

Git Feature Branch Workflow Cheat Sheet

Learn how to use Git in the context of continuous delivery and software release processes.

Step-by-step guide

This cheat sheet is organized around a **Feature Branch** workflow.

The other cheat sheets I'd seen were organized into basic and advanced commands but not around particular steps in a process. I hope this is more helpful by guiding you to the commands needed in context.

The **Feature Branch workflow** assumes a new branch for each new feature.

The master branch never has broken code. It helps with continuous integration and pull requests (reviews of a feature before integration into the master). The Feature Branch workflow is used in other workflows like Github Flow and Git Flow.

Github Flow is used for continuous delivery of web apps.

Feature branches are merged directly into the master branch, and this triggers deployment to production servers. The master branch is not tagged.

Git Flow is used for software release of desktop apps.

Feature branches are merged into a dev branch. Periodically the dev branch is merged into the master, and the master is tagged with a release number.

The command options and examples in this cheat sheet are not complete or 'generic'. My goal was to show them in the way they are most commonly used in this workflow and others. In a Git Flow or other workflow you may need to change the name of the origin or other repositories.

I drew heavily from 'Git in Practice' by Mike McQuaid, published by Manning Publications. I highly recommend it. Other resources that were useful are listed and linked in the cheat sheet.

By: Mike Combs, <mailto:mcombs964@gmail.com>

[Git Workflow Cheat Sheet](#)



Related articles

- [Git Feature Branch Workflow Cheat Sheet](#)

