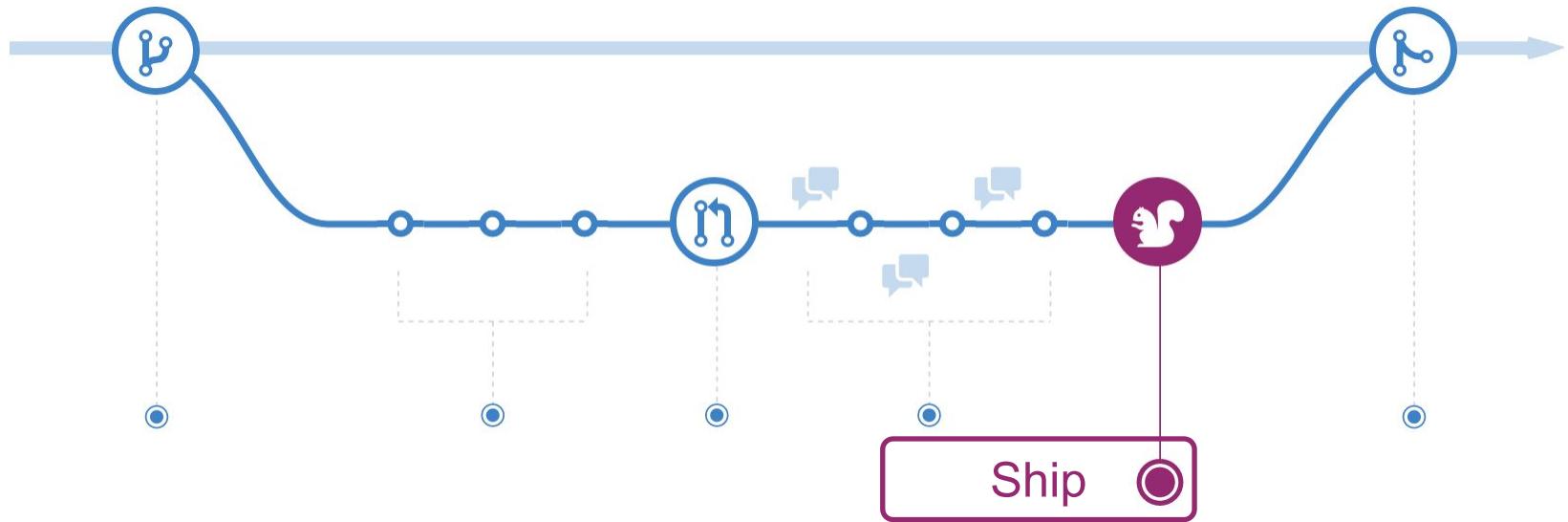


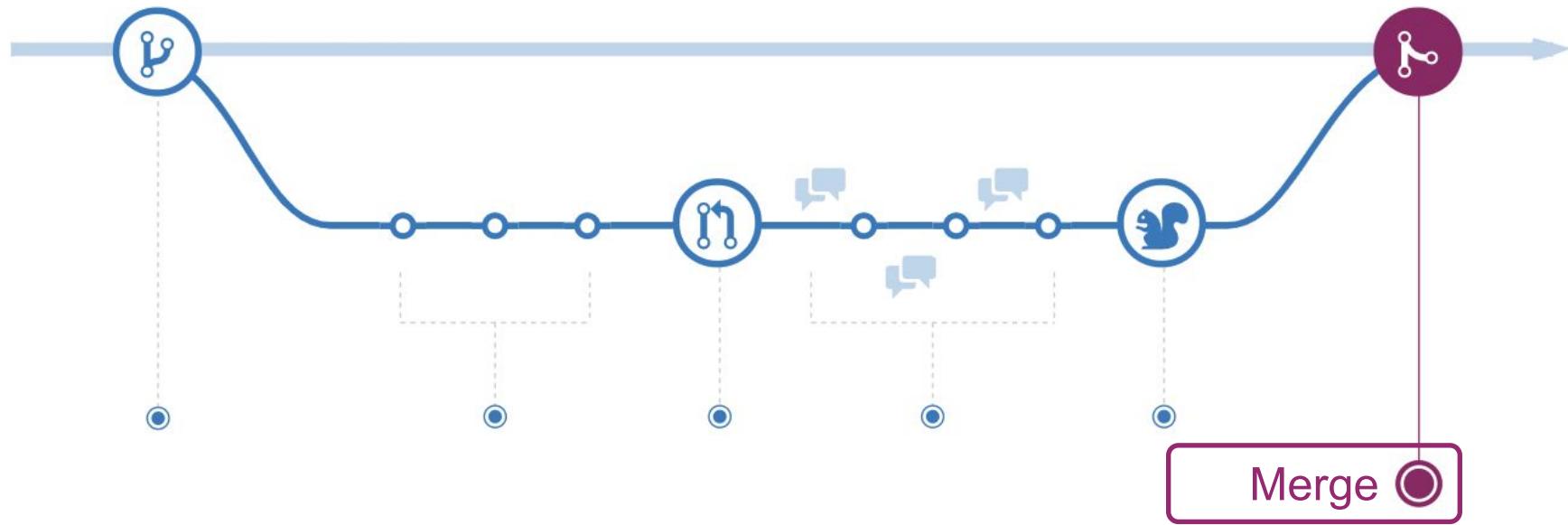












# Project 1: Caption this

- Branching with Git
- Collaborating on code
- Local history
- Local Git configs
- Editing on GitHub
- Streamlining workflows with aliases
- Working locally
- Merging Pull Requests



Demo

# Activity: creating a branch with GitHub

1. Navigate to Code tab of the class repository.
2. Click the *branch dropdown*.
3. Enter the branch name ‘add-caption-YOUR-USERNAME’.
4. Press **Enter**

# Project 1: Caption this

- Branching with Git
- Collaborating on code
- Local history
- Local Git configs
- Editing on GitHub
- Streamlining workflows with aliases
- Working locally
- Merging Pull Requests
-

# Local Git configs



--system



--global



--local



Demo

# **Activity: configuring username and email**

## **Configuring Your User Name and Email**

```
git config --global user.name  
"First Last"
```

```
git config --global user.email  
"you@email.com"
```

## **Viewing Your Configurations**

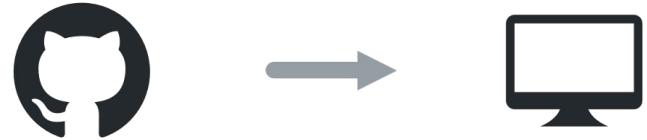
```
git config --list
```

```
git config --global --list
```

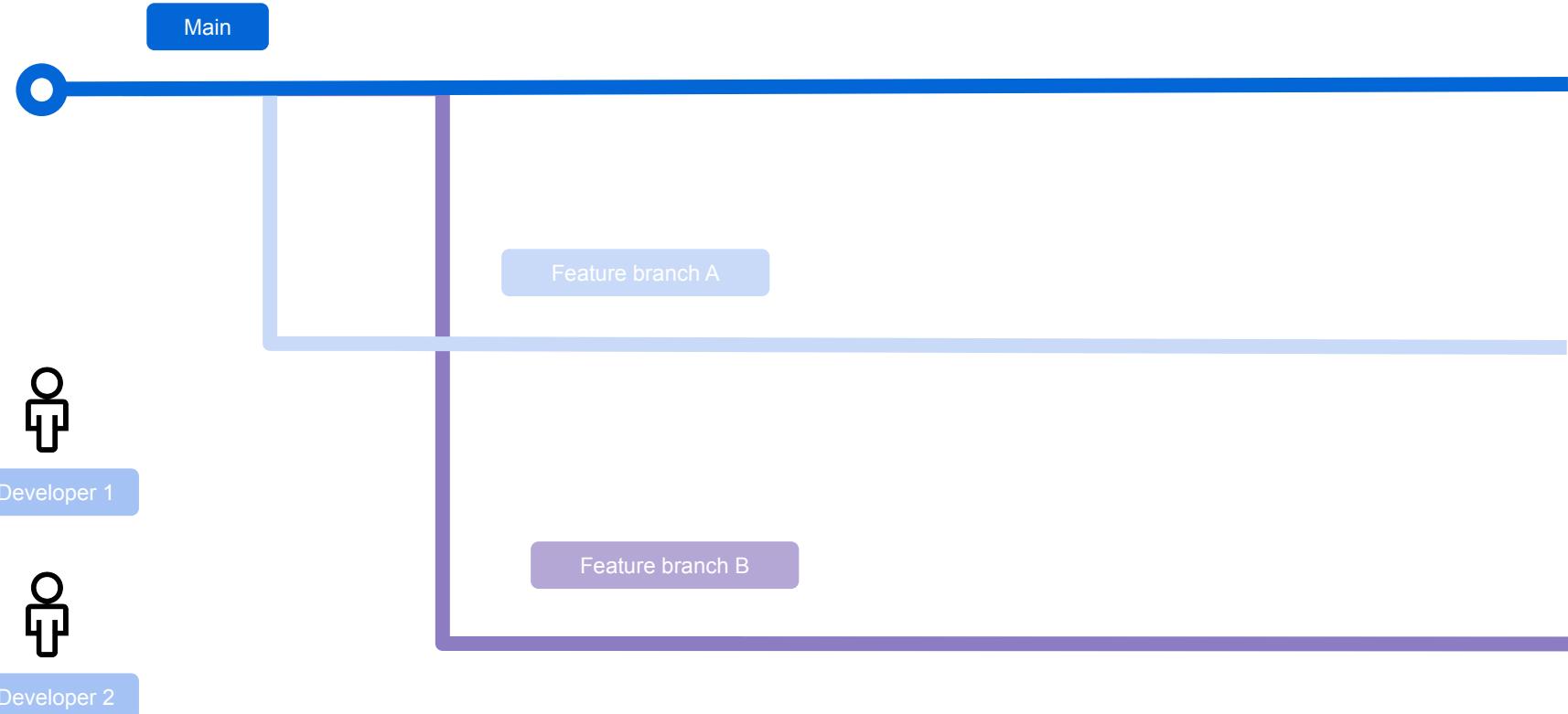
# Project 1: Caption this

- Branching with Git
- Collaborating on code
- Local history
- Local Git configs
- Editing on GitHub
- Streamlining workflows with aliases
- Working locally
- Merging Pull Requests

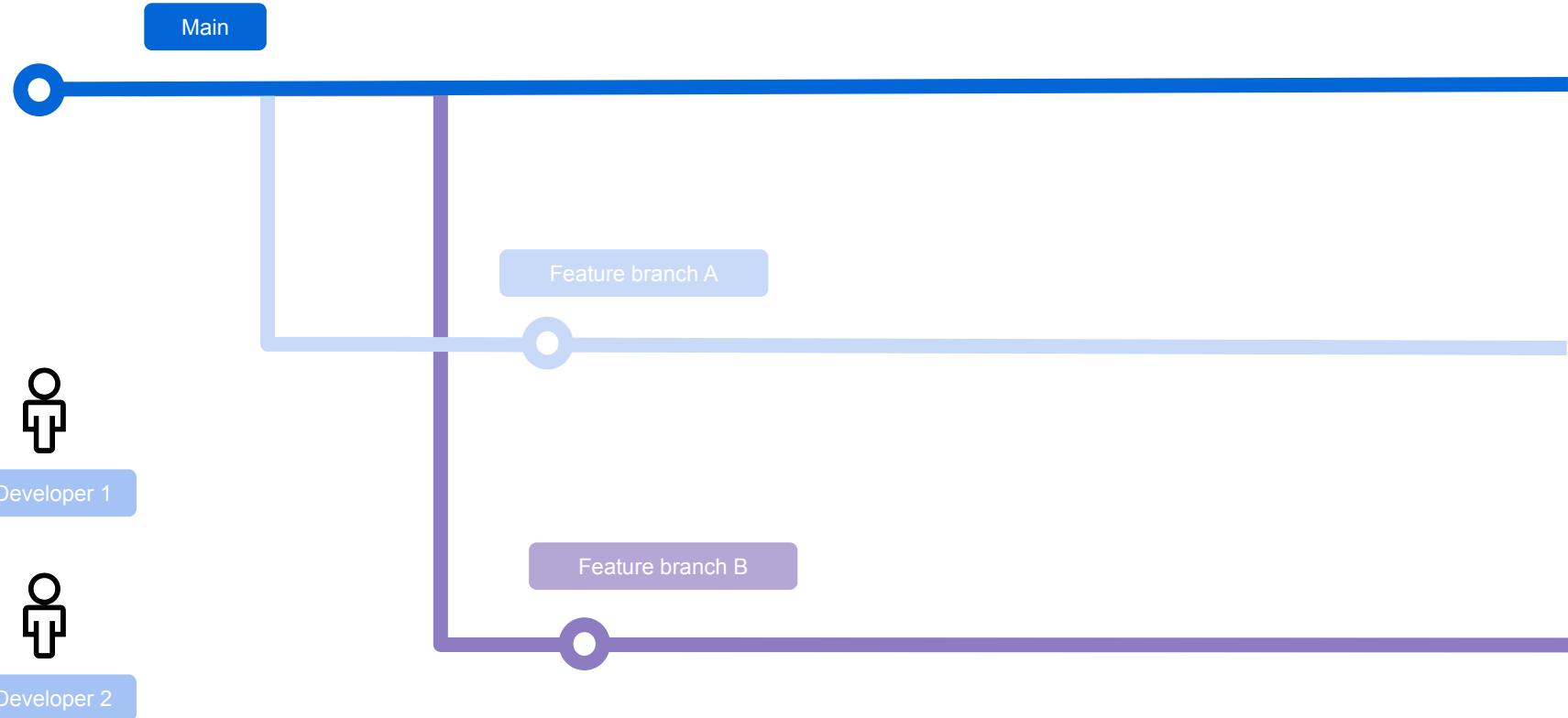
# Working locally



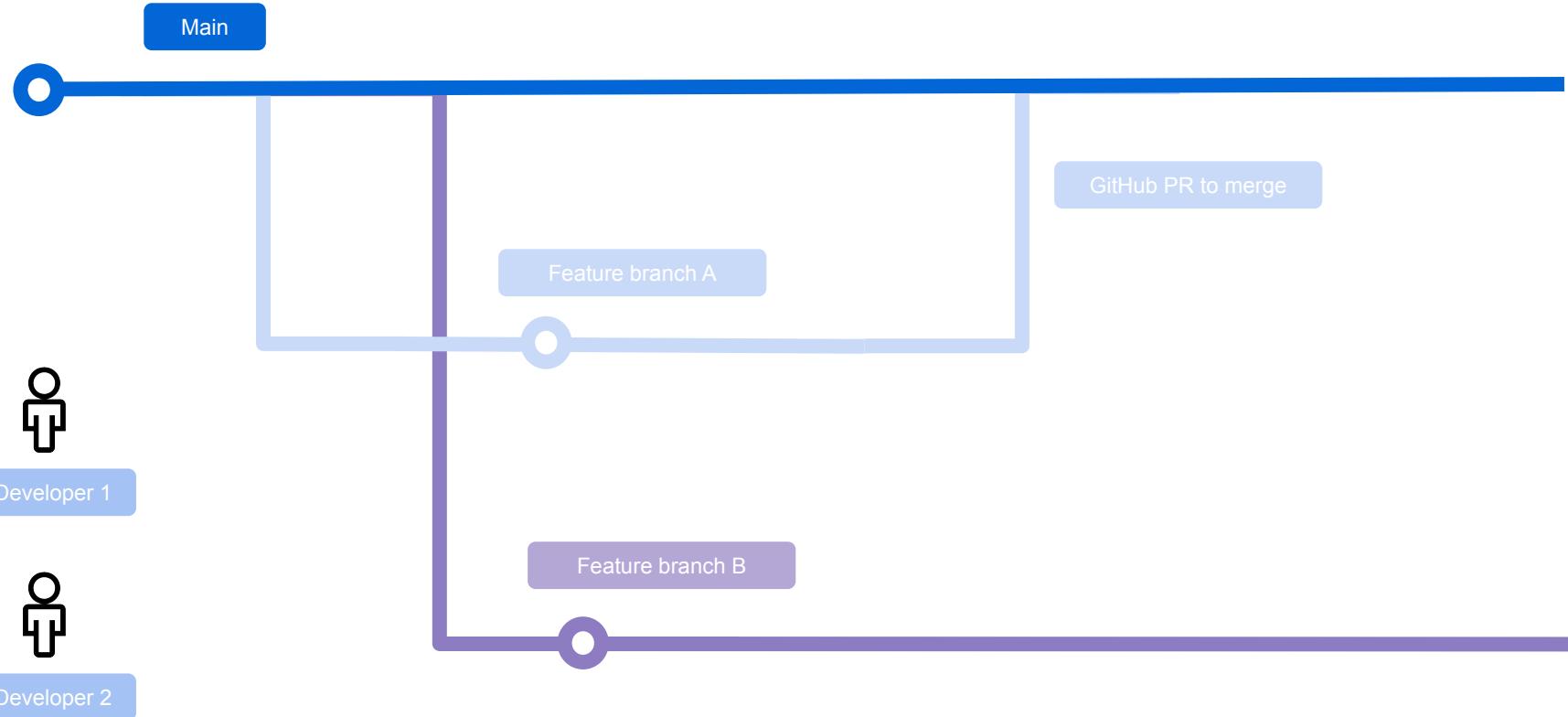
# Branching with Git



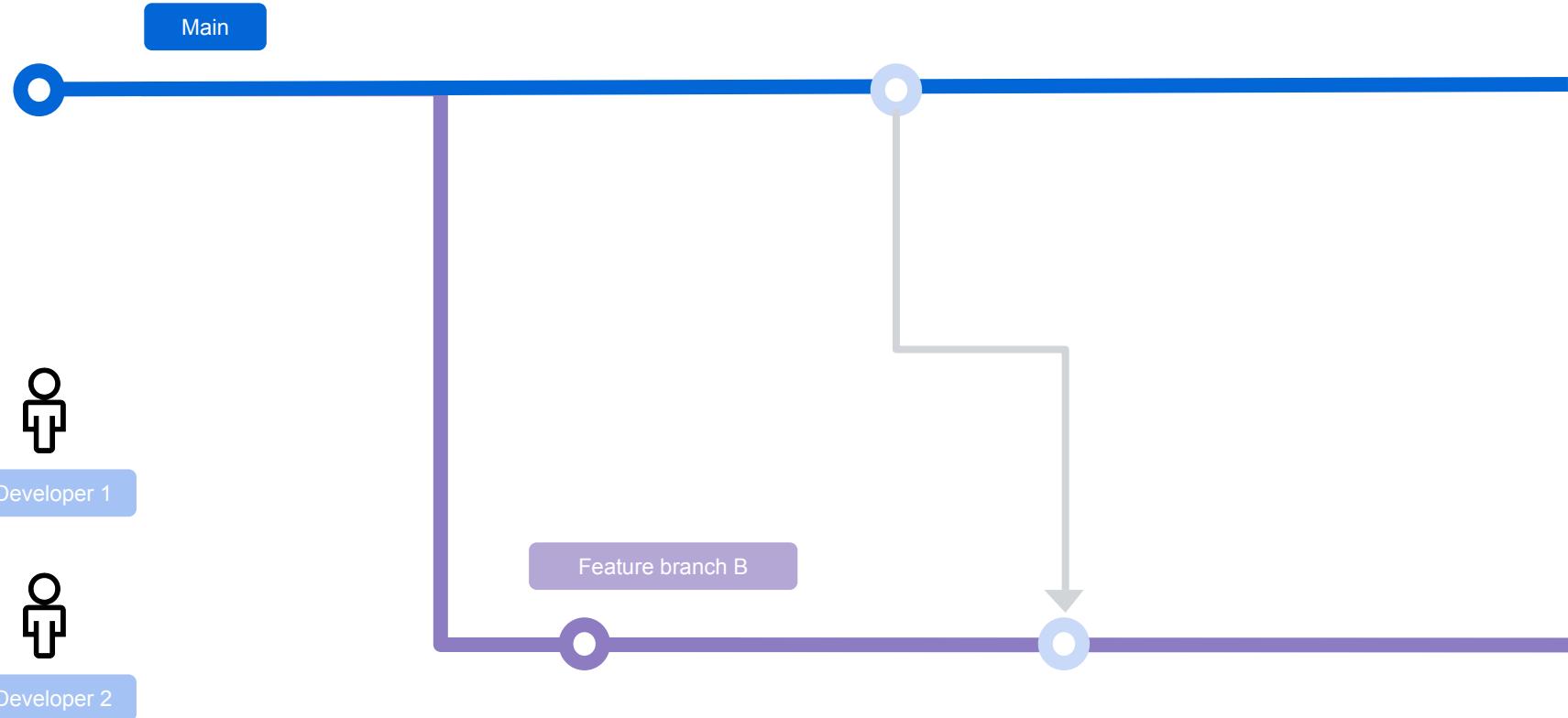
# Branching with Git



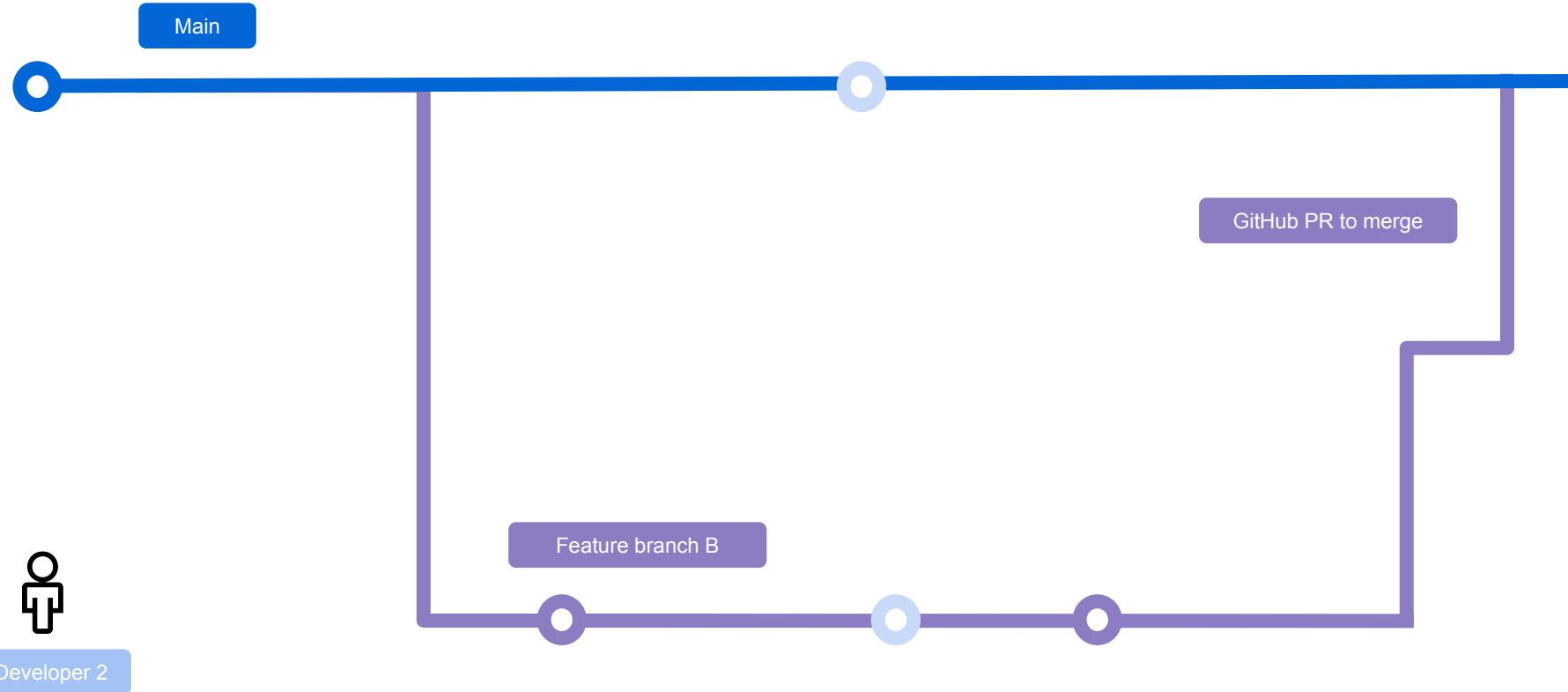
# Branching with Git



# Branching with Git



# Branching with Git



# Branching with Git





Demo

# Activity: create a local copy of the repo

1. Navigate to the Code tab of the class repository on GitHub.
2. Click Clone or download.
3. Copy the clone URL to your clipboard.
4. Open your command line application.
5. Retrieve a full copy of the repository from GitHub: `git clone <CLONE-URL>`
6. Once the clone is complete, cd into the new directory created by the clone operation: `cd <REPOSITORY-NAME>`

# Activity: Switching branches

1. `git branch`
2. `git branch --all`
3. `git branch -a`
4. `git switch <BRANCH-NAME>`

# The two stage commit



working



staging



history

# The working tree



working



staging



history

new



modified



# Staging area



working



staging



history

new



modified

# The commit



working



staging



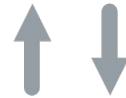
history

new

modified



# Pushing changes to Github



Push    Pull





Demo

## Activity: edit your file

1. Find your file, e.g **05-USERNAME .md**.
2. Open your file in your favorite text editor.
3. On line 6, replace the text with your caption.
4. Save your file.

# Activity: make a commit

1. `git status`
2. `git add my-file.md`
3. `git status`
4. `git commit -m "updating  
caption"`
5. `git status`

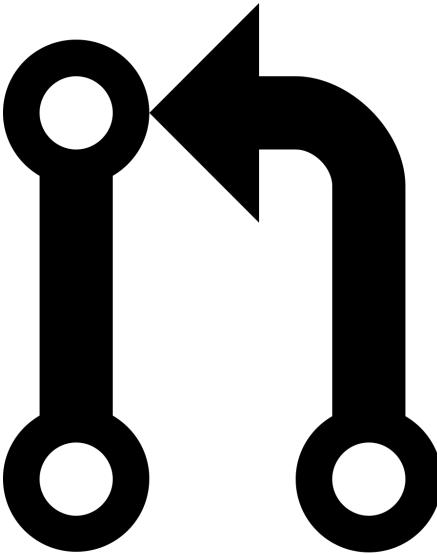
# Activity: push to GitHub

1. `git push`

# Project 1: Caption this

- Branching with Git
- Collaborating on code
- Local history
- Local Git configs
- Editing on GitHub
- Streamlining workflows with aliases
- Working locally
- Merging Pull Requests

# Creating a PR





Search or jump to...

Pull requests Issues Marketplace Explore

Alert +

githubschool / open-door

Watch 1

Star 0

Fork 0

Code Issues 5 Pull requests 11 Discussions Actions Projects 1 Wiki Security Insights Settings

#### Label issues and pull requests for new contributors

Now, GitHub will help potential first-time contributors discover issues labeled with [good first issue](#)

Dismiss

Filters ▾

Labels 10

Milestones 2

New pull request

11 0 Open  27 Closed

Author ▾ Label ▾ Projects ▾ Milestones ▾ Reviews ▾ Assignee ▾ Sort ▾

There aren't any open pull requests.

You could search [all of GitHub](#) or try an [advanced search](#).

New PR

💡 ProTip! Adding [no:label](#) will show everything without a label.





## Update 02-gusshawstewart.md #44

[Edit](#) [Code](#)

PR navigation bar



gusshawstewart commented now

Member



...

TL;DR This PR updates the copy of gusshawstewart.md because it was out of date

This PR closes / fixes / resolves #1

PR timeline

### Done

- made a commit
- created a branch

### To do

- merge the branch
- delete the branch afterwards

cc. @ppremk

→ Update 02-gusshawstewart.md

Verified



2764944

→ gusshawstewart added the documentation label now

→ gusshawstewart added this to the new-milestone milestone now

→ gusshawstewart requested a review from althedoll now

+3 -1



### Reviewers

althedoll

At least 1 approving review is required to merge this pull request.

Still in progress? Convert to draft

### Assignees

gusshawstewart

### Labels

documentation

### Projects

Class Activity: Caption Contest

Awaiting triage ▾

### Milestone

new-milestone

### Linked issues

Successfully merging this pull request may close these issues.

Single sign-on to see issues within the github



Search or jump to...

Pull requests Issues Marketplace Explore

4 + 1

githubschool / open-door

Watch 1

Star 0

Fork 0

Code

Issues 5

Pull requests 1

Discussions

Actions

Projects 1

Wiki

Security

Insights

Settings

## Update 02-gusshawstewart.md #44

Edit

Code

Open

gusshawstewart wants to merge 1 commit into `main` from `gusshawstewart.md-update`

Conversation 0

Commits 1

Checks 1

Files changed 1

+3 -1

PR commits

Commits on Sep 2, 2021

Update 02-gusshawstewart.md

 gusshawstewart committed 7 minutes ago ✓

Verified



2764944



© 2021 GitHub, Inc.

Terms

Privacy Security Status Docs

Site admin mode off



Contact GitHub

Pricing

API

Training

Blog

About



Search or jump to...

Pull requests Issues Marketplace Explore



githubschool / open-door

Watch 1

Star 0

Fork 0

Code Issues 5 Pull requests 1 Discussions Actions Projects 1 Wiki Security Insights Settings

## Update 02-gusshawstewart.md #44

Edit &lt;&gt; Code

Open gusshawstewart wants to merge 1 commit into `main` from `gusshawstewart.md-update`

Conversation 0 Commits 1 Checks 1 Files changed 1 +3 -1

✓ Update 02-gusshawstewart.md 2764944 ▾

Re-run jobs ▾

Test action  
on: pull\_request

✓ build

build

PR checks

build

- succeeded 21 minutes ago in 3s
- >  Set up job 25s
  - >  Run actions/checkout@v2 1s
  - >  Run a one-line script 0s
  - >  Run a multi-line script 0s
  - >  Post Run actions/checkout@v2 0s
  - >  Complete job 0s

Search logs

...

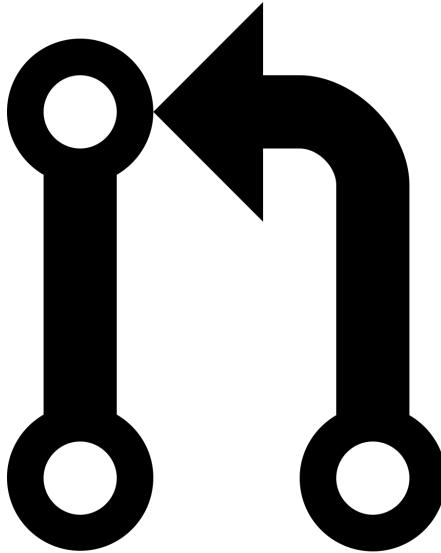


Demo

# Activity: create a Pull Request

1. Click the **Pull Request** tab.
2. Click **New Pull Request**.
3. In the **base** dropdown, choose **main**
4. In the **compare** dropdown, choose **your branch**.
5. Type a subject line and enter a comment.
6. Use markdown formatting to add a header and a checklist to your Pull Request.
7. Include one of the keywords: closes, fixes, or resolves followed by the issue number you created earlier to note which Issue the Pull Request should close. Example: This resolves #3
8. Click **Preview** to see how your Pull Request will look.
9. Assign the Pull Request to yourself.
10. Select your partner as a Reviewer for the Pull Request.
11. Click *Create pull request*.

# Reviewing a PR





Search or jump to...

Pull requests Issues Marketplace Explore

+

githubschool / open-door

Watch 1

Star 0

Fork 0



Code



5



1



1



## Update 02-gusshawstewart.md #44

Edit

Code



gusshawstewart wants to merge 1 commit into main from gusshawstewart.md-update



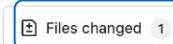
0



1



1



1



Changes from all commits

File filter

Conversations

Jump to



0 / 1 files viewed

Review changes

Click to give  
line comment

Line  
comment and  
suggestion

The screenshot shows a GitHub pull request interface for a file named `_slides/02-gusshawstewart.md`. The pull request has 1 commit and 1 check. The commit message is `title: "Welcome to our slide deck!"`. The check status is `+3 -1`.

A blue callout box points to a line of code at line 6: `- Scotland FC are the best team in the world!!!! not really... I agree`. A blue box highlights this line.

A second blue callout box points to the `Start a review` button in the comment modal, which contains the following text:

```
This looks good 😊, but have you thought about doing it like this:  
```suggestion  
This is a change ABC  
...  
Attach files by dragging & dropping, selecting or pasting them.
```

The modal also includes `Cancel`, `Add single comment`, and `Start a review` buttons.

The file content is as follows:

```
1  @@ -3,7 +3,9 @@ layout: slide  
2  title: "Welcome to our slide deck!"  
3  ---  
4  ---  
5  ---  
6  - Scotland FC are the best team in the world!!!! not really... I agree  
6 + This is a change  
7 +  
8 + This is a second change  
7 9  
8 10  ![constructocat](https://octodex.github.com/images/constructocat2.jpg)  
9 11  {:: .center}
```



Search or jump to...



 [githubschool / open-door](#)

 Watch ▾ 1  Star 0  Fork 0

<> Code Issues 5 Pull requests 1 Discussions Actions Projects 1 Wiki Security Insights

Update 02-gusshawstewart.md #44

Edit Open with ▾

Open gusshawstewart wants to merge 1 commit into main from gusshawstewart.md-upda

0 / 1 files viewed

**Review changes** ▾

**Finish your review**

Write Preview

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

- Comment**  
Submit general feedback without explicit approval.

- Approve**  
Submit feedback and approve merging these changes.
- Request changes**  
Submit feedback that must be addressed before merging.

[Submit review](#)

---

Contact GitHub      Pricing      API      Training      Blog      About

© 2021 GitHub, Inc.

## Terms

Privacy

Security

Status

DOCS

Approve  
PR

# Finish the review



Demo

# Activity: review a Pull Request

1. Click the **Pull Request** tab.
2. Review the PR directly **below** your own
3. Click the '**Files Changed**' tab.
4. Hover over a single line in the file to see the blue +. Click the + to add a line comment.
5. Comment on the line and click Start a review.
6. Click Review changes in the top right corner.
7. 'Approve' their changes
8. Enter a general comment for the review.
9. Click Submit review

# Project 1: Caption this

- Branching with Git
- Collaborating on code
- Local history
- Local Git configs
- Editing on GitHub
- Streamlining workflows with aliases
- Working locally
- Merging Pull Requests
-

# Editing on GitHub

A screenshot of a GitHub pull request interface. The title of the PR is "02-gusshawstewart.md #45". The description states: "gusshawstewart wants to merge 1 commit into main from gusshawstewart-file-update". The commit details show a single file change: "slides/02-gusshawstewart.md". The commit message is: "scotland FC are the best team in the world!!!! not really... I agree". The commit has two changes: "Change A" and "Change B". The code diff shows: "title: "Welcome to our slide deck!"", "scotland FC are the best team in the world!!!! not really... I agree", "Change A", "Change B", and "![constructocat](https://octodex.github.com/images/constructocat2.jpg)". A blue callout box with the text "Edit file" points to the "Edit file" button in the context menu on the right side of the code editor.

02-gusshawstewart.md #45

gusshawstewart wants to merge 1 commit into [main](#) from [gusshawstewart-file-update](#)

on 0 Commits 1 Checks 1 Files changed 1 0 / 1 files viewed Review changes

commits File filter Conversations Jump to ...

slides/02-gusshawstewart.md

-3,7 +3,9 @@ layout: slide

title: "Welcome to our slide deck!"

scotland FC are the best team in the world!!!! not really... I agree

Change A

Change B

![constructocat](https://octodex.github.com/images/constructocat2.jpg)

[: .center}

Show comments

View file

Edit file

Delete file

Open in desktop

Edit file

Contact GitHub Pricing API Training Blog About

# Commit suggested changes

• created a branch

**To do**

merge the branch  
 delete the branch afterwards

cc. @ppremk

-o-  Update 02-gusshawstewart.md Verified ✓ c407342

 **guy-man** approved these changes now [View changes](#)

```
_slides/02-gusshawstewart.md
...
3   3     @@ -3,7 +3,9 @@
4   4       title: "Welcome to our slide deck!"
5   5
6 - Scotland FC are the best team in the world!!!! not really... I agree
6 + Change A
```

 **guy-man** now Collaborator ...

Suggested change ⓘ

```
6 - Change A
6 + Change ABCDEFG
```

[Commit suggestion ▾](#) [Add suggestion to batch](#)

 Reply...

[Resolve conversation](#)

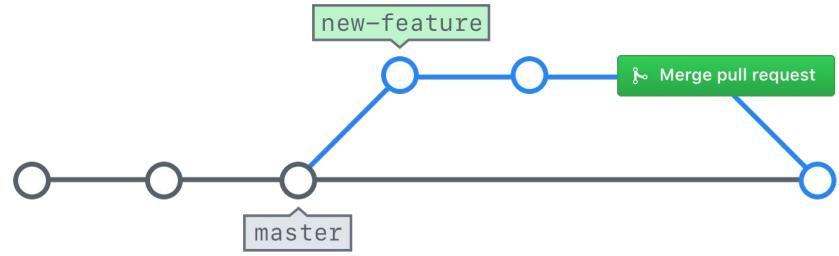


Demo

# Project 1: Caption this

- Branching with Git
- Collaborating on code
- Local history
- Local Git configs
- Editing on GitHub
- Streamlining workflows with aliases
- Working locally
- Merging Pull Requests

# Merging a PR



# Merging on GitHub

The screenshot shows a GitHub pull request interface with several annotations:

- A blue rounded rectangle labeled "Changes approved" is positioned at the top right, pointing to the green "Changes approved" status bar.
- A blue rounded rectangle labeled "Merge PR" is positioned at the bottom center, pointing to the "Merge pull request" button.
- A blue line connects the "Changes approved" annotation to the "Changes approved" status bar.
- A blue line connects the "Merge PR" annotation to the "Merge pull request" button.
- The main interface includes:
  - A header: "Changes approved" with a note: "Add more commits by pushing to the `gusshawstewart-file-update` branch on `githubschool/open-door`".
  - A section: "This branch has not been deployed" with "No deployments".
  - A summary: "Changes approved" (1 approving review by reviewers with write access).
    - "Show all reviewers"
  - A status: "1 approval".
  - A status: "All checks have passed" (1 successful check).
    - "Show all checks"
  - A status: "This branch has no conflicts with the base branch" (Merging can be performed automatically).
  - A button: "Merge pull request" with a dropdown arrow.
  - A note below the button: "You can also open this in GitHub Desktop or view command line instructions."
  - A comment section with "Write" and "Preview" tabs.
  - A text input field: "Leave a comment".
  - A toolbar with icons: H, B, I, etc.

# Delete the branch on GitHub, and locally

A screenshot of a GitHub pull request interface. At the top, a purple banner indicates the pull request has been merged. Below it, a message from user 'guy-man' shows a 'Suggested change' with two items: '6 - Change A' (red background) and '6 + Change ABCDEFG' (green background). Buttons for 'Commit suggestion' and 'Add suggestion to batch' are at the bottom of this section. A reply input field and a 'Resolve conversation' button follow. The next section shows a merge commit from 'gusshawstewart' merging commit '3f18860' into 'main'. It includes 'View details' and 'Revert' buttons. A success message states 'Pull request successfully merged and closed' and 'You're all set—the gusshawstewart-fil... branch can be safely deleted.' with a 'Delete branch' button. A blue callout bubble points to this 'Delete branch' button with the text 'Delete the branch'.

Merged

Update 02-gusshawstewart.md #45  
gusshawstewart merged 1 commit into main from gusshawstewart-file-update now

guy-man 5 minutes ago Collaborator

Suggested change ⓘ

6 - Change A

6 + Change ABCDEFG

Commit suggestion Add suggestion to batch

Reply...

Resolve conversation

gusshawstewart merged commit 3f18860 into main now  
1 check passed

View details Revert

Pull request successfully merged and closed  
You're all set—the gusshawstewart-fil... branch can be safely deleted.

Delete branch

Write Preview

Leave a comment

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

Delete the branch



Demo

## Activity: Merge Pull Request

1. Navigate to your Pull Request (HINT: Use the Author or Assignee drop downs to find your Pull Request quickly)
2. Click *Conversation*
3. Scroll to the bottom of the Pull Request and click the *Merge pull request* button
4. Click *Confirm merge*
5. Click *Delete branch*

### Update local repo

1. Start by switching back to your default branch: `git switch main`
2. Retrieve all of the changes from GitHub: `git pull`

# Activity: Cleaning Up the Unneeded Branches

1. `git branch --all`
2. `git branch --merged`
3. `git branch -d <branch-name>`
4. `git branch --all`
5. `git pull --prune`
6. `git config --global`  
`fetch.prune true`

# Project 1: Caption this

- Branching with Git
- Collaborating on code
- Local history
- Local Git configs
- Editing on GitHub
- Streamlining workflows with aliases
- Working locally
- Merging Pull Requests

# Local history

```
→ high-revs git:(master) git log --oneline
bae9b9e (HEAD -> master, origin/master, origin/HEAD) Merge branch 'master' into nfr-enhancement
2ccd1c2 Merge branch 'master' into nfr-enhancement
9924622 Merge pull request #13 from githubschool/add-caption
7ef2f6f Merge pull request #12 from githubschool/add-caption
d64fb9e Merge pull request #22 from githubschool/add-caption
794c7d7 Added caption.
89b004c Merge pull request #21 from githubschool/add-caption
1164e01 Merge pull request #14 from githubschool/add-caption
234a546 Merge pull request #15 from githubschool/add-caption
405e588 Merge pull request #20 from githubschool/add-caption
127fae2 Merge pull request #25 from githubschool/add-caption
07be9e4 Merge pull request #18 from githubschool/add-caption
0316df7 Merge pull request #19 from githubschool/add-caption
756c4b4 Merge pull request #10 from githubschool/add-caption
ecd6130 Merge pull request #23 from githubschool/add-caption
7f3a1c8 Merge pull request #9 from githubschool/add-caption
d935717 Merge pull request #16 from githubschool/add-caption
1d09278 Merge pull request #24 from githubschool/add-caption
7704793 Merge pull request #11 from githubschool/add-caption
7c02414 Merge pull request #17 from githubschool/add-caption
21c4855 Update _slides/09-rd-boris-mechkov.md
947ad2e updates caption text
14bb5b0 Added my enhancement
751b3d5 updates text
4733d10 Update caption
f0a79d1 Add caption gif
e7b1787 update slide caption
aa81119 updates caption text
e446280 Added caption for Carsten
d9cbc7c Update caption
ae77cf2 update text of slide
e3fd454 Update text of slide 11
```



Demo

## Activity: using Git Log

- `git log`
- `git log --oneline`
- `git log --oneline --graph`
- `git log --oneline --graph  
--decorate`
- `git log --oneline --graph  
--decorate --all`

# Project 1: Caption this

- Branching with Git
- Collaborating on code
- Local history
- Local Git configs
- Editing on GitHub
- Streamlining workflows with aliases
- Working locally
- Merging Pull Requests
-

# Streamlining workflows with aliases

```
alias.s=status -s
alias.ch=checkout
alias.cob=checkout -b
alias.ec=config --global -e
alias.lol=log --oneline --graph
alias.bclean=!f() { branches=$(git branch --merge
}); [ -z "$branches" ] || git branch -d $branch
```



Demo

# Activity: creating custom aliases

## Original Command

```
git log --oneline --graph  
--decorate --all
```

## Creating the Alias

```
git config --global alias.lol  
"log --oneline --graph --decorate  
--all"
```

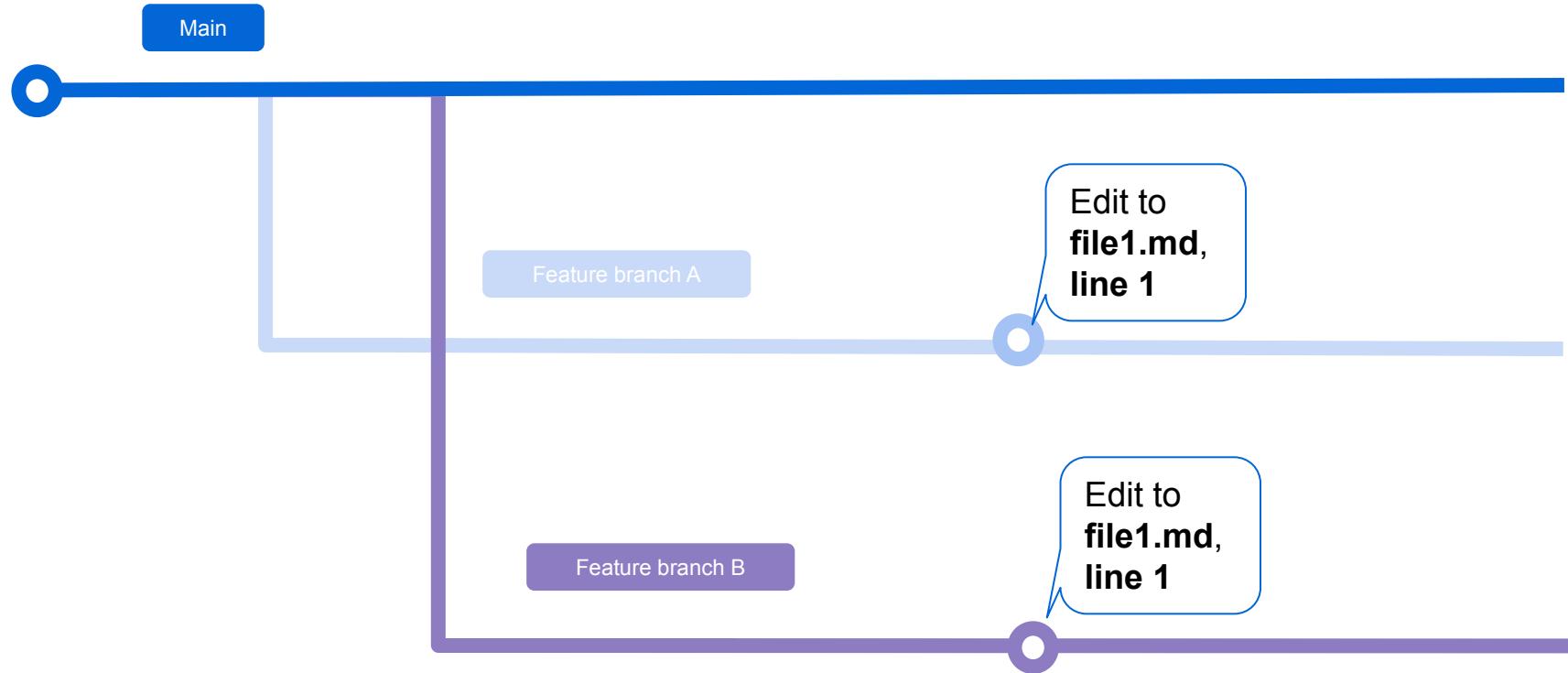
## Using the Alias

```
git lol
```

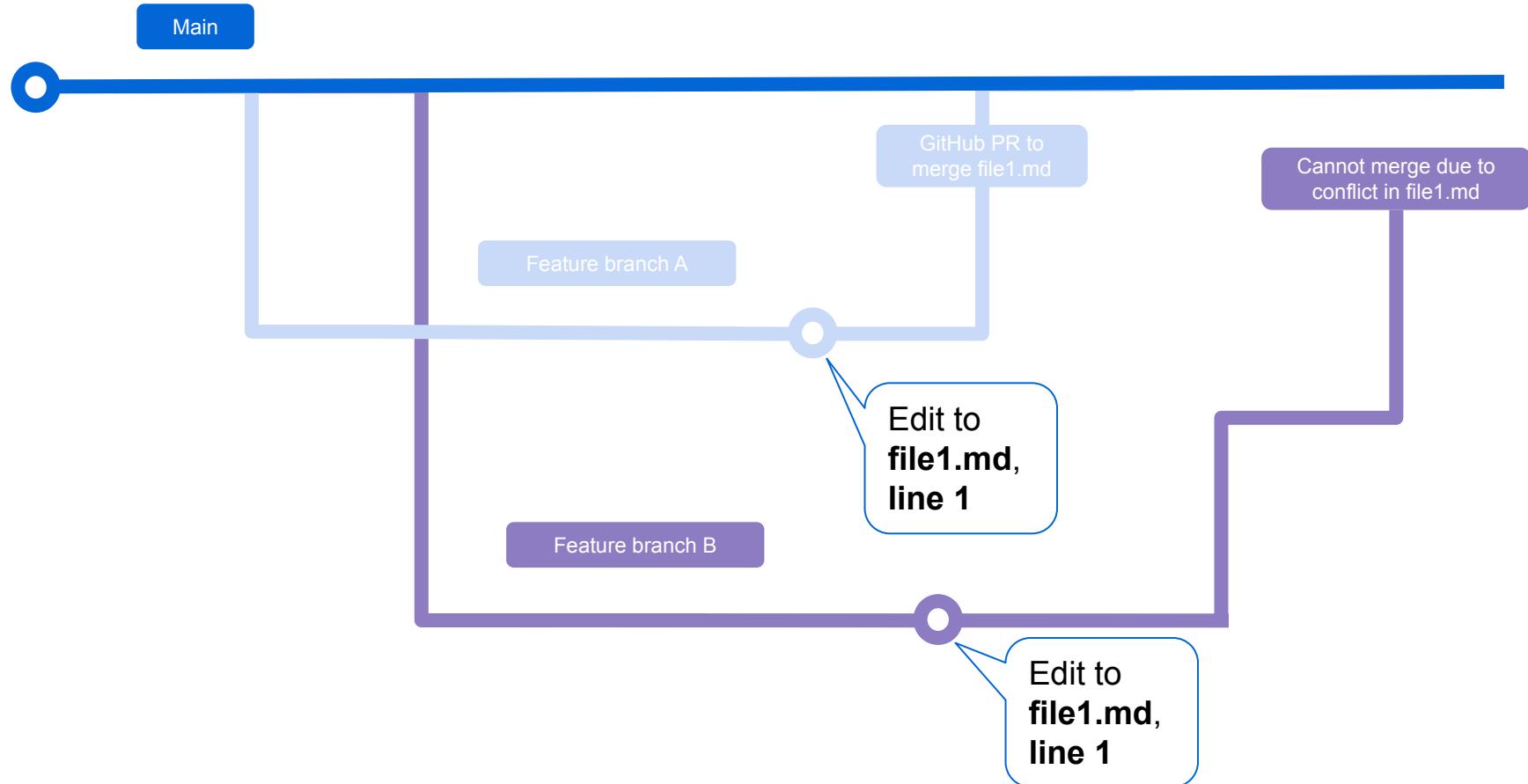
# Project 2: Merge conflicts

Defining a merge conflict • Resolving a merge conflict

# Defining a merge conflict



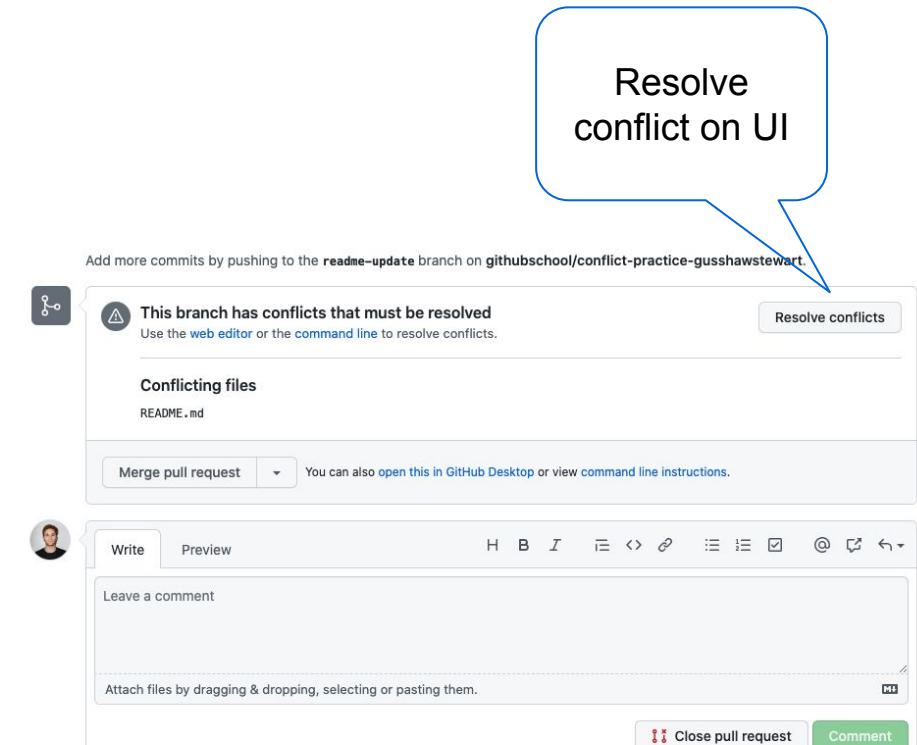
# Defining a merge conflict



# Project 2: Merge conflicts

Defining a merge conflict • Resolving a merge conflict

# Fixing a merge conflict



A screenshot of a GitHub pull request page. At the top, a blue callout bubble says "Resolve conflict on UI". Below it, a message says "Add more commits by pushing to the `readme-update` branch on `githubschool/conflict-practice-gusshawstewart`". A warning icon says "This branch has conflicts that must be resolved. Use the [web editor](#) or the [command line](#) to resolve conflicts." A "Resolve conflicts" button is on the right. The main area shows a conflict in `README.md`. A "Merge pull request" dropdown menu is open, showing options like "Merge pull request" and "You can also open this in GitHub Desktop or view command line instructions". Below the conflict, there's a comment section with a "Write" tab, a rich text editor toolbar, and a "Comment" button.

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

ⓘ ProTip! Add `.patch` or `.diff` to the end of URLs for Git's plaintext views.

Conflict resolved in  
feature branch

## Update README #3

Resolving conflicts between `readme-update` and `main` and committing changes → `readme-update`

Mark as resolved

1 conflicting file

README.md  
README.md

README.md

1 conflict

Prev ^

Next ^



Mark as resolved

Merge conflict  
markers

```
1 
2
3 **[PLAY IT HERE (http://ncase.me/polygons)](http://ncase.me/polygons)**
4
5 ===
6
7 This playable post is [public domain (CC0)](http://creativecommons.org/publicdomain/zero/1.0).
8
9 <<<< readme-update
10 This repository also has some baked in merge conflicts for practice.
11 =====
12 This repository is also being used as a part of GitHub training, to practice Git and GitHub.
13 >>>> main
14
15 Please feel free to use this in your classrooms,
16 make video/picture/text adaptations,
17 or modify the source code!
18 Attribution is not mandatory, but super appreciated.
19 We are [Vihart](http://vihart.com/) and [Nicky Case](http://ncase.me/).
20
21 Show us how you're using and remixing Parable of the Polygons!
22 Tweet us at
23 [@vihartvihart](https://twitter.com/vihartvihart) and
24 [@ncasenmare](https://twitter.com/ncasenmare).
```



Demo

## **Activity: resolve a merge conflict**

1. Use the GitHub UI to resolve your conflict
2. Use your command line to resolve the third merge conflict

# Project 3: GitHub Games

- Workflow review
  - Protected Branches and CODEOWNERS
  - Git Bisect
- Reverting Commits
  - Helpful Git commands
  - Viewing local changes
- Tags and Releases
  - Branching strategy (Git flow)

# GitHub Pages

## GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

✓ Your site is published at <https://githubschool.github.io/open-door/>

### Source

Your GitHub Pages site is currently being built from the [main branch](#). [Learn more](#).

Branch: main ▾

/ (root) ▾

Save

### Theme Chooser

Select a theme to publish your site with a Jekyll theme. [Learn more](#).

Choose a theme

# Workflow review

github-games-gusshawstewart / README.md in main

Edit file Preview

```
1 ## Welcome to GitHub Games
2
3 A project based learning activity for people who are getting started with Git and GitHub.
4
5 You can play the game at: https://githubschool.github.io/github-games/
6
7 >>_*SUPPORTED BROWSERS*: Chrome, Firefox, Safari, Opera and IE9+_
8
9 This fun open source game was cloned from: https://github.com/jakesgordon/javascript-tetris
10
```

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



Demo

# Activity: updating the README.md

1. `git clone URL`
2. `git switch -c readme-update`
3. Edit the URL in the README.md.
4. Commit the changes to your branch.
5. `git push -u origin  
readme-update`
6. Create a Pull Request *in your repository*  
(base: `main`, compare:  
`readme-update`)
7. Merge your Pull Request.
8. Delete the branch on GitHub.
9. Update your local copy of the repository:  
`git pull --prune`

# Project 3: GitHub Games

Workflow review

- Protected Branches and CODEOWNERS
- Git Bisect

Reverting Commits

- Helpful Git commands
- Viewing local changes

Tags and Releases

- Branching strategy (Git flow)

# Protected Branches

## Branch protection rule

### Branch name pattern

main

### Protect matching branches

#### **Require pull request reviews before merging**

When enabled, all commits must be made to a non-protected branch and submitted via a pull request with the required number of approving reviews and no changes requested before it can be merged into a branch that matches this rule.

#### **Require status checks to pass before merging**

Choose which [status checks](#) must pass before branches can be merged into a branch that matches this rule. When enabled, commits must first be pushed to another branch, then merged or pushed directly to a branch that matches this rule after status checks have passed.

#### **Require conversation resolution before merging**

When enabled, all conversations on code must be resolved before a pull request can be merged into a branch that matches this rule. [Learn more](#).

#### **Require signed commits**

Commits pushed to matching branches must have verified signatures.

#### **Require linear history**

Prevent merge commits from being pushed to matching branches.

#### **Include administrators**

Enforce all configured restrictions above for administrators.

#### **Restrict who can push to matching branches**

Specify people, teams or apps allowed to push to matching branches. Required status checks will still prevent these people, teams and apps from merging if the checks fail.

Rules applied to everyone including administrators

# CODEOWNERS

github-games-gusshawstewart / CODEOWNERS in main

[Edit new file](#)

[Preview](#)

```
1 *          @global-owner1 @global-owner2
2 README.md    @global-owner1
3 *.js         @js-experts
4 docs/*       @docs-team
```



Demo

# Activity: Protected Branches

1. Select the Settings tab.
2. Select Branches from the menu on the left side of the screen.
3. Click the Add rule button next to Branch protection rules.
4. In the Apply rule to textbox type the name of the branch you would like to protect, for example, **main**.
5. Click the Create button.

# Activity: create a CODEOWNERS file

1. Go out to the Code tab of your repository.
2. Click the Create new file button.
3. In the Name your file... textbox enter **CODEOWNERS** (no extension necessary). You can add this to a `.github/` directory if desired by entering `.github/CODEOWNERS`.
4. On the first line, type `* @YOUR_USERNAME`
  - This means that you will be the default owner for everything in the repo, unless a later match takes preference.
5. On the next line, type `*.js @GITHUBTEACHER`
  - Order is important. The last matching pattern for a given change takes precedence.
6. Scroll down, and type a commit message into the Commit new file dialog box.
7. Click the Commit new file button to save your changes.
8. Now that you have created a CODEOWNERS file, go back to your branch protection settings and click the Edit button next to **main**.
9. Under Rule settings, select the option to Require pull request reviews before merging and Require review from Code Owners. Remember to click Save changes.

# Project 3: GitHub Games

Workflow review

- Protected Branches and CODEOWNERS
- Git Bisect

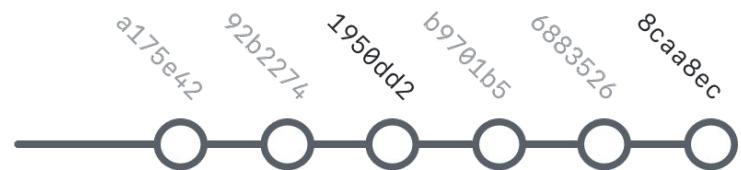
Reverting Commits

- Helpful Git commands
- Viewing local changes

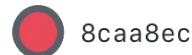
Tags and Releases

- Branching strategy (Git flow)

# Git Bisect



git bisect





Demo

# Activity: Git Bisect- the long way

1. Initiate the binary search: `git bisect start`.
2. Specify the commit where you noticed the code was broken: `git bisect bad <SHA>`.
3. Specify the commit where you knew things were working: `git bisect good <SHA>`.
4. Bisect will check you out to the midpoint between good and bad.
5. Run a test to see if the game would work at this point. Our test is to use `ls` to see if an `index.html` file exists.
6. If the game is still broken (there *is no* `index.html` file), type: `git bisect bad`.
7. If the game works (and there *is* an `index.html` file), type: `git bisect good`.
8. Git will bisect again and wait for you to test. This will happen until Git has enough information to pinpoint the first bad commit.
9. When Git has detected the error, it will provide a message that `SHA is the first bad commit`.
10. Exit the bisect process: `git bisect reset`.

# Activity: Git Bisect- the short way

1. `git bisect start <bad-SHA>`  
`<good-SHA>`
2. `git bisect run ls index.html`
3. `git bisect reset`

# Project 3: GitHub Games

Workflow review

- Protected Branches and CODEOWNERS
- Git Bisect

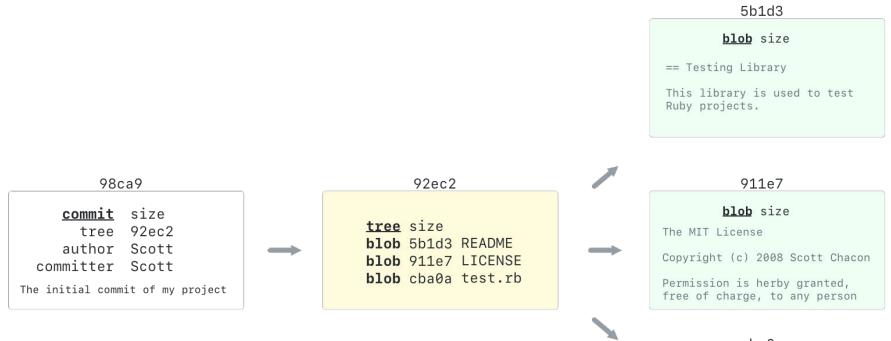
Reverting Commits

- Helpful Git commands
- Viewing local changes

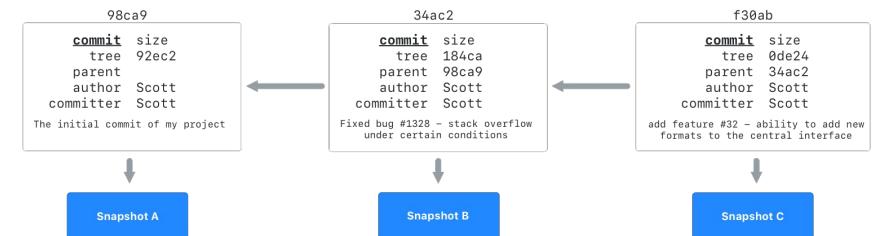
Tags and Releases

- Branching strategy (Git flow)

# Reverting commits



Commit object structure



Each commit pointing to its parent



Demo

## Activity: Revert a commit

1. Initialize the revert: `git revert <SHA>`
2. Type a commit message.
3. Push your changes to GitHub.

# Project 3: GitHub Games

Workflow review

- Protected Branches and CODEOWNERS
- Git Bisect

Reverting Commits

- Helpful Git commands
- Viewing local changes

Tags and Releases

- Branching strategy (Git flow)

# Helpful Git commands

```
diff --git a/_slides/02-gusshawstewart.md b/_slides/02-gusshawstewart.md
index 0d0dbbf..105c3d9 100644
--- a/_slides/02-gusshawstewart.md
+++ b/_slides/02-gusshawstewart.md
@@ -3,7 +3,7 @@ layout: slide
 title: "Welcome to our slide deck!"
 ---
-My enhancement
+My enhancement and change

![skitchtocat](https://octodex.github.com/images/skitchtocat.png)
{: .center}
(1/1) Stage this hunk [y,n,q,a,d,e,?]?
```



Demo

# Activity: moving and renaming files with Git

1. Create a new branch named `slow-down`.
2. On *line 9* of the `index.html` file, change the background url to `images/texture.jpg`.
3. On *line 78*, change the timing for the game to speed it up or slow it down.
4. Save your changes.
5. See what git is tracking:
6. `git status`
7. Create a new, empty directory:
8. `mkdir images`
9. Move the texture file into the directory with git:
10. `git mv texture.jpg images/texture.jpg`

# Activity: staging chunks of changes

1. See what git is tracking:
2. `git status`
3. Move some parts of some files to the staging area with the `--patch` flag:
4. `git add -p`
5. Stage the hunk related to the image move: y
6. Leave the hunk related to the speed change in the working area: 'n'

# Project 3: GitHub Games

Workflow review

- Protected Branches and CODEOWNERS
- Git Bisect

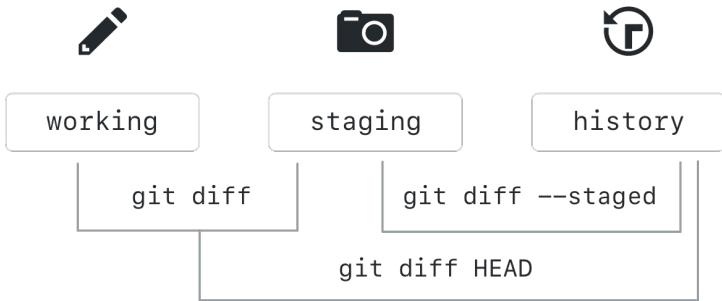
Reverting Commits

- Helpful Git commands
- Viewing local changes

Tags and Releases

- Branching strategy (Git flow)

# Viewing local changes





Demo

# Activity: comparing changes within the repository

1. `git diff`
2. `git diff --staged`
3. `git diff HEAD`
4. `git diff --color-words`
5. `git diff <REF-1> <REF-2>`
6. `git diff main slow-down`
7. `git diff origin/main main`
8. `git diff 2710 b745`

# Project 3: GitHub Games

Workflow review

- Protected Branches and CODEOWNERS
- Git Bisect

Reverting Commits

- Helpful Git commands
- Viewing local changes

Tags and Releases

- Branching strategy (Git flow)

# Tags

[Releases](#) [Tags](#)

🏷 Tags

v1.0

🕒 on 26 May · c0e3215 · [zip](#) · [tar.gz](#)

# Releases

Atom GitHub Releases Page

The screenshot shows the GitHub releases page for the Atom repository. It displays two releases: '1.59.0-beta0' (Pre-release) and '1.58.0' (Latest release). Each release includes a summary of notable changes, a 'Notable Changes' section with links to pull requests, and a 'Reacted' section showing user interactions.

**1.59.0-beta0** (Pre-release)

- v1.59.0-beta0 · atom-build · 27 Jul
- Compare

**Notable Changes**

- #16206 - Enable default editor keymaps for editors located outside atom-workspace
- #18321 - Implementing tree-sitter based indentation logic

All Changes

Reacted: 9, 2, 11 people reacted

**1.58.0** (Latest release)

- v1.58.0 · atom-build · 27 Jul
- Compare

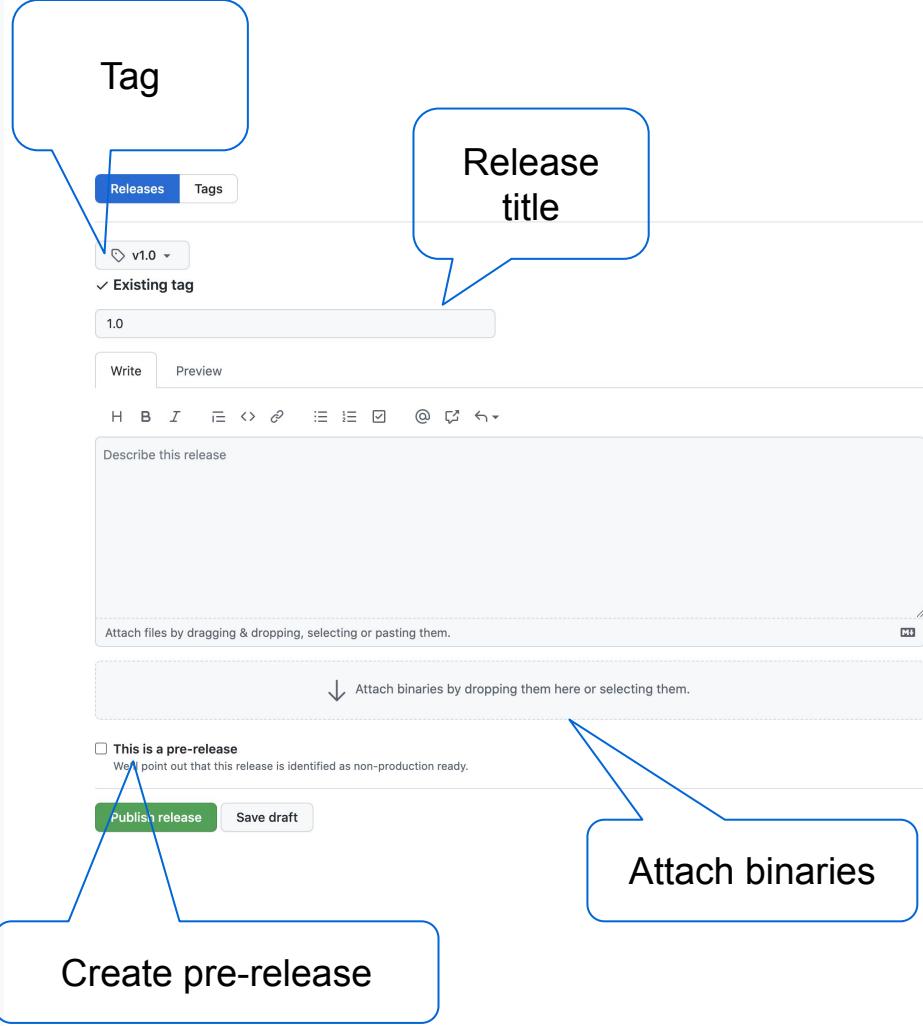
**Notable Changes**

- #22315 - Update to macOS Big Sur icon.
- #22424 - Fix reopening a project in safeMode and devMode.
- #22123 - Improve contrast on Windows app icons
- atom/archive-view#73 - Add ability to collapse archived directories (zip, tar, e.t.c)
- atom/bracket-matcher#405 - Handle multicursor selection inside brackets
- atom/find-and-replace#932 - Add "Open in New Tab" and "Open in New Window" right-click context menu options

All Changes

Reacted: 26, 5, 13, 8, 3, 5, 39 people reacted

# Creating a release





Demo

## Activity: Tags

1. git log --oneline
2. git tag v1.0 <SHA>
3. git show v1.0
4. git tag -a v2.0 <SHA>
5. git push --tags
6. git config --global  
push.followTags true

# Activity: Releases

1. On GitHub, navigate to the Code tab of the repository.
2. Under your repository name, click Releases.
3. Click Draft a new release.
4. Type a name for the tag. We recommend you use semantic versioning.
5. Select a branch that contains the project you want to release. Usually, you'll want to release against your main branch, unless you're releasing beta software. You can also select a recent commit by choosing the recent commits tab.
6. Type a title and description that describes your release.
7. If you're ready to publicize your release, click Publish release. Otherwise, click Save draft to work on it later.

# Project 3: GitHub Games

Workflow review

- Protected Branches and CODEOWNERS
- Git Bisect

Reverting Commits

- Helpful Git commands
- Viewing local changes

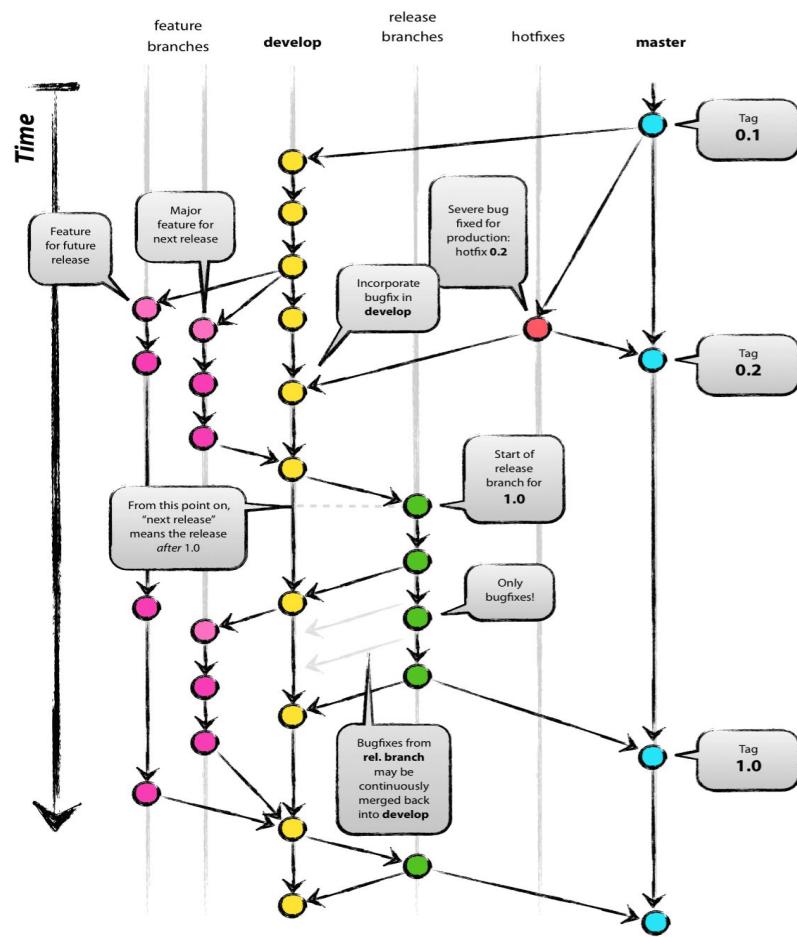
Tags and Releases

- Branching strategy (Git flow)

# Branching Strategy

Git Flow

# Git Flow



# Activity: workflow discussion

1. Which branching strategy will we use?
2. Which branch will serve as our "main" or deployed code?
3. How will you protect your code?
4. Will we use naming conventions for our branches?
5. How will we use labels and assignees?
6. Will we use milestones?
7. Will we have required elements of Issues or Pull Requests (e.g. shipping checklists)?
8. Who is expected to review your work? Do you plan to involve other teams?
9. How will we indicate sign-off on Pull Requests?
10. Who will merge pull requests?
11. How will you teach your workflow to your team? If it already exists, how is it taught to new hires?
12. What integrations will be used in different stages of development? Will all teams be using the same tools?
13. If users have questions about Git, GitHub, or their workflows, who do they ask? How do they know who to ask?

# Project 4: Local repository

- Create a local repo
- Fixing commit mistakes
- Rewriting history with Git Reset
- Cherry picking
- Merge strategies

# Create a local repo

```
[→ scratch git init practice-repo  
Initialized empty Git repository in  
po/.git/  
[→ scratch cd practice-repo  
→ practice-repo git:(main) █
```



Demo

# Activity: initialize a new local repository

1. Navigate to the directory where you will place your practice repo (cd .. to get back to the parent folder).
2. Create a new directory and initialize it as a git repository: `git init practice-repo`
3. CD into your new repository: `cd practice-repo`
4. Create an empty new file named `README.md`:
  - o Bash: `touch README.md`
  - o PowerShell: `Out-File README.md`
5. Add and commit the `README.md` file.

Bash:

```
for d in {1..6}; do touch "file${d}.md";
git add "file${d}.md"; git commit -m
"adding file ${d}"; done
```

PowerShell:

```
for ($d=1; $d -le 6; $d++) { Out-File
file$d.md; git add file$d.md; git commit
-m "adding file$d.md"; }
```

# Project 4: Local repository

- Create a local repo
- Fixing commit mistakes
- Rewriting history with Git Reset
- Cherry picking
- Merge strategies

# Fixing commit mistakes

```
→ practice-repo git:(main) ✘ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   file7.md

→ practice-repo git:(main) ✘ git commit --amend
```



Demo

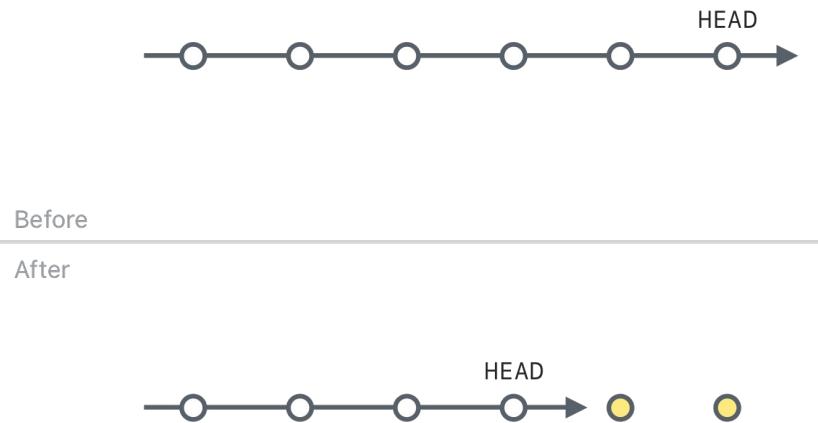
# Activity: revising your last commit

1. Create a new file:
  - Bash: `touch file7.md`
  - PowerShell: `Out-File file7.md`
2. When you are adding files to the previous commit, they should be in the staging area. Move your file to the staging area: `git add file7.md`
3. `git commit --amend`
4. The text editor will open, allowing you to edit your commit message.

# Project 4: Local repository

- Create a local repo
- Fixing commit mistakes
- Rewriting history with Git Reset
- Cherry picking
- Merge strategies

# Rewriting history with Git Reset



# Reset Soft





Demo

# Activity: Reset Soft

1. View the history of our project: `git log --oneline --decorate`
2. Identify the current location of `HEAD`.
3. Go back two commits in history: `git reset --soft HEAD~2`
4. See the tip of our branch (and `HEAD`) is now sitting two commits earlier than it was before: `git log --oneline --decorate`
5. The changes we made in the last two commits should be in the staging area: `git status`
6. All the files still exist locally: `ls`
7. Let's remove the extra file we created earlier: `git rm --cached file7.md`
8. Now, we'll re-commit these changes without the extra file: `git commit -m "re-add file 5 and 6"`

# Reset Mixed





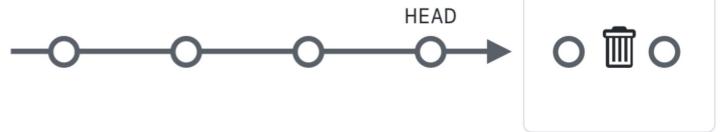
Demo

## Activity: Reset Mixed

1. Once again, we will start by viewing the history of our project: `git log --oneline`
2. Go back one commit in history: `git reset HEAD~`
3. See where the tip of the branch is pointing: `git log --oneline --decorate`
4. The changes we made in the last commit have been moved back to the working directory: `git status`
5. All the files still exist locally: `ls`
6. Move the files to the staging area before we can commit them: `git add file5.md file6.md`
7. Re-commit the files: `git commit -m "re-add file 5 and 6"`

# Reset Hard

--hard





Demo

# Activity: Reset Hard

1. Start by viewing the history of our project with: `git log --oneline`
2. Reset to the point in time where the only file that existed was the README.md: `git reset --hard <SHA>`
3. See that all of the commits are gone: `git log --oneline`
4. Notice your working directory is clean: `git status`
5. See that the only files in your repository are the README.md and file7.md: `ls`

# Project 4: Local repository

- Create a local repo
- Cherry picking
- Fixing commit mistakes
- Merge strategies
- Rewriting history with Git Reset

# Cherry picking and reflog

```
→ practice-repo git:(main) git reflog
e7f4fba (HEAD -> main) HEAD@{0}: reset: moving to e7f4fba
98e59b4 HEAD@{1}: cherry-pick: adding file 2
e7f4fba (HEAD -> main) HEAD@{2}: reset: moving to e7f4fba
2e5d407 HEAD@{3}: commit: adding file 6
0b369d5 HEAD@{4}: commit: adding file 5
ba09430 HEAD@{5}: commit: adding file 4
42e5989 HEAD@{6}: commit: adding file 3
305dfdc HEAD@{7}: commit: adding file 2
e7f4fba (HEAD -> main) HEAD@{8}: commit (initial): adding
```

```
→ practice-repo git:(main) git cherry-pick 42e5989
[main 37e4397] adding file 3
Date: Mon Sep 6 13:34:17 2021 +0100
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file3.md
```

```
→ practice-repo git:(main) git log --oneline
37e4397 (HEAD -> main) adding file 3
e7f4fba adding file 1
```



Demo

## Activity: cherry-pick a commit

1. Find the commit ID where you added file4.md: `git reflog`
2. Cherry-pick that commit: `git cherry-pick <SHA>`

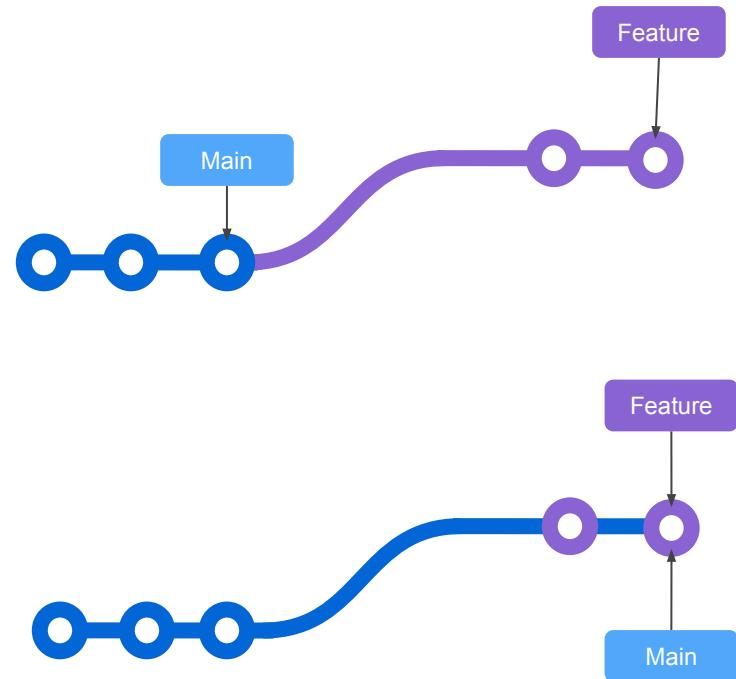
## Activity: directional reset

1. View the history of everywhere HEAD has pointed: `git reflog`
2. Reset to the point in time where the original `file6.md` was created: `git reset --hard <SHA>`
3. See your restored history: `git log --oneline`

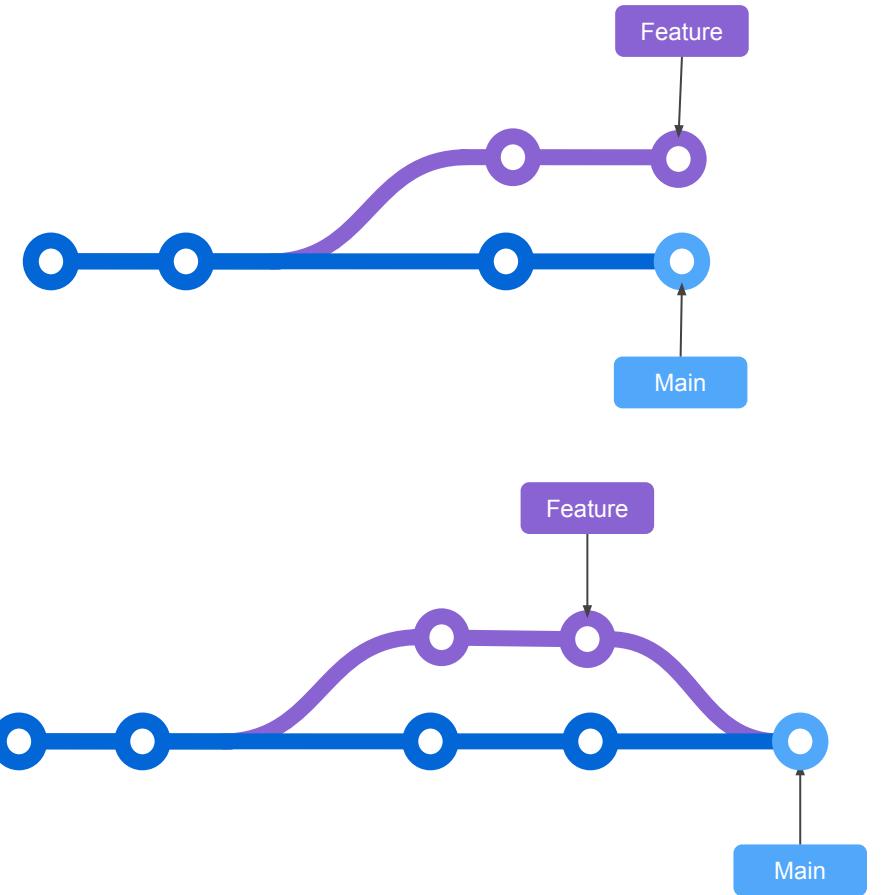
# Project 4: Local repository

- Create a local repo
  - Fixing commit mistakes
  - Rewriting history with Git Reset
- Cherry picking
  - Merge strategies

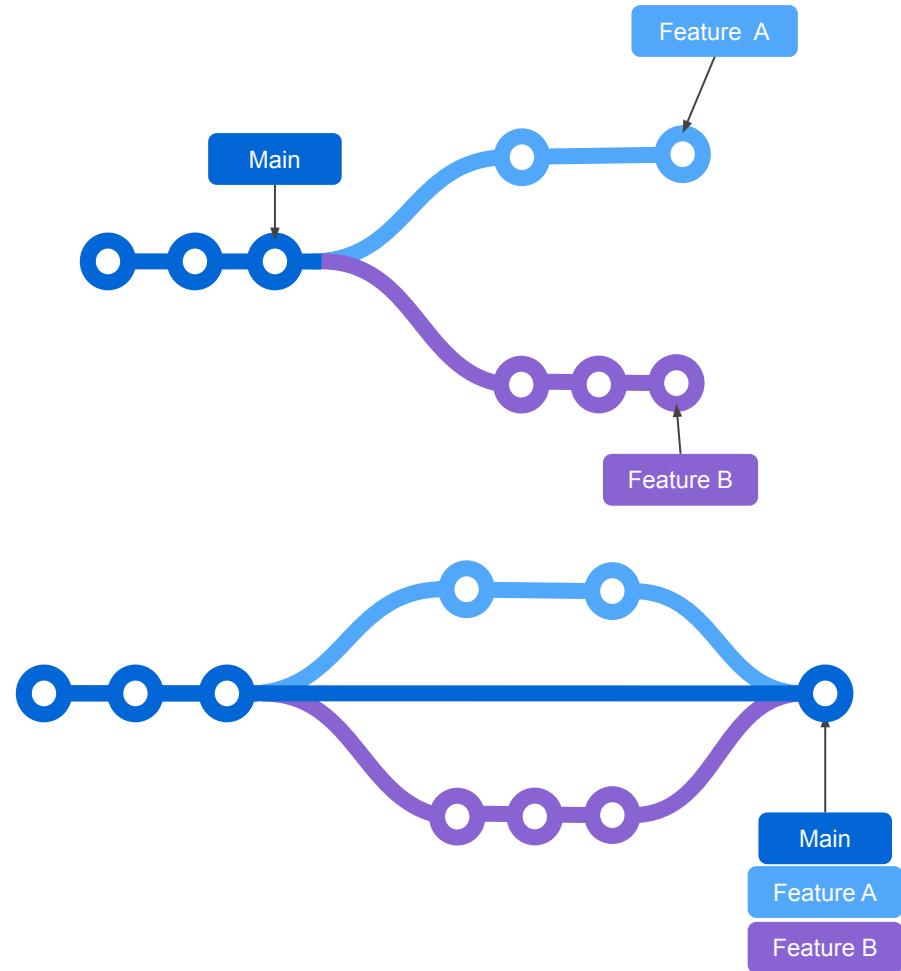
# Merge strategies: fast-forward merge



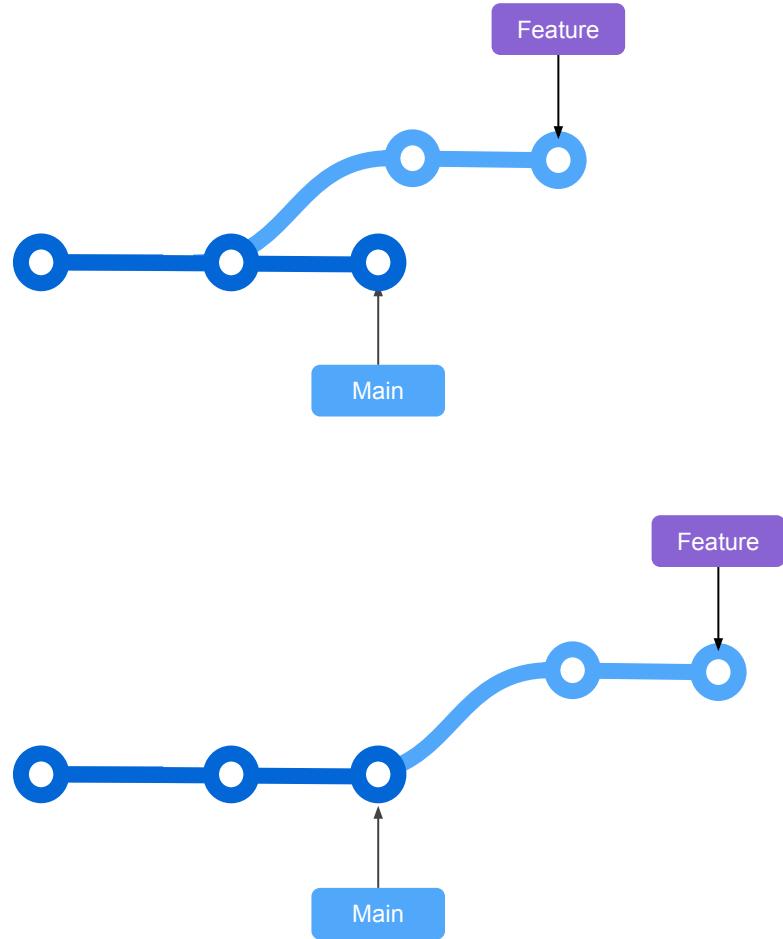
## Merge strategies: Recursive merge



# Merge strategies: Octopus merge



# Git Rebase





Demo

# Activity: setup to rebase

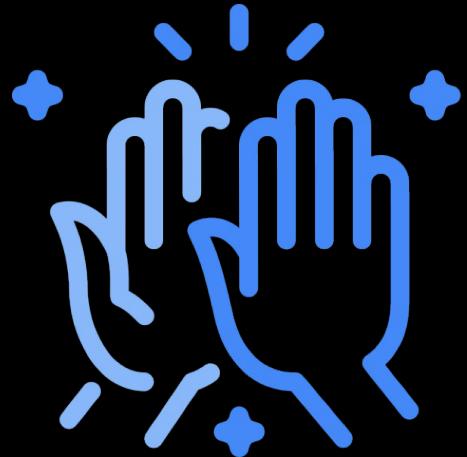
1. Find the SHA of the initial commit: `git log --oneline`
2. Reset to the SHA of the initial commit: `git reset --hard SHA`
3. Create a new branch and check out to it: `git switch -c rebase-me`
4. Cherry-pick files 4-6 onto the `rebase-me` branch using the reflog.
5. Switch to main: `git switch main`
6. Cherry-pick files 1-3 onto the `main` branch using the reflog.
7. Look at your history: `git log --oneline --graph --decorate --all`
8. If you merged now, it would be a recursive merge.

# Activity: Begin the rebase

1. Switch to the `rebase-me` branch: `git switch rebase-me`
2. Start the merge: `git rebase -i main`
3. Your text editor will open, allowing you to see the commits to be rebased.
4. Save and close the `rebase-todo`.
5. Watch your rebase happen on the command line.
6. Take another look at your history: `git log --oneline --graph --decorate --all`
7. If you merged now, it would be a fast-forward merge.

## Finish the Merge

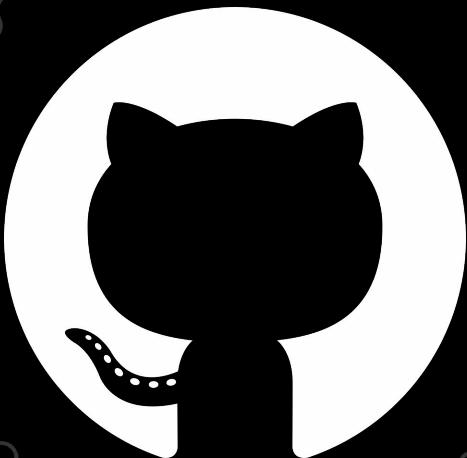
1. Switch to `main`, the branch you will merge into: `git switch main`
2. Merge your changes in to `main`: `git merge rebase-me`



# Recap Q&A

# Appendix

1. [https://githubtraining.github.io/training-manual/#/01\\_getting\\_ready\\_for\\_class](https://githubtraining.github.io/training-manual/#/01_getting_ready_for_class)
2. <https://guides.github.com/introduction/flow/>
3. <https://training.github.com/downloads/github-git-cheat-sheet.pdf>
4. <http://git-school.github.io/visualizing-git/>



**Thank you**