<https://guides.github.com/introduction/git-handbook/>

The GitHub flow has six steps, each with distinct benefits when implemented:

1. **Create a branch:** Topic branches created from the canonical deployment branch (usually master) allow teams to contribute to many parallel efforts. Short-lived topic branches, in particular, keep teams focused and results in quick ships.
2. **Add commits:** Snapshots of development efforts within a branch create safe, revertible points in the project’s history.
3. **Open a pull request:** Pull requests publicize a project’s ongoing efforts and set the tone for a transparent development process.
4. **Discuss and review code:** Teams participate in code reviews by commenting, testing, and reviewing open pull requests. Code review is at the core of an open and participatory culture.
5. **Merge:** Upon clicking merge, GitHub automatically performs the equivalent of a local ‘git merge’ operation. GitHub also keeps the entire branch development history on the merged pull request.
6. **Deploy:** Teams can choose the best release cycles or incorporate continuous integration tools and operate with the assurance that code on the deployment branch has gone through a robust workflow.

**Example: Contribute to an existing repository**

# download a repository on GitHub.com to our machine

git clone https://github.com/me/repo.git

# change into the `repo` directory

cd repo

# create a new branch to store any new changes

git branch my-branch

# switch to that branch (line of development)

git checkout my-branch

# make changes, for example, edit `file1.md` and `file2.md` using the text editor

# stage the changed files

git add file1.md file2.md

# take a snapshot of the staging area (anything that's been added)

git commit -m "my snapshot"

# push changes to github

git push --set-upstream origin my-branch