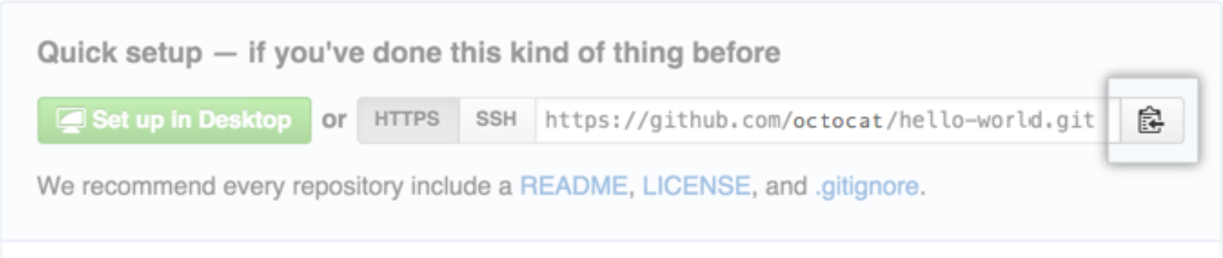
GITHub

go to GitHub.com
create an account
create/name a repository

```
git init
git add .
git commit -m "Initialize Files in ex_files"
git remote add origin https://github.com/acs46/test1.git
git remote -v
git push -u origin master
```

git pull   syncs local repository with remote (ie pulls any recent changes into your local repo)

- Create a new repository on GitHub. To avoid errors, do not initialize the new repository with *README*, license, or gitignore files. You can add these files after your project has been pushed to GitHub.
- Open Terminal.
- Change the current working directory to your local project.
- Initialize the local directory as a Git repository.
  git init
-
- Add the files in your new local repository. This stages them for the first commit.
  git add .
- # Adds the files in the local repository and stages them for commit. To unstage a file, use 'git reset HEAD YOUR-FILE'.
-
- Commit the files that you've staged in your local repository.
  git commit -m "First commit"
- # Commits the tracked changes and prepares them to be pushed to a remote repository. To remove this commit and modify the file, use 'git reset --soft HEAD~1' and commit and add the file again.
-
-



Quick setup — if you've done this kind of thing before

We recommend every repository include a README, LICENSE, and .gitignore.

- At the top of your GitHub repository's Quick Setup page, click to copy the remote repository URL.
- In Terminal, add the URL for the remote repository where your local repository will be pushed.
  git remote add origin remote repository URL
- # Sets the new remote
- git remote -v
- # Verifies the new remote URL
-
- Push the changes in your local repository to GitHub.
  git push -u origin master
- # Pushes the changes in your local repository up to the remote repository you specified as the origin

Keeping the Local Repository in Sync

As we discussed previously, Git doesn’t require an internet connection which means it doesn’t communicate with remote repositories unless explicitly instructed to do so.

When you merge your Pull Request on GitHub.com, the changes you made locally are merged into the master branch on the remote repository on GitHub. However, you won’t automatically see those changes reflected in your local copy until you do a git pull.

Git Pull

Frequently, collaborators will make changes to our files during the Pull Request process. Before we can go on, we will need to update our local copies of the files. Similar to how we use push to send our changes to the remote, we use pull to retrieve changes from the remote. When we pull the files from the remote, Git downloads a copy of the new commits that have been added to the branch since our last pull and then attempts to merge them into our local branch.

You should pull your repository regularly to ensure you are always working with the most recent copies of the files in the repository.