



# Fork A Repo

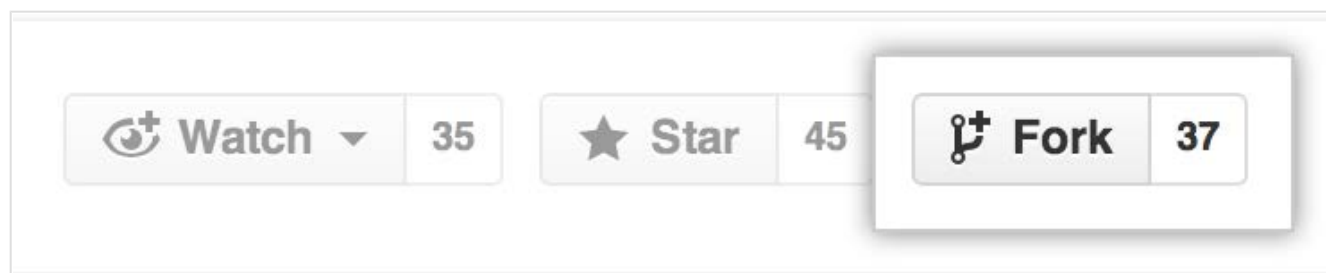
If you've found yourself on this page, we're assuming you're brand new to Git and GitHub. This guide will walk you through the basics and explain a little bit about how everything works along the way.

## Contributing to a project

At some point you may find yourself wanting to contribute to someone else's project, or would like to use someone's project as the starting point for your own. This is known as "forking". For this tutorial, we'll be using the [Spoon-Knife](#) project, hosted on GitHub.com.

### Step 1: Fork the "Spoon-Knife" repository

To fork this project, click the "Fork" button in the GitHub.com repository.



### Step 2: Clone your fork

You've successfully forked the Spoon-Knife repository, but so far it only exists on GitHub. To be able to work on the project, you will need to clone it to your local machine.

Run the following code:

```
$ git clone https://github.com/username/Spoon-Knife.git
# Clones your fork of the repository into the current directory in terminal
```

### Step 3: Configure remotes

When a repository is cloned, it has a default remote called `origin` that points to your fork on GitHub, not the original repository it was forked from. To keep track of the original repository, you need to add another remote named `upstream`:

More about remotes

```
$ cd Spoon-Knife
# Changes the active directory in the prompt to the newly cloned "Spoon-Knife"
directory
$ git remote add upstream https://github.com/octocat/Spoon-Knife.git
# Assigns the original repository to a remote called "upstream"
$ git fetch upstream
# Pulls in changes not present in your local repository, without modifying your
files
```

## More Things You Can Do

You've successfully forked a repository, but get a load of these other cool things you can do:

### Push commits



Once you've made some commits to a forked repository and want to push it to your forked project, you do it the same way you would with a regular repository:

 More about commits

```
$ git push origin master
# Pushes commits to your remote repository stored on GitHub
```

### Pull in upstream changes

If the original repository you forked your project from gets updated, you can add those updates to your fork by running the following code:

```
$ git fetch upstream
# Fetches any new changes from the original repository
$ git merge upstream/master
# Merges any changes fetched into your working files
```

 What is the difference between fetch and pull?

### Create branches

Branching allows you to build new features or test out ideas without putting your main project at risk. In git, branch is a sort of bookmark that references the last commit made in the branch. This makes branches very small and easy to work with.

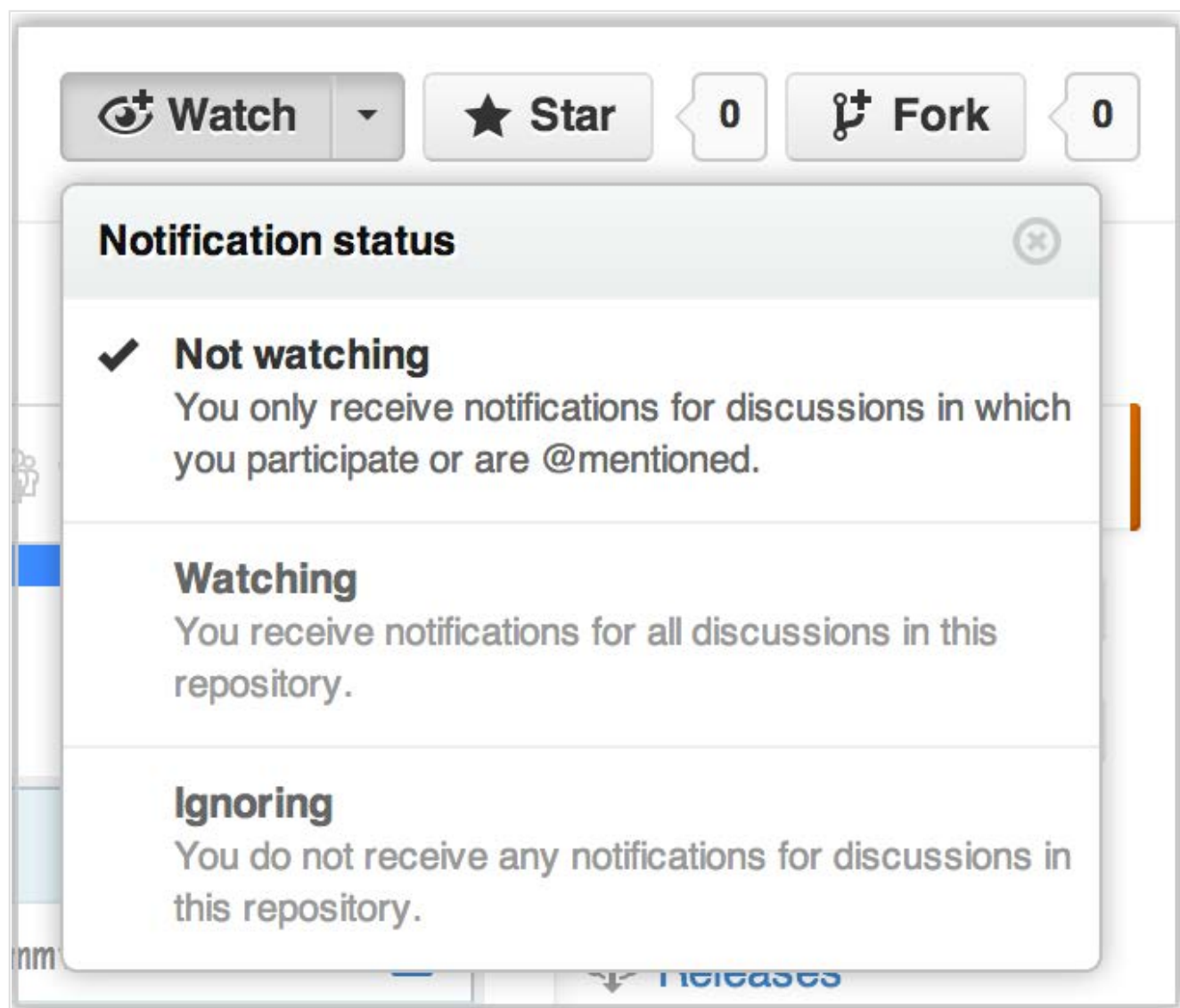
 How do I use branches?

### Pull requests

If you are hoping to contribute back to the original fork, you can send the original author a [pull request](#).

## Unwatch the main repository

When you fork a particularly popular repository, you may find yourself with a lot of unwanted updates about it. To unsubscribe from updates to the main repository, click the "Unwatch" button on the **main repository** and select "Not Watching".



## Delete your fork

At some point you may decide that you want to delete your fork. To delete a fork, just follow the same steps as you would to [delete a regular repository](#).

## Celebrate

You have now forked a repository. What do you want to do next?

- › [Set Up Git](#)
- › [Create A Repository](#)
- › **Fork A Repository**
- › [Be Social](#)

---

 [contact a human](#)

© 2014 GitHub Inc. All rights reserved.



[Terms of Service](#) [Privacy](#) [Security](#)