

✨ Git 101 ✨

Kristen Kwong



✨ Git 101 ✨

Kristen Kwong

Slides: <https://kristen.dev/blog/git>

Install Git: git-scm.com/book/en/v2/Getting-Started-Installing-Git

Make a GitHub account: github.com/join



Hey I'm Kristen! 🙋

- ✨ final year of Computer Science
- ✨ Organized Local Hack Day 2016, 2017, & 2018
- ✨ Interned at Netgear and Apple
- ✨ Soon: full-time software engineer @ Apple
- ✨ Operating systems, networking, compilers
- ✨ Also have experience with Python scripting, web development, CSS for fun



Agenda

✨ What is Git? Why use Git?

✨ How does Git work?

✨ Installing & setting up Git and GitHub

✨ Your Project + Git = 💖

Let's talk about collaboration.

If we were building a project as a team, how would you get your code to each other?

Collaborating on projects☆

☆without git :(

What about sending snippets on FB Messenger?

```
with open("flag.txt", "r") as f:
    data = f.readlines()

for line in data:
    words = line.split(" ")

    output = ""
    for word in words:
        dihdah = word.split("-")
        for i in range(0, len(dihdah)):
            if dihdah[i] == "di" or dihdah[i] == "dit":
                dihdah[i] = "."
            else:
                dihdah[i] = "-"
        output += "".join(dihdah)
    output += " "

print output
```

```
with open("flag.txt", "r") as f:
    data = f.readlines()

for line in data:
    words = line.split(" ")

    output = ""
    for word in words:
        dihdah = word.split("-")
        for i in range(0, len(dihdah)):
            if dihdah[i] == "di" or dihdah[i] == "dit":
                dihdah[i] = "."
            else:
                dihdah[i] = "-"
        output += "".join(dihdah)
    output += " "
```

✨ inconvenient to describe where & how to insert code

✨ what if we have small changes?

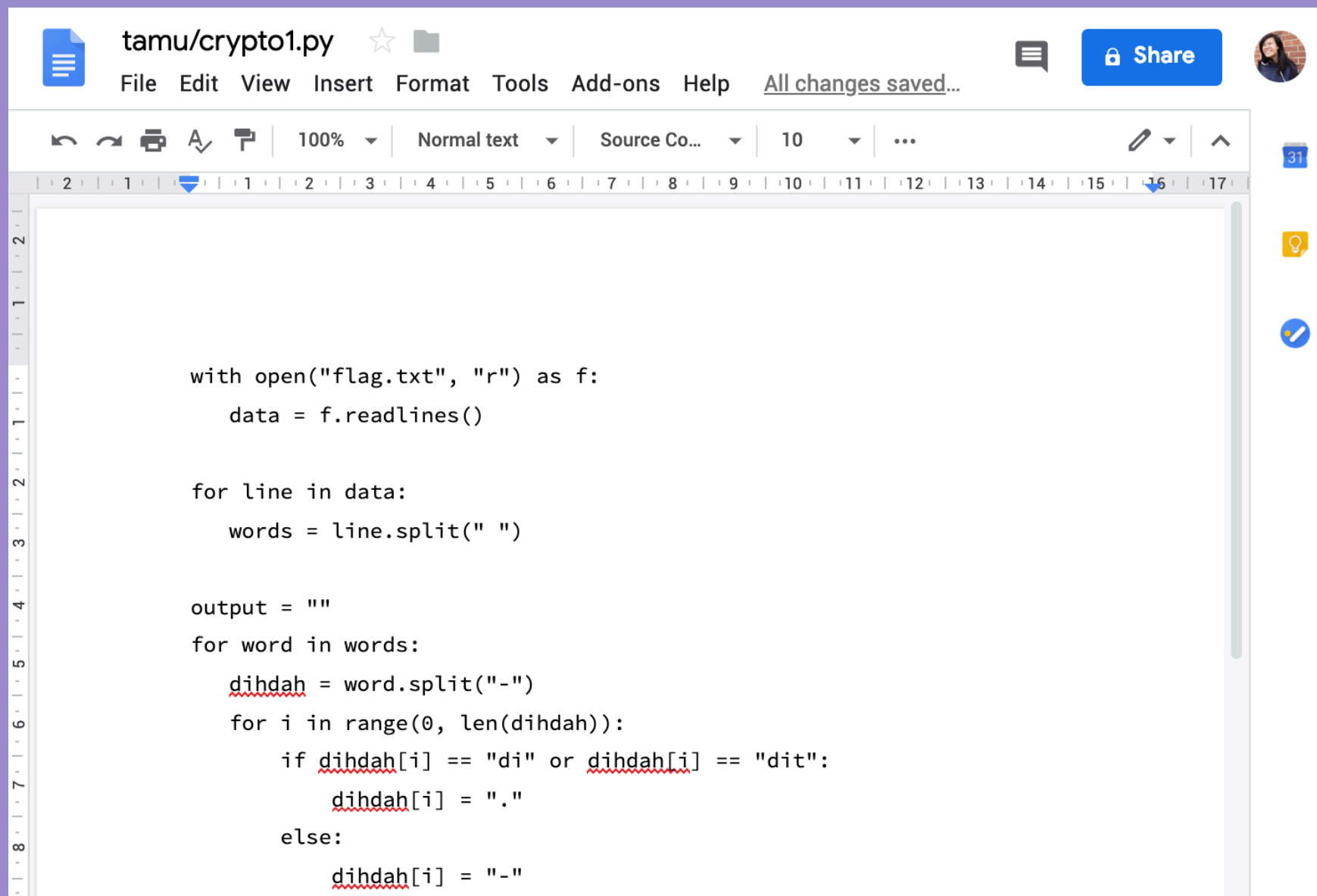
ie. we want to change the strings on line 15 & 17

✨ how do we know what we changed in our code previously?

Collaborating on projects☆

☆without git :(

Maybe Google Docs?

A screenshot of a Google Docs document titled 'tamucrypto1.py'. The interface includes a top menu bar with 'File', 'Edit', 'View', 'Insert', 'Format', 'Tools', 'Add-ons', and 'Help'. A status bar below the menu indicates 'All changes saved...'. The document content is a Python script for processing a file named 'flag.txt'. The script uses 'f.readlines()' to read the file, splits each line into words, and then processes each word by splitting it on hyphens. It checks if any segment of the split word matches 'di' or 'dit' and replaces them with a period. The script is displayed in a monospaced font with line numbers on the left margin. The right sidebar shows a 'Share' button and a user profile picture.

```
with open("flag.txt", "r") as f:
    data = f.readlines()

for line in data:
    words = line.split(" ")

output = ""
for word in words:
    dihdah = word.split("-")
    for i in range(0, len(dihdah)):
        if dihdah[i] == "di" or dihdah[i] == "dit":
            dihdah[i] = "."
        else:
            dihdah[i] = "-"
```



```
tamu/crypto1.py
File Edit View Insert Format Tools Add-ons Help All changes saved...

with open("flag.txt", "r") as f:
    data = f.readlines()

for line in data:
    words = line.split(" ")

output = ""
for word in words:
    dihdah = word.split("-")
    for i in range(0, len(dihdah)):
        if dihdah[i] == "di" or dihdah[i] == "dit":
```

✨ annoying to have to copy & paste code into file that can actually be run locally

✨ low-key kinda ugly to look at code this way - format not meant for coding projects

Collaborating on projects☆

☆without git :(

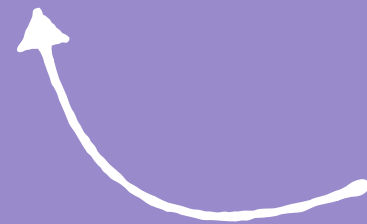
Some other ideas:

- ✧✧ Working all on the same computer - inefficient
- ✧✧ Emailing files - what if files are large? or there are many?
- ✧✧ Just don't have teammates??

Version control to the rescue!

What is Version Control?

- ✨ Keeps track of a history of changes
- ✨ Allows for collaborative development
- ✨ Go back and revert to an older version



for when we mess up so badly we just want to give up 🙄 (but also if requirements change)

What is Git?

✨ type of **version control** - not the only one!

✨ Subversion, Perforce, Bazaar

✨ integrates directly into your project workflow - most IDEs support version control

Repository

✨ ✨ “repo” for short

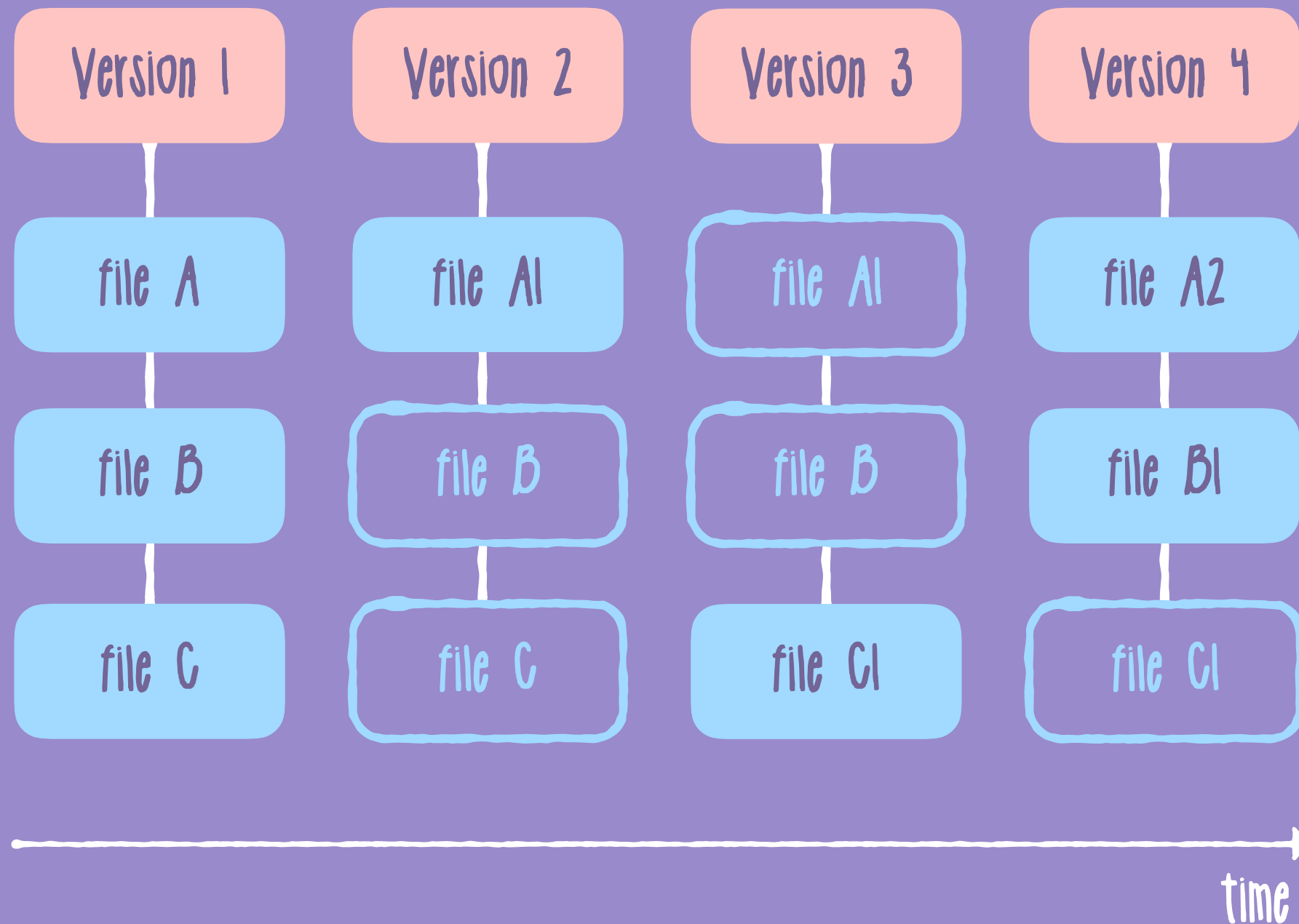
✨ ✨ each developer has a copy of the code in a **local** repository, and (most likely) a **remote** repository on a server

✨ ✨ where Git stores the metadata and object database for the project

Snapshots

- ✨ Other VCs tend to have a single file and records changes to that base file - but not Git
- ✨ Git saves a “picture” of what your files look like
- ✨ If it hasn’t changed, Git doesn’t store the file again - just link to previous unchanged version

Snapshots



Committing

✨ refers to making a snapshot

✨ common phrases: “I just committed code”, “I made a commit”

✨ projects are a bunch of commits

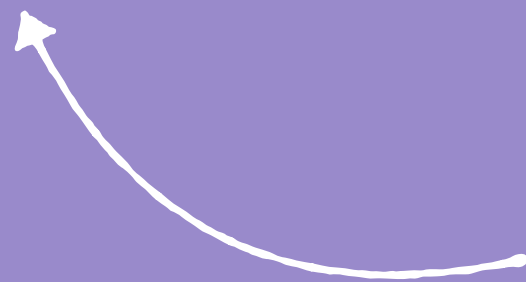
✨ like saving in a video game

A Git File's Three States

✨ ✨ **Modified** - changed but not stored in repository yet

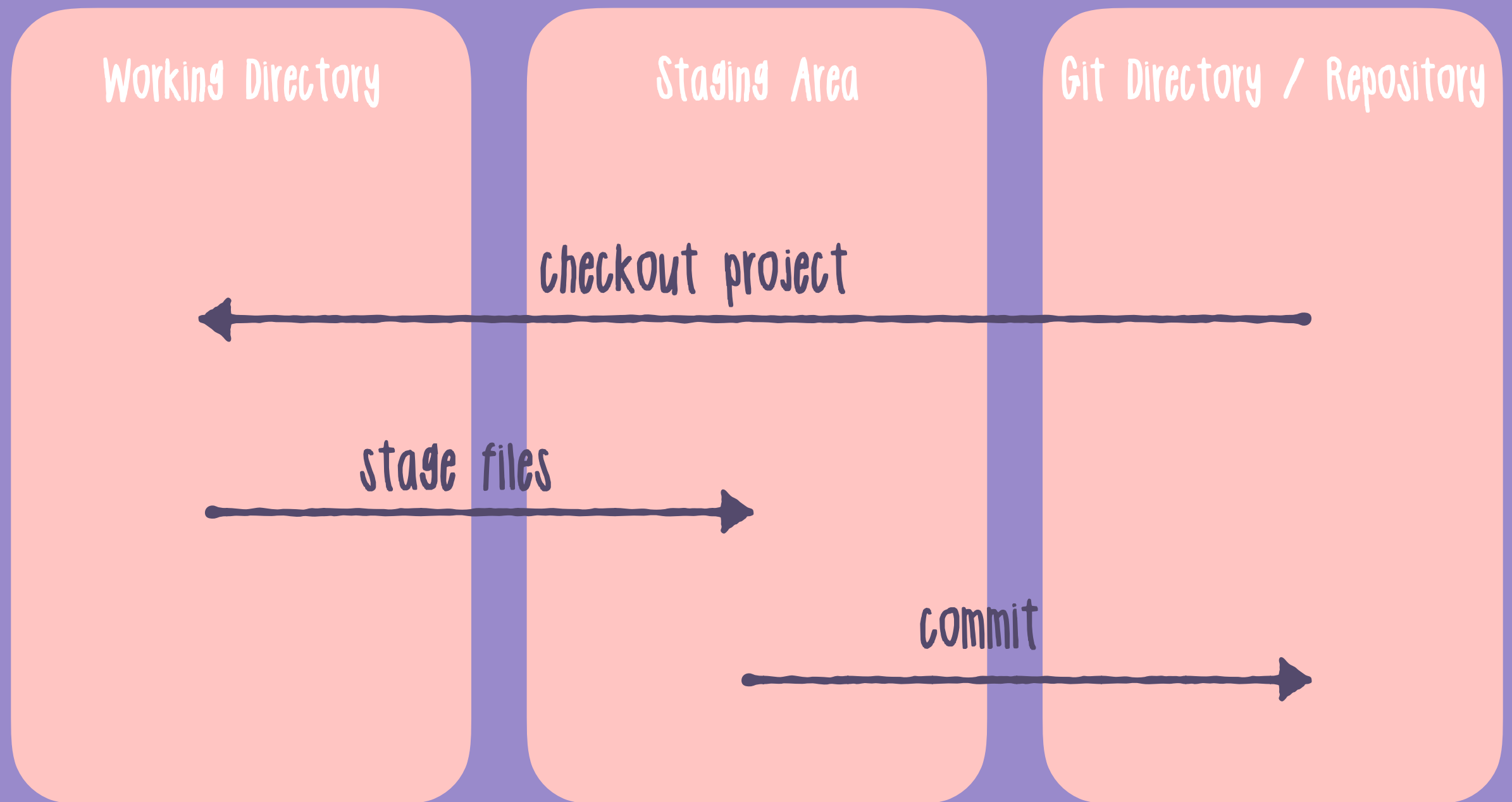
✨ ✨ **Staged** - marked that this goes into the next commit to
be stored into repository

✨ ✨ **Committed** - stored into the repository



all files in your Git project will
always be in one of these states

Three Sections of Git



Let's GIT going!

Installing Git


✨ Instructions: <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

✨ Check if installed correctly:

```
$ git --version  
git version 2.10.0
```

Setting up a Git repo

Use `git init` to make any folder a git repository.



```
$ mkdir project-folder  
$ cd project-folder  
$ git init  
Initialized empty Git repository in /Users/kristen/  
Documents/repos/temp/project/.git/
```

Adds a local git repo to the project



Configuring Git

Use `git config --global user.name "<your name>"`
and `git config --global user.email <email>` so
your contributions can be identified.

To show your configs, just use `git config --list`

Let's add some code

Let's make a file in our new Git directory:

A terminal window with a dark background and three light-colored window control buttons in the top-left corner. It contains two lines of text in a light blue monospace font. A white arrow points from the text below to the first line of code.

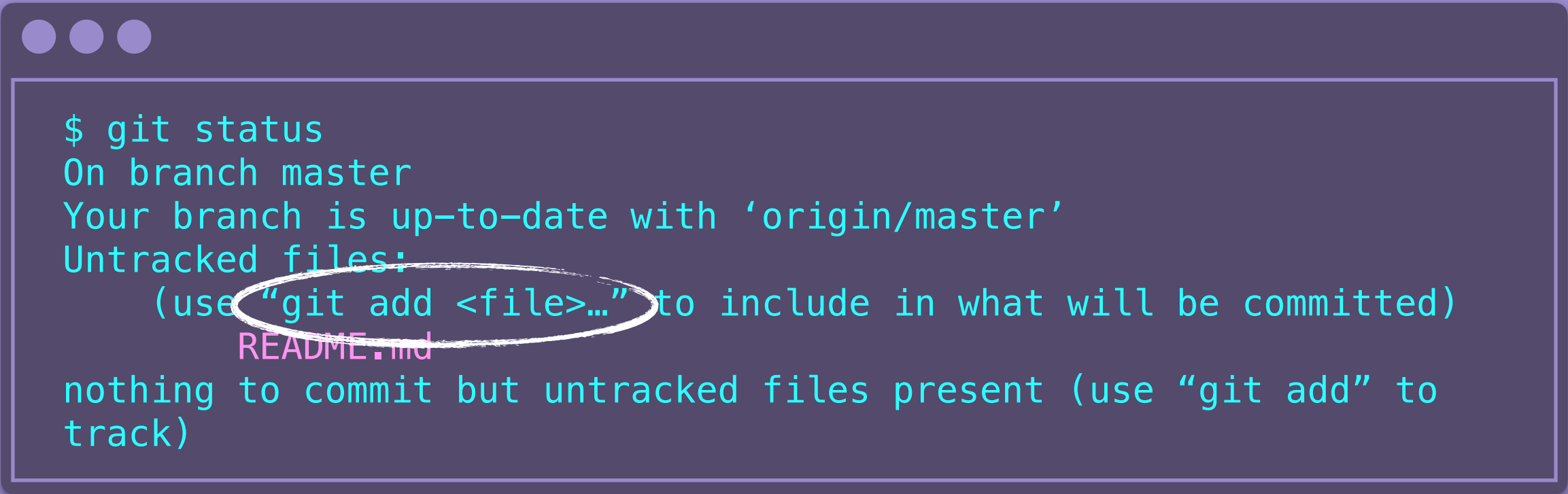
```
$ touch README.md  
$ echo "I'm learning Git" > README.md
```

This makes a file called README.md and writes "I'm learning Git" to it!

You could also just make a new file with Notepad or TextEdit and save it to the folder.

See changes

Use `git status` to view differences between your working directory and the git local repo.

A terminal window with a dark background and light blue text. The output of the 'git status' command is shown. The text 'git add <file>...' is circled in white. The file name 'README.md' is highlighted in pink. The window has three small circles in the top-left corner.

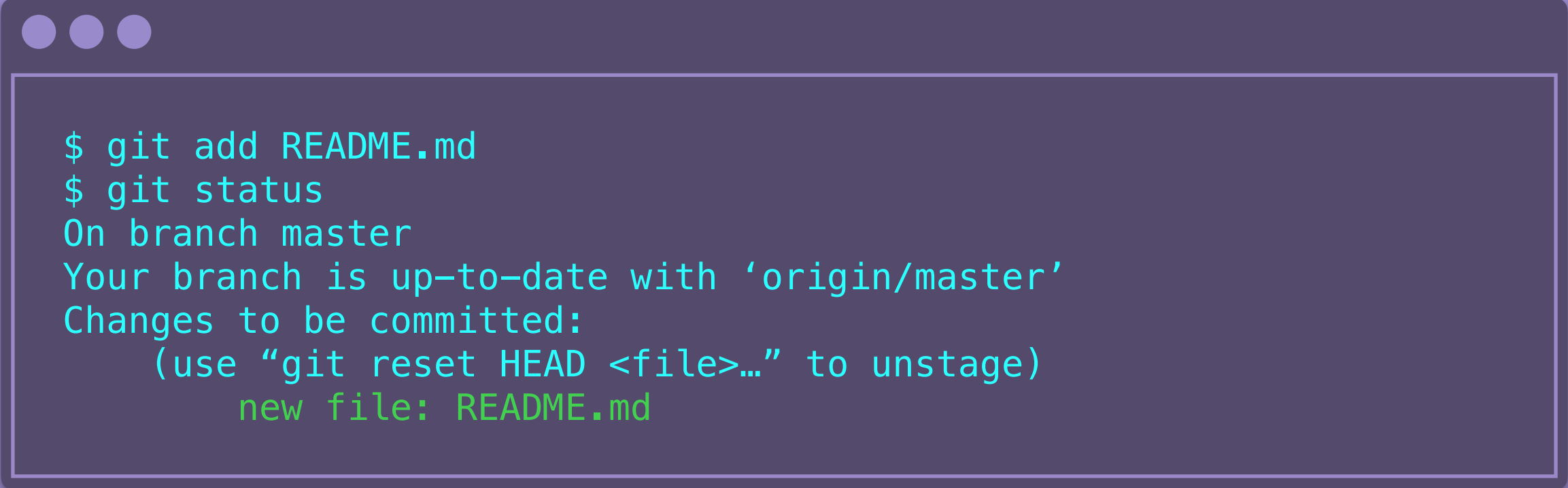
```
$ git status
On branch master
Your branch is up-to-date with 'origin/master'
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        README.md
nothing to commit but untracked files present (use "git add" to track)
```

Staging Changes

`git add .` to add all files in the folder to staging

`git add <file>` to add a single file

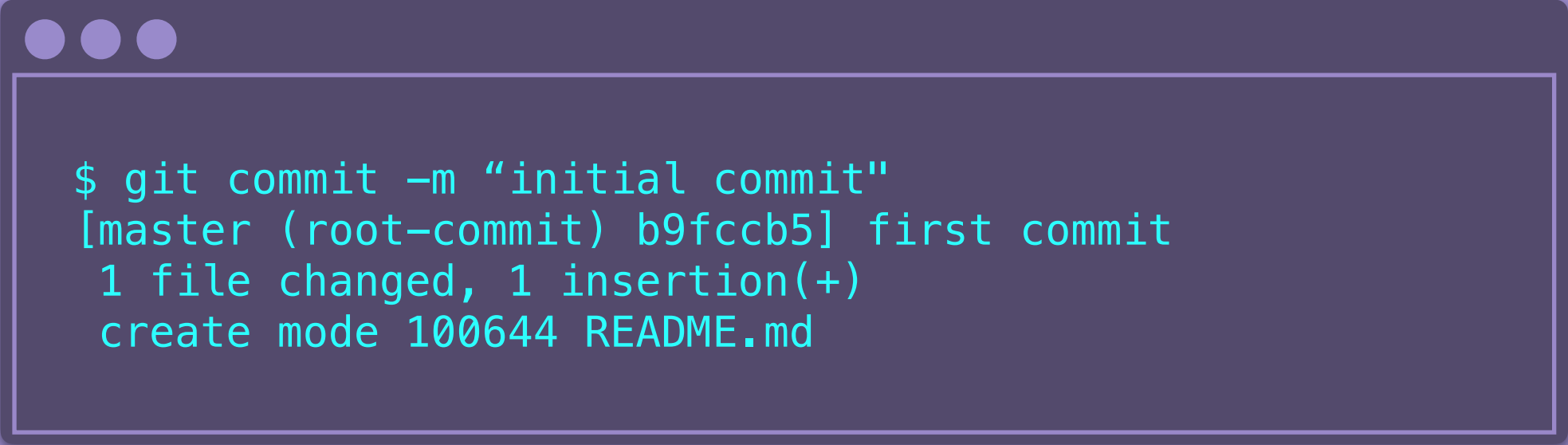
`git add <file1> <file2> <file3>` to add multiple



```
$ git add README.md
$ git status
On branch master
Your branch is up-to-date with 'origin/master'
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)
    new file:   README.md
```

Making a commit

`git commit -m "commit message"` makes a new commit to the local repository with the given commit message

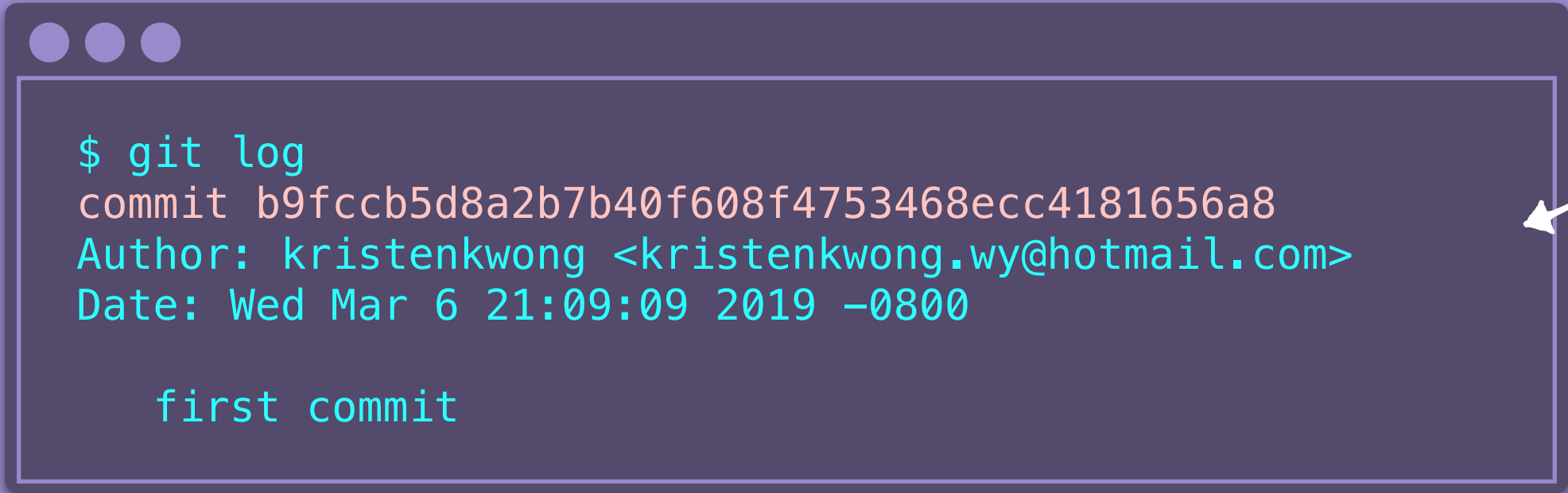
A terminal window with a dark background and light blue text. It shows the output of a git commit command. The window has three small circles in the top-left corner, representing window control buttons.

```
$ git commit -m "initial commit"
[master (root-commit) b9fccb5] first commit
1 file changed, 1 insertion(+)
create mode 100644 README.md
```

Showing all commits

`git log` shows all the commits you've made up to this point

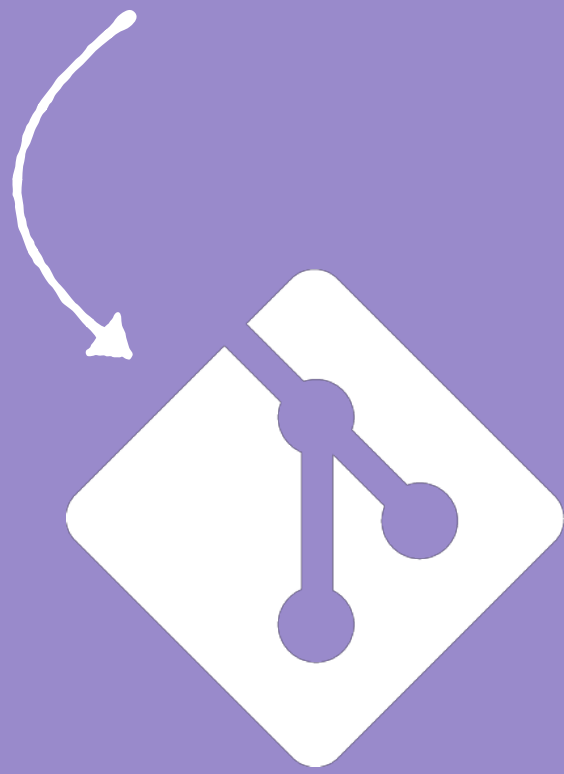
author, date, and commit message



```
$ git log
commit b9fccb5d8a2b7b40f608f4753468ecc4181656a8
Author: kristenkwong <kristenkwong.wy@hotmail.com>
Date: Wed Mar 6 21:09:09 2019 -0800

    first commit
```

Branches!

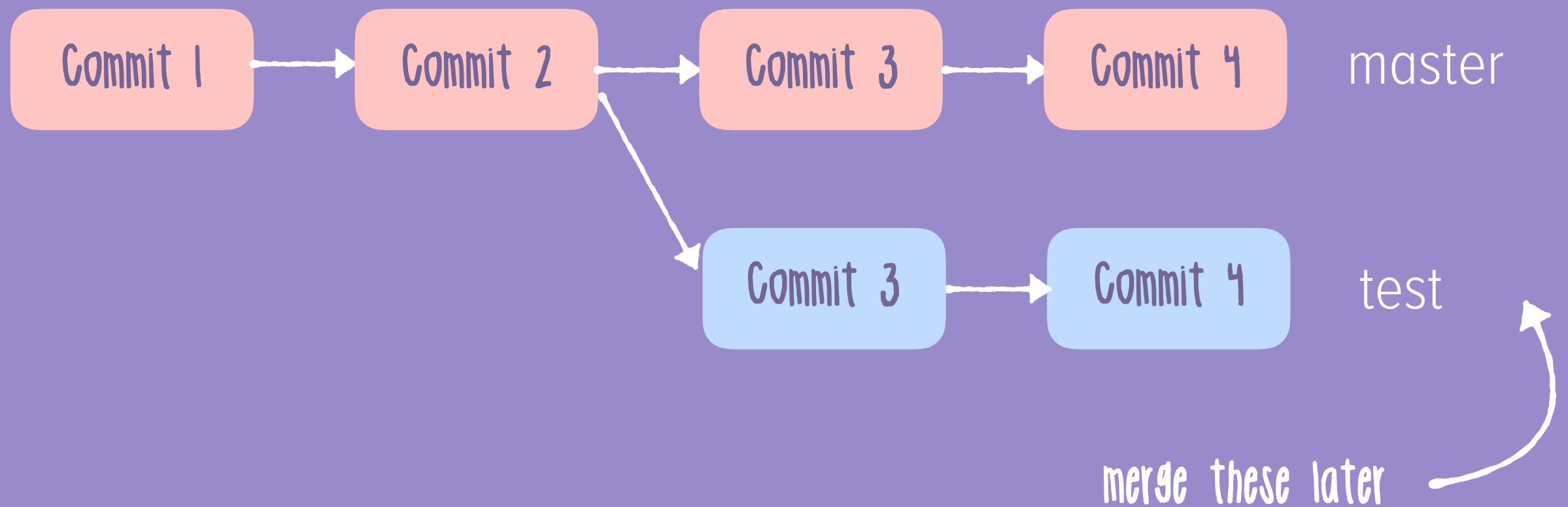


git

Branches in Git



Branches in Git



Make a new branch


Use `git branch <name>` to create a new branch.

A dark-themed terminal window with three light-colored window control buttons (minimize, maximize, close) in the top-left corner. The terminal contains the command `$ git branch test` in a light blue/cyan monospace font.

```
$ git branch test
```


List all branches


Use `git branch` to see all the branches in the repo.

A terminal window with a dark background and three light-colored window control buttons in the top-left corner. The terminal displays the output of the 'git branch' command.

```
$ git branch  
* master  
test
```

Switch branches


Use `git checkout <name>` to switch to that branch.

A dark-themed terminal window with three light-colored window control buttons (minimize, maximize, close) in the top-left corner. The terminal displays the command `$ git checkout test` and its output `Switched to branch 'test'` in a light blue/cyan monospace font.

```
$ git checkout test
Switched to branch 'test'
```


Let's add some stuff

Let's add some stuff to our new branch so we can merge:



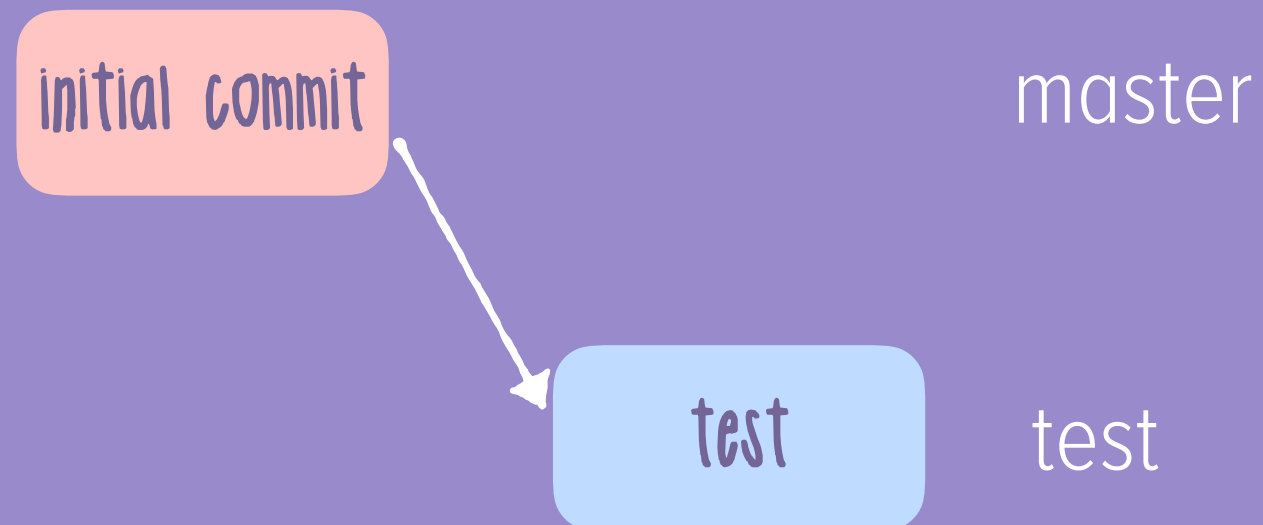
```
$ touch hello.txt  
$ echo "hello" > hello.txt
```

Then add and commit to the repo:



```
$ git add hello.txt  
$ git commit -m "test"
```

Example

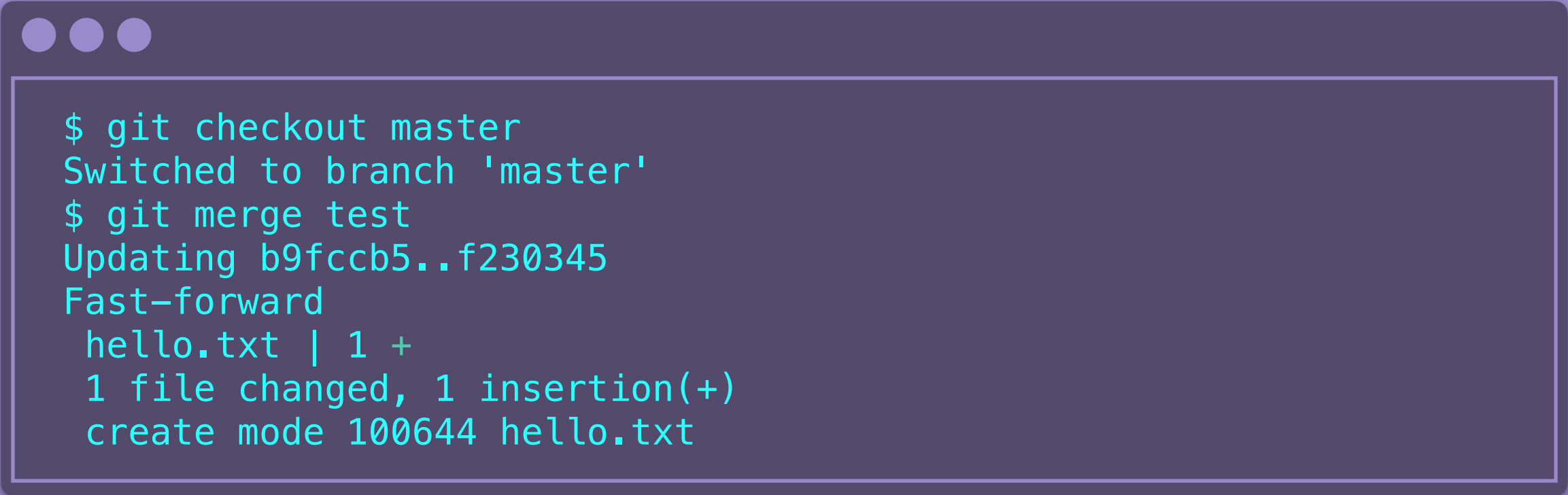


test is ahead of master by 1 commit

Merging branches

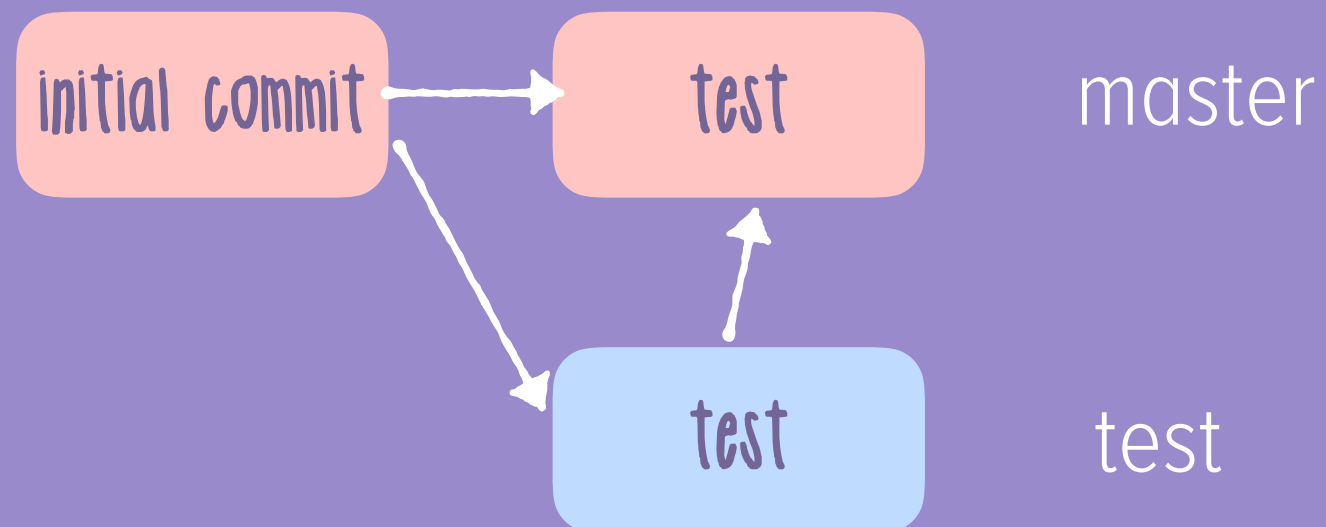
Use `git checkout master` to switch back to master.

Use `git merge <branch>` to copy code into checked out branch from the specified one.

A terminal window with a dark background and light blue text. It shows the execution of two git commands: 'git checkout master' and 'git merge test'. The output of the first command is 'Switched to branch 'master''. The output of the second command shows the merge process, including the commit hash 'b9fccb5..f230345', the word 'Fast-forward', and a summary of changes: '1 file changed, 1 insertion(+)' and 'create mode 100644 hello.txt'.

```
$ git checkout master
Switched to branch 'master'
$ git merge test
Updating b9fccb5..f230345
Fast-forward
 hello.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 hello.txt
```

Example



successfully merged code in test into master!

Resolving conflicts

You have to manually edit files to remove conflicts and then use `git add <filename>` to mark them as merged.

You can preview merges with:

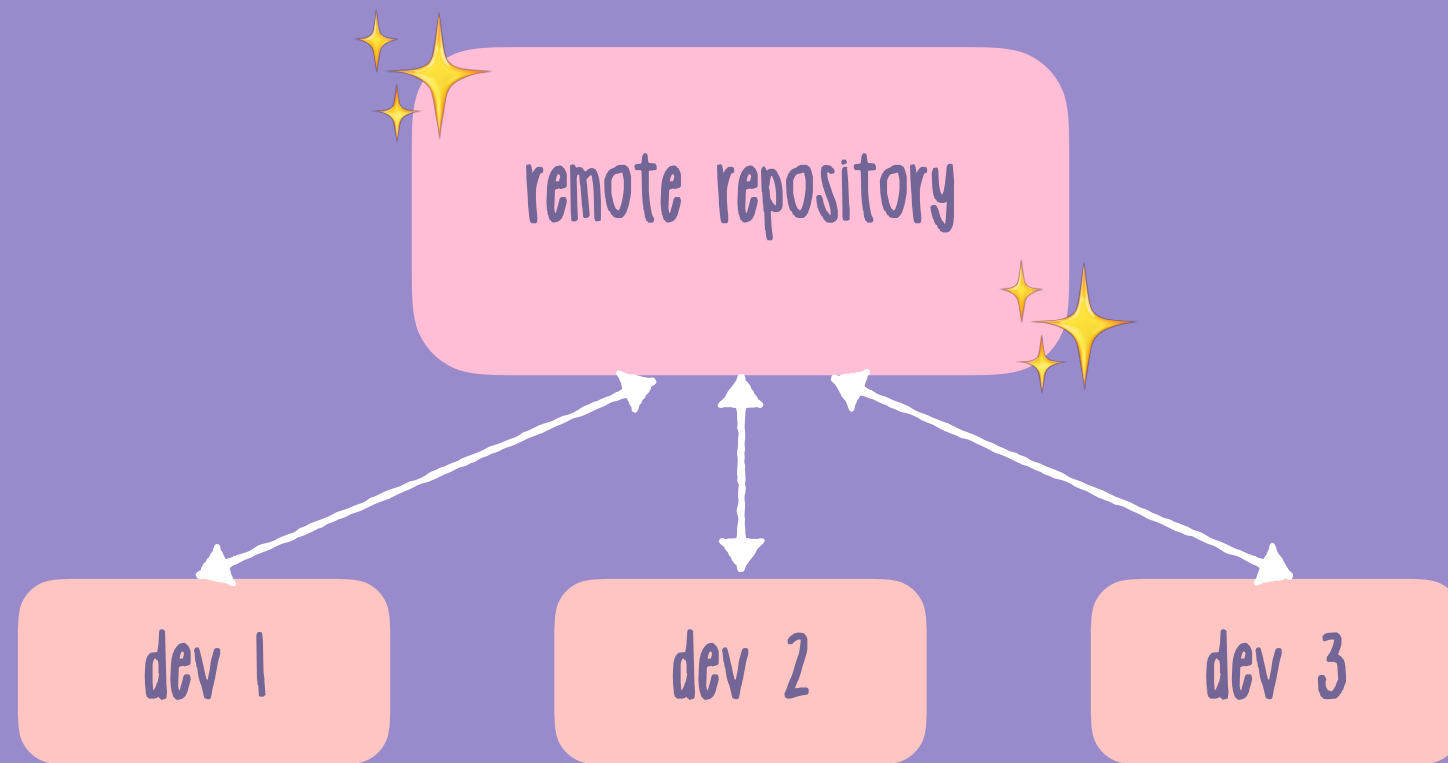
```
git diff <source-branch> <dest-branch>
```

But wait!

✨ So far... we've only put our changes in our LOCAL repo

✨ I said it was about collaboration! 😡

Remote Repository



so that everyone can see your code :)

GitHub

- ✨ Web-hosted version control
- ✨ Make an account at github.com/join
- ✨ Students: education.github.com/pack
- ✨ Free :)



Make a new GitHub project

The screenshot shows the GitHub profile of Kristen Kwong. The profile includes a bio, a profile picture, and a list of repositories. A green button labeled 'New' is circled in the top right of the repository list, with a handwritten arrow pointing to it and the text 'create new repo' written below it.

Profile Information:

- Name:** Kristen
- Username:** kristenkwong
- Developer Program Member:** Yes
- PRO:** Yes
- Bio:** Computer Science @ University of British Columbia. Co-founder @ubchacks; Local Hack Day director @nwplus. Previously interned @apple.

Repositories:

- dns-resolver-client** (Private)
 - Language: Java
 - Updated 2 days ago
- kristenkwong.github.io**
 - Language: HTML
 - Stars: 1
 - Updated 3 days ago
- firstname.dev**
 - Forked from CydeWeys/firstname.dev
 - Description: A list of {firstname}.dev domain names
 - Stars: 11
 - License: Apache License 2.0
 - Updated 3 days ago
- ftpcclient**

Create a new repository

A repository contains all project files, including the revision history.

Owner

 kristenkwong ▾


Repository name *

/ helloworld ✓

Great repository names are short and memorable. Need inspiration? How about **psychic-adventure**?

Description (optional)

☒  **Public**
Anyone can see this repository. You choose who can commit.

☐  **Private**
You choose who can see and commit to this repository.

☐ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾

Add a license: **None** ▾



Create repository

Quick setup — if you've done this kind of thing before

 Set up in Desktop

or

HTTPS

SSH

`https://github.com/kristenkwong/helloworld.git`



Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

repo link

...or create a new repository on the command line

```
echo "# helloworld" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/kristenkwong/helloworld.git
git push -u origin master
```



...or push an existing repository from the command line

```
git remote add origin https://github.com/kristenkwong/helloworld.git
git push -u origin master
```



...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code

Adding a remote repo

`git remote add origin <server>` connects your local repo with a remote server

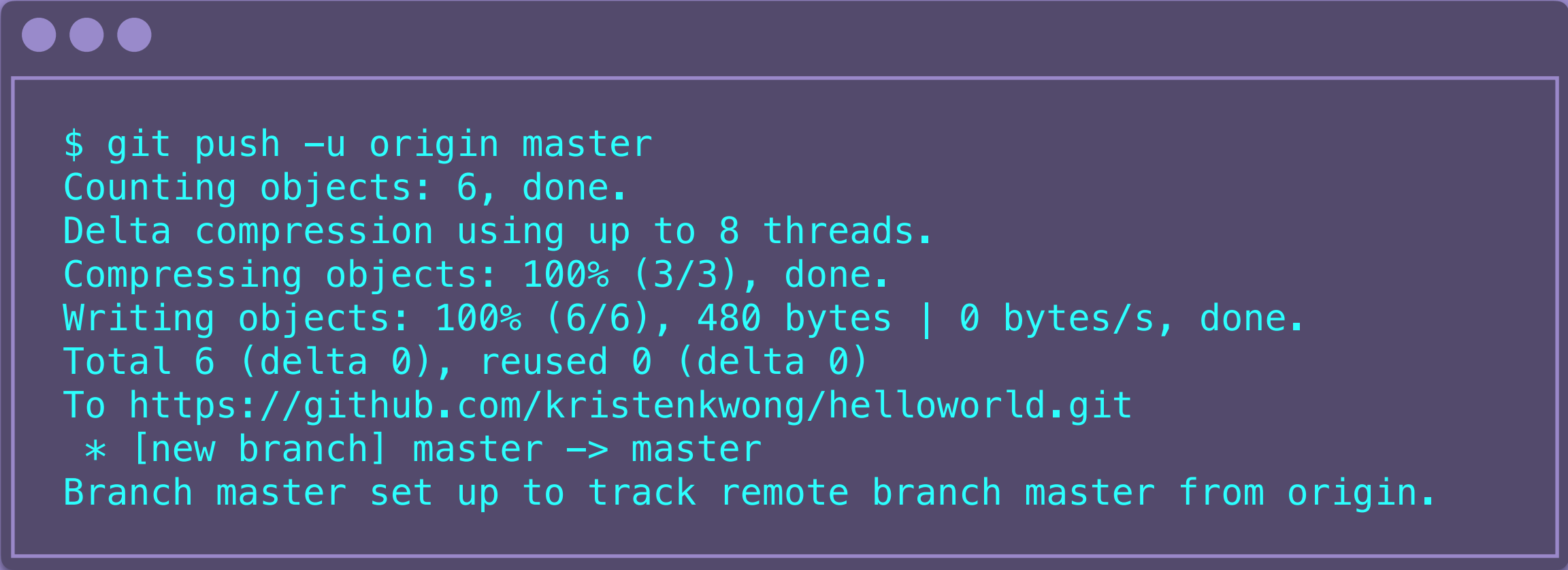
short name

repo link goes here

```
$ git remote add origin https://github.com/kristenkwong/helloworld.git
```

Pushing Code Changes

`git push -u origin master` copies code from the master branch in local to the remote.



```
$ git push -u origin master
Counting objects: 6, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (6/6), 480 bytes | 0 bytes/s, done.
Total 6 (delta 0), reused 0 (delta 0)
To https://github.com/kristenkwong/helloworld.git
 * [new branch] master -> master
Branch master set up to track remote branch master from origin.
```

Pushing Code Changes

set upstream tracking
`git push -u origin master`

`-u` makes sure that a **tracking connection** between the local and remote branch is established.

You can use `git push` to push changes on all upstream branches to the remote.

Pushing Branches

Local branches are also not visible on the remote repo unless you use `git push -u origin <branch>`

```
$ git checkout test ← switch back to the test branch
Switched to branch 'test'

$ git push -u origin test ← push the test branch
Counting objects: 3, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 285 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'test' on GitHub by visiting:
remote:   https://github.com/kristenkwong/helloworld/pull/new/test
remote:
To https://github.com/kristenkwong/helloworld.git
 * [new branch] test -> test
Branch test set up to track remote branch test from origin.
```

Pull Requests

- ✨ you might want others to review your code before you push to master
- ✨ you might want to share code that's incomplete
- ✨ it's also not that great to push to master - might break other people's code

Pushing Branches

After pushing branch, you can create a **pull request** to merge it with master.

Making a Pull Request

kristenkwong / helloworld Private


Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Your recently pushed branches:

test (14 minutes ago) [Compare & pull request](#)

Filters is:pr is:open Labels 8 Milestones 0 [New pull request](#)



Welcome to Pull Requests!

Pull requests help you collaborate on code with other people. As pull requests are created, they'll appear here in a searchable and filterable list. To get started, you should [create a pull request](#).

Making a Pull Request

The screenshot shows the GitHub interface for a repository named 'helloworld' by user 'kristenkwong'. The repository is marked as 'Private'. At the top, there are buttons for 'Unwatch' (1), 'Star' (0), and 'Fork' (0). Below this is a navigation bar with links to 'Code', 'Issues' (0), 'Pull requests' (0), 'Projects' (0), 'Wiki', 'Insights', and 'Settings'. The main heading is 'Open a pull request', followed by a subtext: 'Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).' Below this is a comparison bar showing 'base: master' and 'compare: test', with a green checkmark and the text 'Able to merge. These branches can be automatically merged.' The main content area is a text editor for the pull request comment, titled 'Merging the test branch!'. It has tabs for 'Write' and 'Preview'. The text 'this is a pull request comment' is entered. Below the text area is a note: 'Attach files by dragging & dropping, selecting them, or pasting from the clipboard.' and a footer note: 'Styling with Markdown is supported'. On the right side, there are sections for 'Reviewers' (No reviews), 'Assignees' (No one—assign yourself), 'Labels' (None yet), 'Projects' (None yet), and 'Milestone' (No milestone). At the bottom right, a green button labeled 'Create pull request' is circled in blue. The bottom of the page shows summary statistics: '1 commit', '1 file changed', '0 commit comments', and '1 contributor'.

kristenkwong / helloworld Private

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base: master ← compare: test ✓ Able to merge. These branches can be automatically merged.

Merging the test branch!

Write Preview

this is a pull request comment

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

Styling with Markdown is supported

Create pull request

Reviewers: No reviews

Assignees: No one—assign yourself

Labels: None yet

Projects: None yet

Milestone: No milestone

1 commit 1 file changed 0 commit comments 1 contributor

Merging the Pull Request

The screenshot shows a GitHub pull request interface for the repository 'kristenkwong / helloworld'. The pull request is titled 'Merging the test branch! #1' and is in the 'Open' state. It shows that 'kristenkwong' wants to merge 1 commit into the 'master' branch from the 'test' branch. The interface includes tabs for 'Code', 'Issues', 'Pull requests', 'Projects', 'Wiki', 'Insights', and 'Settings'. A comment from 'kristenkwong' is visible, stating 'this is a pull request comment'. The pull request details section shows that continuous integration has not been set up, but the branch has no conflicts with the base branch, allowing for automatic merging. A green 'Merge pull request' button is highlighted with a red circle. The right sidebar contains sections for 'Reviewers', 'Assignees', 'Labels', 'Projects', 'Milestone', 'Notifications', and '1 participant'.

kristenkwong / helloworld Private

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 1 Projects 0 Wiki Insights Settings

Merging the test branch! #1

Edit

Open kristenkwong wants to merge 1 commit into master from test

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

kristenkwong commented just now

this is a pull request comment

test 2629247

Add more commits by pushing to the test branch on kristenkwong/helloworld.

Continuous integration has not been set up
Several apps are available to automatically catch bugs and enforce style.

✓ This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request You can also open this in GitHub Desktop or view command line instructions.

Write Preview

Leave a comment

Reviewers
No reviews

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Notifications
Unsubscribe

You're receiving notifications because you authored the thread.

1 participant

Merged Pull Request!

kristenkwong / helloworld Private

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 1 Projects 0 Wiki Insights Settings

Merging the test branch! #1

Merged kristenkwong merged 1 commit into master from test just now

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

kristenkwong commented a minute ago

this is a pull request comment

test 2629247

kristenkwong merged commit b7914f9 into master just now

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

Delete branch

Write Preview

Leave a comment

Reviewers: No reviews

Assignees: No one—assign yourself

Labels: None yet

Projects: None yet

Milestone: No milestone

Notifications

Getting changes

`git pull origin master` grabs code from the remote master branch to the local. The remote is updated by developers, so pulling is important!

`git pull` will pull code from all upstream branches.

Onwards!

Foundations 

 Shortcuts!

Shortcuts & Basic Team Workflow

- 1 Make a repo right on GitHub. Share link with team.
- 2 Team members will use `git clone <link>`
- 3 Someone works on stuff. They push with `git push -u origin <branch>` to the remote.
- 4 They will make and merge the pull request.
- 5 Everyone else uses `git pull` to get the changes.
- 6 Repeat Steps 3 - 5 until project is done ✨

That's it!

Where to find me!

✨ Twitter: @kristenkwng

✨ LinkedIn: Kristen Kwong

✨ kristen.dev

✨ Slides: kristen.dev/blog/git

