



# Adding a remote parent to a forked repository

Git allows developers to [fork](#) and [clone](#) repositories.

The main difference between the two commands is that when you clone a repository you create an exact copy of the “main” repository, to which you’re not likely to have access or permissions to make changes. However when you fork a repository, you create a copy that you own and you can update and get ready for pull requests.

Forking introduces a problem into the equation, do you see?

Because forking creates your own copy of the repository, it is no longer updated from the original. Whatever changes you make are exclusive to your version and whatever changes happened in the original, “main” repository since you forked the code are not available to you, unless...

## Setting up a parent repository

Git allows you to have multiple remote parents for your repository but doesn’t create any outside your own copy by default.

Using Git’s [remote](#) command we can set up links to as many external repositories as we need.

The first command will add a remote repository with the name given (in this case upstream) and pointing to the indicated URL.

```
git remote add upstream https://github.com/ORIGINAL_OWNER/repo.git
```

The second version of the command will show you the remotes currently configured for your project.

```
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/repo.git (fetch)
> upstream  https://github.com/ORIGINAL_OWNER/repo.git (push)
```

You can then update from the parent/master/main repository as needed

```
git fetch upstream
```

And then you can choose whether to merge the upstream code with your own changes and resolve conflicts as necessary following the instructions from [Syncing a fork](#) in Github's documentation...