Code changes!


```
# make a new branch
git checkout -b name-
of-new-branch

# as you code, save your work
git add (name of changed
file(s))

git commit -m "good
explanation of your
changes here"

git push
```

Generic workflow

```
# download the code
git clone <SSH URL to
your repository>

# move into the code folder
cd <Name of Git
repository>
```

Why are these tools important?

- create space for code reviews
- enable engineers to all work on the same project simultaneously
- save code in the cloud, so if you spill your morning cold brew, the code will live on
- make it super easy to revert changes if needed

VCS
version control system, tool to track changes to code

GitHub
a product you use to host and manage your Git projects

Git
a version control system you download onto your computer

Need to grab code changes?

```
# switch to the main branch
of the repo
git checkout main

# grab any new code
git pull

# switch back to your branch
git checkout your-branch

# combine the code
git merge main
```

❌ Run into merge conflicts?

To view conflicts in your code editor, search files for the conflict marker <<<<<<<

The code right after <<<<<<< HEAD marks the changes from the base branch, and the chunk of code between the ======= and <<<<<<< BRANCH-NAME is the change on your branch.

Decide which code changes you want to keep, and delete the unwanted code along with the conflict markers.

✅ **Add, commit, push your code**

[hip, hip]!

Time to make a pull request and get feedback from your team

Emma Schreifels

Resources

go/git

go/gitcodeschool

go/mini-git

go/gitdocs

go/gitfix

MiniApp

minis

GIT