

Beginner

**Getting Started** 

Collaborating Syncing

git remote

git fetch

git push

git pull

Making a Pull Request Using branches Comparing workflows

Migrating to Git

Advanced Tips

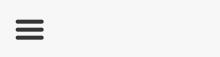


# git fetch

git remote / git fetch / git push / git pull

The git fetch command downloads commits, files, and refs from a remote repository into your local repo. Fetching is what you do when you want to see what everybody else has been working on. It's similar to svn update in that it lets you see how the central history has progressed, but it doesn't force you to actually merge the changes into your repository. Git isolates fetched content as a from existing local content, it has absolutely no effect on your local development work. Fetched content has to be explicitly checked out using the gitcheckout command. This makes fetching a safe way to review commits before integrating them with your local repository.

When downloading content from a remote repo, git pull and git fetch commands are available to



Beginner

**Getting Started** 

### Collaborating Syncing

git remote

git fetch

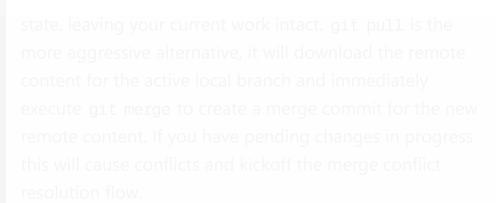
git push

git pull

Making a Pull Request
Using branches
Comparing workflows

Migrating to Git

Advanced Tips

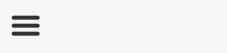


# How git fetch works with remote branches

To better understand how git fetch works let us discuss how Git organizes and stores commits. Behind the scenes in the repository's ./.git/objects directory, Git stores all commits, local and remote. Git keeps remote and local branch commits distinctly separate through the use of branch refs. The refs for local branches are stored in the ./.git/refs/heads/. Executing the gitbranch command will output a list of the local branch refs. The following is an example of git branch output with some demo

git branch master featurel debug2

Examining the contents of the /.git/refs/heads/directory would reveal similar output.







l parn Git

Beginner

**Getting Started** 

# Collaborating Syncing

git remote

git fetch

git push

git pull

Making a Pull Request
Using branches
Comparing workflows

Migrating to Git

Advanced Tips



merge. The git pull command is a convenient shortcut for this process.

# Git fetch commands and options

git fetch <remote>

Fetch all of the branches from the repository. This also downloads all of the required commits and files from the other repository.

git fetch <remote> <hranch>

Same as the above command, but only fetch the specified branch

git fetch --all

A power move which fetches all registered remotes and their branches:

git fetch --dry-run



Beginner

**Getting Started** 

### Collaborating Syncing

git remot git fetch git push

Making a Pull Request
Using branches

Migrating to Git

Advanced Tips

# Git fetch examples

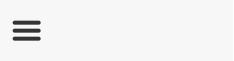
## git fetch a remote branch

The following example will demonstrate how to fetch a remote branch and update your local working state to the remote contents. In this example, lets assume there is a central repo origin which the local repository has been cloned from using the git clone command. Let us also assume an additional remote repository named coworkers\_repo that contains a feature\_branch which we will configure and fetch. With these assumptions set let us continue the example.

Firstly we will need to configure the remote repo using the gitremote command.

git remote coworkers\_repo git@bitbucket.org:cov

Here we have created a reference to the coworker's repousing the repo URL. We will now pass that remote name to git fetch to download the contents.



Beginner

**Getting Started** 

# Collaborating Syncing

git remotes

git push

laking a Pull Requessing branches

Comparing workflows

Migrating to Git

Advanced Tips



```
git checkout coworkers/feature_branch
Note: checking out coworkers/feature_branch'.

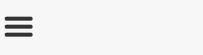
You are in 'detached HEAD' state. You can look changes and commit them, and you can discard as state without impacting any branches by perform

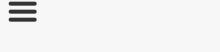
If you want to create a new branch to retain condo so (now or later) by using -b with the check git checkout -b <new-branch-name>
```

The output from this checkout operation indicates that we are in a detached HEAD state. This is expected and means that our HEAD ref is pointing to a ref that is not in sequence with our local history. Being that HEAD is pointed at the coworkers/feature\_branch ref, we can create a new local branch from that ref. The 'detached HEAD' output shows us how to do this using the git checkout command:

```
git checkout -b local_feature_branch
```

Here we have created a new local branch named ocal\_feature\_branch this puts updates HEAD to point at







Enter Your Email For Git News

Q

Learn Git

Beginner

**Getting Started** 

Collaborating
Syncing

git remote

giticita

ait null

Making a Pull Request
Using branches
Comparing workflows

Migrating to Git

Advanced Tips

git checkout master git log origin/master

Then we can use git merge origin/master:

git merge origin/master

The origin/master and master branches now point to the same commit, and you are synchronized with the upstream developments.

## Git fetch summary

In review, git fetch is a primary command used to download contents from a remote repository. git fetch is used in conjunction with git remote, git branch, git checkout, and gitreset to update a local repository to the state of a remote. The git fetch command is a critical piece of collaborative git work flows. git fetch has similar behavior to git pull however git fetch can be considered a safer, nondestructive version.

#### ATLASSIAN Bitbucket

### dir basii

#### START NEXT TUTORIAL

Beginner

Learn Git

**Getting Started** 

### Collaborating Syncing

git fetch

git pull

Making a Pull Request

**Using branches** 

**Comparing workflows** 

Migrating to Git

**Advanced Tips** 

Powered By

Atlassian

Recommend

Want future articles?

Site hosted by

Enter Your Email For Git News



Except where otherwise noted, all content is licensed under a Creative Commons Attribution 2.5 Australia License.