  - `git clone git@github.com:USER-NAME/REPOSITORY-NAME.git`
  - `git push` or `git push origin main` (Both accomplish the same goal in this context)
- Commands related to the workflow:
  - `git add .`
  - `git commit -m "A message describing what you have done to make this snapshot different"`
- Commands related to checking status or log history
  - `git status`
  - `git log`

The basic Git syntax is `program | action | destination`.

For example,

- `git add .` is read as `git | add | .`, where the period represents everything in the current directory;
- `git commit -m "message"` is read as `git | commit -m | "message"`; and
- `git status` is read as `git | status | (no destination)`.

Git best practices

There’s a lot to learn about using Git. But it is worth taking the time to highlight some best practices so that you can be a better collaborator. Git is not only helpful when collaborating with others. It’s also useful when working independently. You will be relying more and more on your own commit history in the future when revisiting old code.

Two helpful best practices to consider are **atomic commits** and leveraging those atomic commits to make your commit messages more useful to future collaborators.

An atomic commit is a commit that includes changes related to only one feature or task of your program. There are two main reasons for doing this: first, if something you change turns out to cause some problems, it is easy to revert the specific change