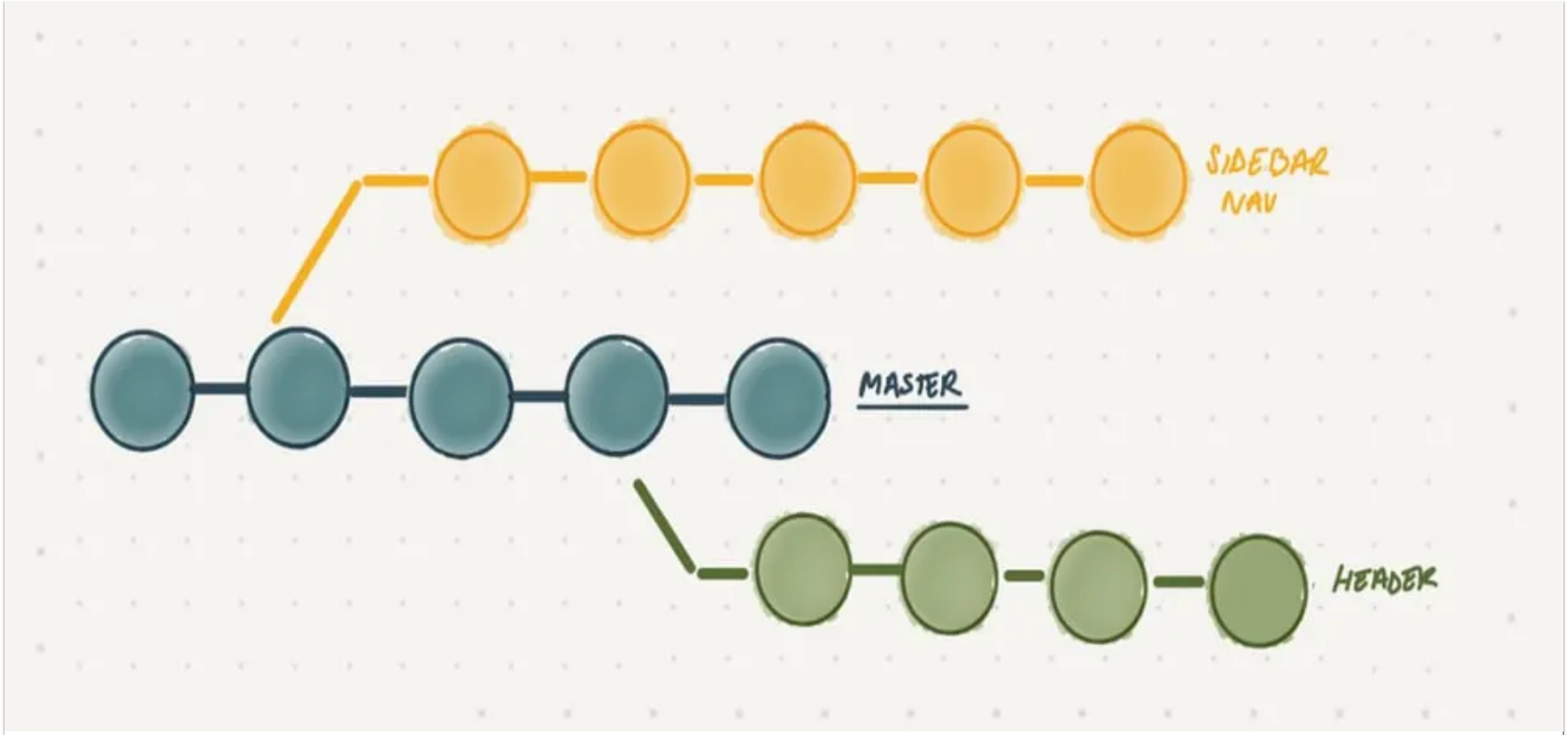


Git Explained: The Basics

Milu May 9 · 3 min read

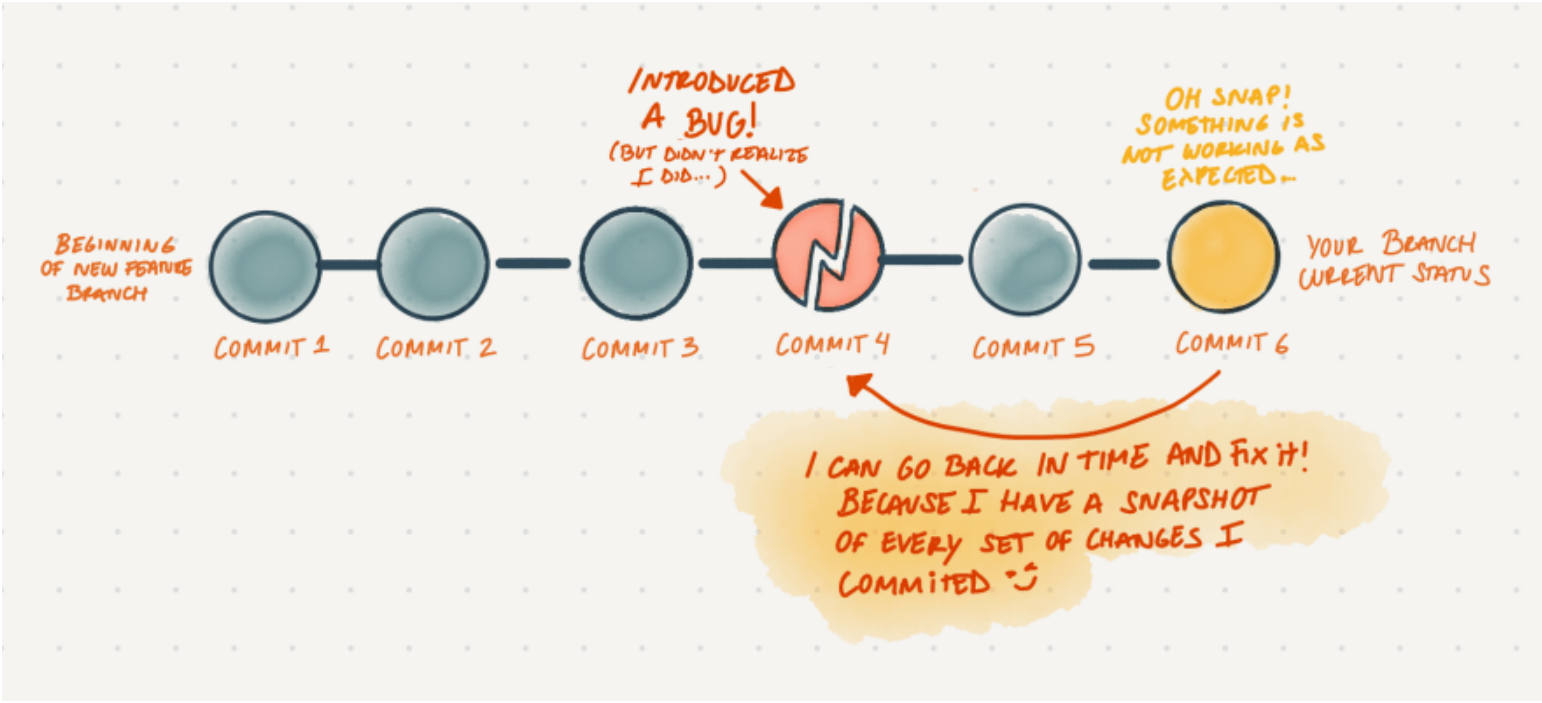
#git #beginners #github

Git is scary when you are starting up. There is this imminent fear that you could possibly lose hours of hard work by using the wrong commands.

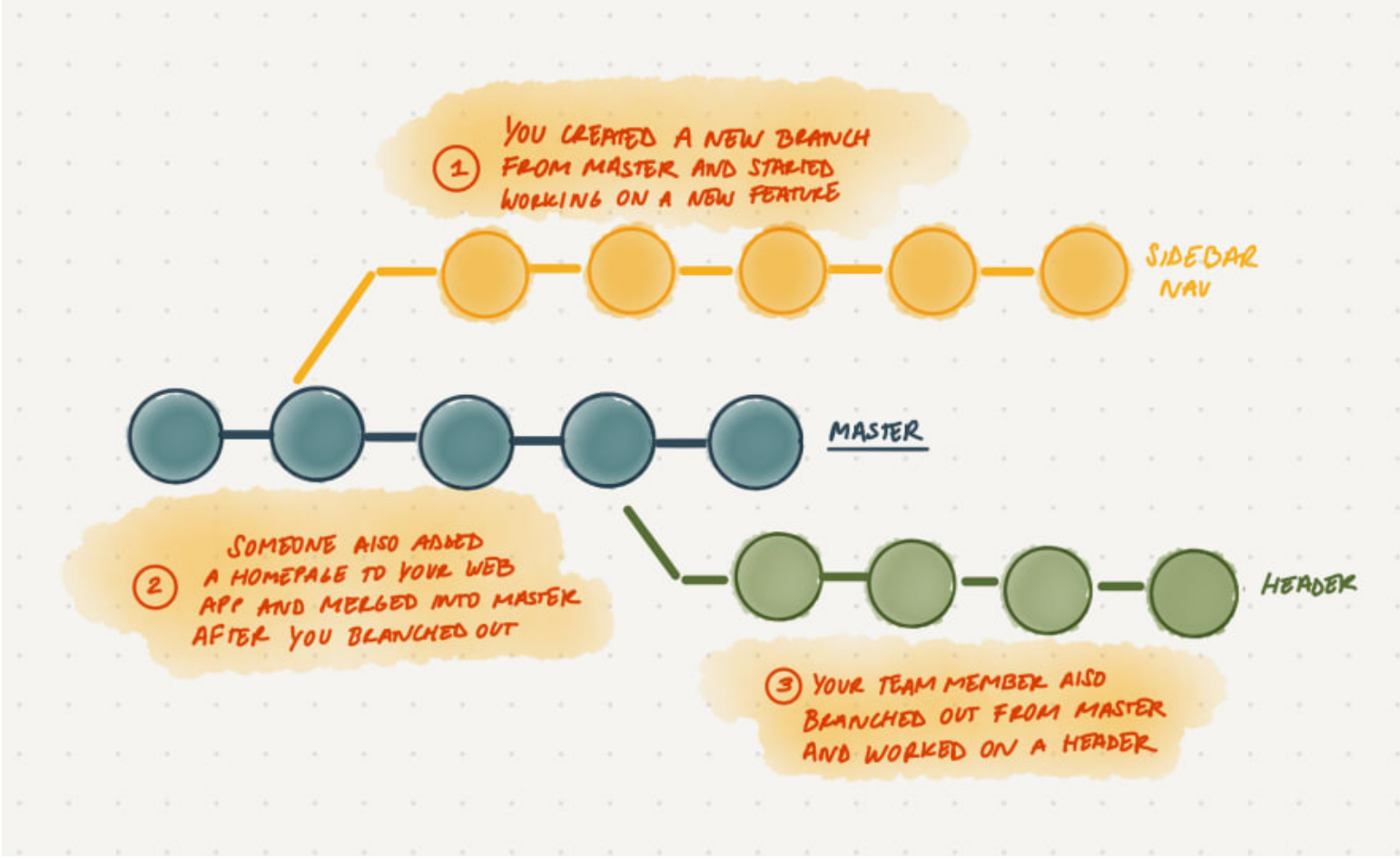
Keep reading if this is how you feel every time you are about to use git. This post is going to define git, explain why you need to learn it, and break down the following basic commands in detail: **clone, checkout, pull, add, commit, stash, and push.**

What is git and why do you need it?

Git is one of the most popular version control systems and it helps software developing teams manage changes to their source code over time. In other words, version control keeps track of every change in your code and it allows you to go back in time when something goes wrong.

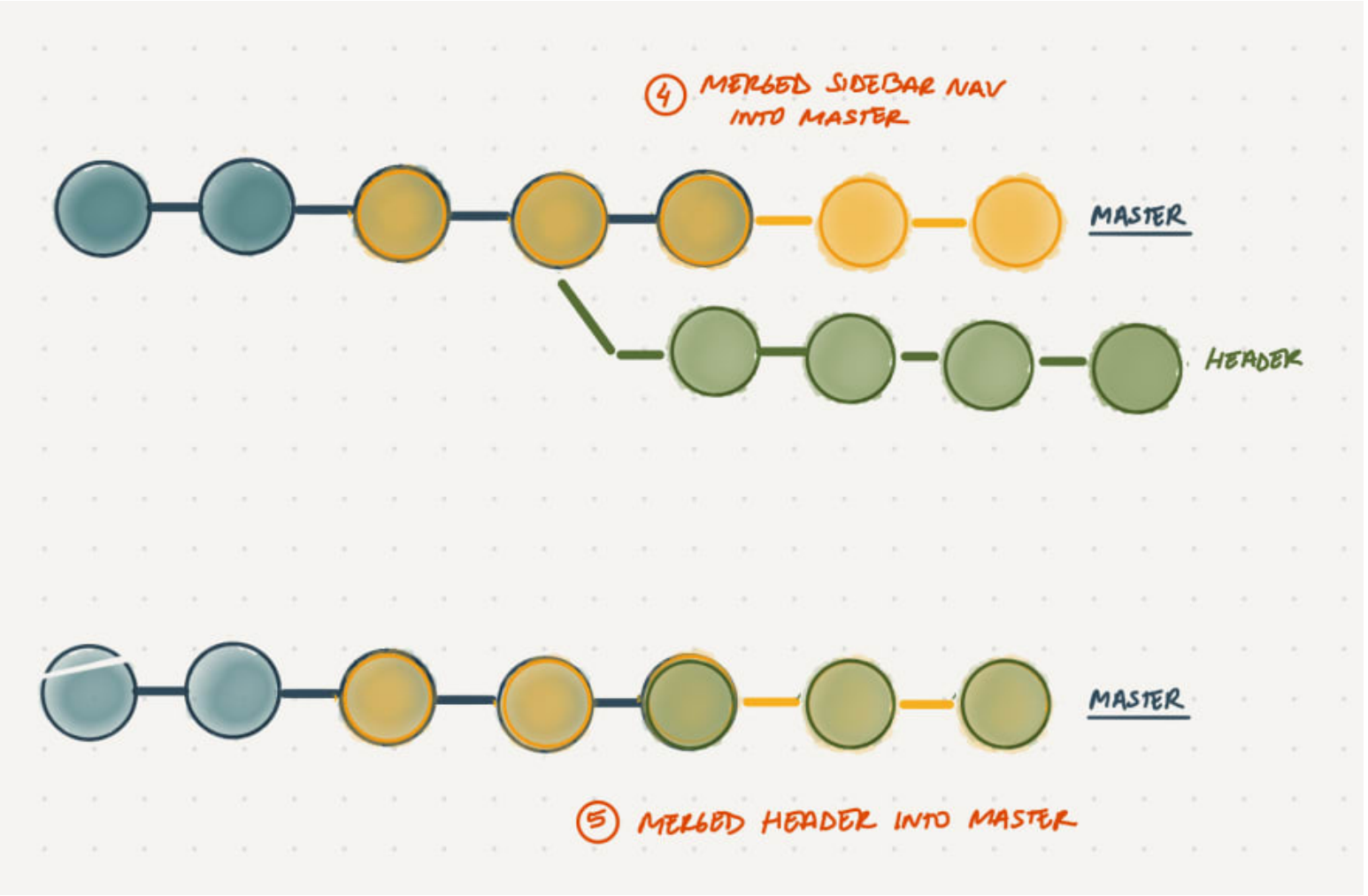


Also, it is really helpful to prevent concurrent work from conflicting when multiple people are working in the same project. An individual may be working on the sidebar navigation while another one is simultaneously updating the header.



Version control systems facilitate the work of multiple individuals by allowing them to use different branches as part of one “file tree” and merge

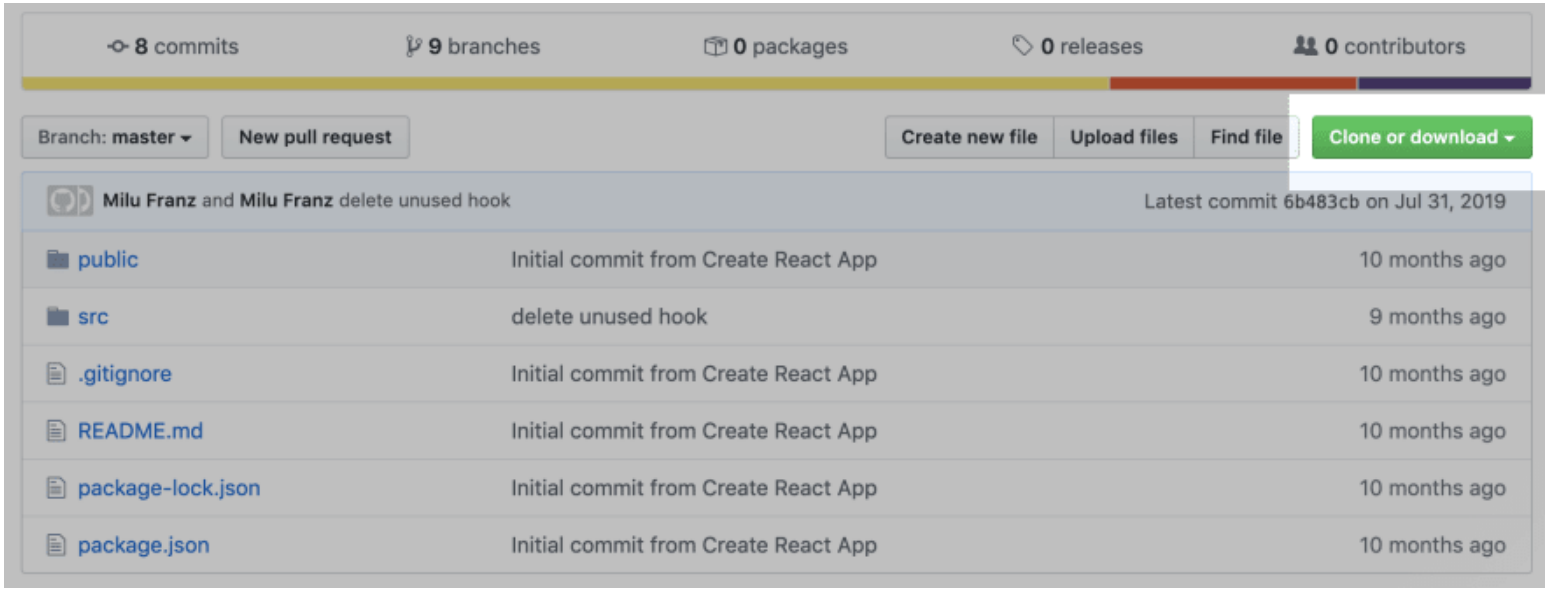
their updated code to one source of truth when it's ready.



Git clone

Downloads a copy of the project (remote repository) to your local computer. To use this command, follow these steps:

1) Copy the clone or download link



2) Open your terminal

3) Access the location on your computer where you want to copy the project:
`cd [desired-location]`

3) Clone the project: `git clone [copied-link]`

4) Use the commands `cd [project-name]`, followed by `ls` and you should see the list of files you just cloned!

Git branch

Lists all your project branches and highlights your current branch. If you just cloned a project, you are going to be in the master branch (your source of truth). It is not advised to work directly on the master branch because there is a big chance you would have a broken project while you are adding some progress. Instead, you want to create a different branch, work on your changes or new feature and when you are done and everything is working as expected, merge that branch into master.

Git checkout

Selects the branch you want to work on.

If the branch already exists:

```
git checkout [branch-name]
```

If you want to create a new branch:

```
git checkout -b [branch-name]
```

Git pull

Pulls recent changes from the remote repository to your local files. If one of your team members merged a new feature into master, you need to pull the recent code added to master to keep your local files up to date.

Git status

Lists all the files that you have worked on and have not been saved in a commit yet.

Git add

Stages file that you would like to commit.

If you want to stage all the files that you have worked on: `git add .`

If you want to stage only one particular file: `git add [file-name]`

Git commit

Saves the group of changes you staged to your local repository. Add a comment with each commit that summarizes the change.

```
git commit -m “commit message”
```

Git stash

Saves any changes you have made locally and it reverts the working directory to match the HEAD of your last commit.

See a list of all the changes you have stashed: `git stash list`

Bring back the last changes you stashed: `git stash pop`

Clear all your stashed entries: `git stash clear`

Git push

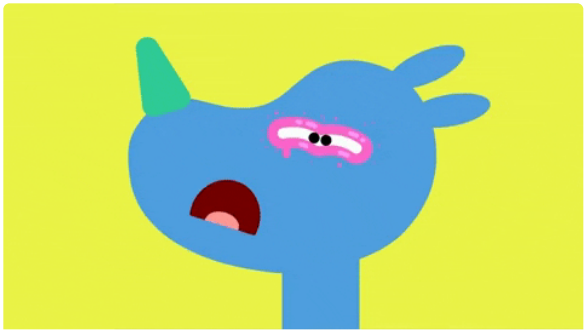
Saves your local committed changes into the remote repository so everybody else has access to your work.

```
git push origin [branch-name]
```

What is next?

This is the first one from a series of posts about git. We will explore more advanced commands, git submodules, multiple remotes, and squashing commits before a pull request in the next weeks.

I hope this information was helpful and you feel a little more confident about working with git!



Milu **+ FOLLOW**
Software Engineer, co-founder of @imagine_dat_dyt and mother of mutts.
@milu_franz milu_franz milufranz08 milufranz08.github.io

Discussion

Add to the discussion

PREVIEW SUBMIT

Arit Amana

May 10

This is AMAZING! Please keep the git tutorials coming 🙏 Your explainations are quite beginner-friendly 😄

2

REPLY

fabiocaettano

May 11

Thank you very much.

Finally I understood how to use the branches!!!

How do I reverse a commit correctly?

1

REPLY


Nick Leake

May 11

Awesome explanations. 🙏

1

REPLY

code of conduct - report abuse

Classic DEV Post from Jul 1 '18

Web Assembly


Akram Saouri

the what, the why and the how

155 3

From one of our Community Sponsors

Feature Flags and GitOps. 5 Use Cases to Help You ‘Git’r Done.


Kristin Baskett

Even if adopting GitOps is still aspirational for your team, you can use CloudBees Rollout to manage your feature flags.

25 2

Another Post You Might Like

Write Merge Requests Like You're Posting to Instagram


Colby Fayock

Merge requests (or pull requests) are a huge part of a team's development process. It's the main gate...

71 4

Another Post You Might Like

How to Contribute to Open Source Software


Matt Eland

If you're anything like me, you want to contribute to open source software but are too intimidated to...

543 25

An Introduction to Browser Extensions

Ramit Mittal - May 10

Implementation of a Graph -JavaScript

codebond - May 10

Lessons In Multi-Step Form With React & Material UI

Max Ong Zong Bao - May 10

Notes on Vue

Axel Uriel Martínez Castillo - May 10