## Fork a repo

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.

Most commonly, forks are used to either propose changes to someone else's project or to use someone else's project as a starting point for your own idea.

##### **Use someone else's project as a starting point for your own idea.**

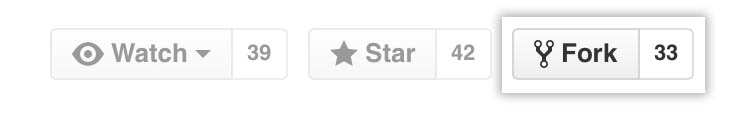
At [the heart of open source](http://opensource.org/about) is the idea that by sharing code, we can make better, more reliable software.

When creating your public repository from a fork of someone's project, make sure to include a [license file](http://choosealicense.com/) that determines how you want your project to be shared with others.

For more information on open source, specifically how to create and grow an open source project, we've created [Open Source Guides](https://opensource.guide/) that will help you foster a healthy open source community by recommending best practices for creating and maintaining repositories for your open source project. You can also take a free [GitHub Learning Lab](https://lab.github.com/) course on maintaining open source communities.

### **Fork an example repository**

Forking a repository is a simple two-step process. We've created a repository for you to practice with!

1. On GitHub, navigate to the [octocat/Spoon-Knife](https://github.com/octocat/Spoon-Knife) repository.
2. In the top-right corner of the page, click **Fork**.

That's it! Now, you have a *fork* of the original octocat/Spoon-Knife repository.

### **Keep your fork synced**

You might fork a project in order to propose changes to the *upstream*, or original, repository. In this case, it's good practice to regularly sync your fork with the upstream repository. To do this, you'll need to use Git on the command line. You can practice setting the upstream repository using the same [octocat/Spoon-Knife](https://github.com/octocat/Spoon-Knife) repository you just forked!

#### **Step 1: Set up Git**

If you haven't yet, you should first [set up Git](https://help.github.com/articles/set-up-git). Don't forget to [set up authentication to GitHub from Git](https://help.github.com/articles/set-up-git" \l "next-steps-authenticating-with-github-from-git) as well.

#### **Step 2: Create a local clone of your fork**

Right now, you have a fork of the Spoon-Knife repository, but you don't have the files in that repository on your computer. Let's create a *clone* of your fork locally on your computer.

1. On GitHub, navigate to **your fork** of the Spoon-Knife repository.
2. Under the repository name, click **Clone or download**.
3. In the Clone with HTTPs section, click  to copy the clone URL for the repository.
4. Open Terminal.
5. 

Type git clone, and then paste the URL you copied in Step 2. It will look like this, with your GitHub username instead of YOUR-USERNAME:

git clone https://github.com/YOUR-USERNAME/Spoon-Knife

1. Press **Enter**. Your local clone will be created.

git clone https://github.com/YOUR-USERNAME/Spoon-Knife

Cloning into `Spoon-Knife`...

remote: Counting objects: 10, done.

remote: Compressing objects: 100% (8/8), done.

remove: Total 10 (delta 1), reused 10 (delta 1)

Unpacking objects: 100% (10/10), done.

Now, you have a local copy of your fork of the Spoon-Knife repository!

#### **Step 3: Configure Git to sync your fork with the original Spoon-Knife repository**

When you fork a project in order to propose changes to the original repository, you can configure Git to pull changes from the original, or *upstream*, repository into the local clone of your fork.

1. On GitHub, navigate to the [octocat/Spoon-Knife](https://github.com/octocat/Spoon-Knife) repository.
2. Under the repository name, click **Clone or download**.
3. In the Clone with HTTPs section, click  to copy the clone URL for the repository.
4. Open Terminal.
5. Change directories to the location of the fork you cloned in [Step 2: Create a local clone of your fork](https://help.github.com/articles/fork-a-repo/" \l "step-2-create-a-local-clone-of-your-fork).
   * To go to your home directory, type just cd with no other text.
   * To list the files and folders in your current directory, type ls.
   * To go into one of your listed directories, type cd your\_listed\_directory.
   * To go up one directory, type cd ...
6. Type git remote -v and press **Enter**. You'll see the current configured remote repository for your fork.

git remote -v

origin https://github.com/YOUR\_USERNAME/YOUR\_FORK.git (fetch)

origin https://github.com/YOUR\_USERNAME/YOUR\_FORK.git (push)

1. Type git remote add upstream, and then paste the URL you copied in Step 2 and press **Enter**. It will look like this:

git remote add upstream https://github.com/octocat/Spoon-Knife.git

1. To verify the new upstream repository you've specified for your fork, type git remote -vagain. You should see the URL for your fork as origin, and the URL for the original repository as upstream.

git remote -v

origin https://github.com/YOUR\_USERNAME/YOUR\_FORK.git (fetch)

origin https://github.com/YOUR\_USERNAME/YOUR\_FORK.git (push)

upstream https://github.com/ORIGINAL\_OWNER/ORIGINAL\_REPOSITORY.git (fetch)

upstream https://github.com/ORIGINAL\_OWNER/ORIGINAL\_REPOSITORY.git (push)

Now, you can keep your fork synced with the upstream repository with a few Git commands. For more information, see "[Syncing a fork](https://help.github.com/articles/syncing-a-fork)."

## Configuring a remote for a fork

You must configure a remote that points to the upstream repository in Git to [sync changes you make in a fork](https://help.github.com/articles/syncing-a-fork) with the original repository. This also allows you to sync changes made in the original repository with the fork.

1. Open Terminal.
2. List the current configured remote repository for your fork.

git remote -v

origin https://github.com/YOUR\_USERNAME/YOUR\_FORK.git (fetch)

origin https://github.com/YOUR\_USERNAME/YOUR\_FORK.git (push)

1. Specify a new remote *upstream* repository that will be synced with the fork.

git remote add upstream https://github.com/ORIGINAL\_OWNER/ORIGINAL\_REPOSITORY.git

1. Verify the new upstream repository you've specified for your fork.

git remote -v

origin https://github.com/YOUR\_USERNAME/YOUR\_FORK.git (fetch)

origin https://github.com/YOUR\_USERNAME/YOUR\_FORK.git (push)

upstream https://github.com/ORIGINAL\_OWNER/ORIGINAL\_REPOSITORY.git (fetch)

upstream https://github.com/ORIGINAL\_OWNER/ORIGINAL\_REPOSITORY.git (push)

## Syncing a fork

Sync a fork of a repository to keep it up-to-date with the upstream repository.

Before you can sync your fork with an upstream repository, you must [configure a remote that points to the upstream repository](https://help.github.com/articles/configuring-a-remote-for-a-fork) in Git.

1. Open Terminal.
2. Change the current working directory to your local project.
3. Fetch the branches and their respective commits from the upstream repository. Commits to master will be stored in a local branch, upstream/master.

git fetch upstream

remote: Counting objects: 75, done.

remote: Compressing objects: 100% (53/53), done.

remote: Total 62 (delta 27), reused 44 (delta 9)

Unpacking objects: 100% (62/62), done.

From https://github.com/ORIGINAL\_OWNER/ORIGINAL\_REPOSITORY

\* [new branch] master -> upstream/master

1. Check out your fork's local master branch.

git checkout master

Switched to branch 'master'

1. Merge the changes from upstream/master into your local master branch. This brings your fork's master branch into sync with the upstream repository, without losing your local changes.

git merge upstream/master

Updating a422352..5fdff0f

Fast-forward

README | 9 -------

README.md | 7 ++++++

2 files changed, 7 insertions(+), 9 deletions(-)

delete mode 100644 README

create mode 100644 README.md

If your local branch didn't have any unique commits, Git will instead perform a "fast-forward":

git merge upstream/master

Updating 34e91da..16c56ad

Fast-forward

README.md | 5 +++--

1 file changed, 3 insertions(+), 2 deletions(-)