

Key concepts

HEAD	Head is your current branch. You can see what HEAD points to by typing <code>cat .git/HEAD,</code>
Remote	Remotes are non-local repositories you can interact with (push/pull). The default remote is <i>origin</i> (you can see that using <code>git remote -v</code>).
Branch	Branches are a way of safely work on new features without messing other peoples work (one feature, one branch).
Commit	A commit is a change or a set of changes you wish to register and save.

Repositories

Create a local repository	<code>mkdir ./my_repo && cd ./my_repo && git init</code>
Clone a repository from GitHub	<code>git clone https://github.com/TME520/etm.git</code>
Clone a specific branch from a repo	<code>git clone -b development https://github.com/TME520/etm.git</code>

Branches

List branches	<code>git branch</code>	<code>git branch --list</code>
Clone a specific branch	<code>git clone -b <branch> <remote_repo></code>	<code>git clone -b development git@github-b.com:user/-myproject.git</code>
Switch to an existing branch	<code>git checkout <branch></code>	<code>git checkout feat/robert-add-pub-holidays-2020</code>

Branches (cont)

Switch to a new branch	<code>git checkout -b <new_branch></code>	<code>git checkout -b conf/marcel-use-EditorConfig</code>
Push changes to a remote branch	<code>git push origin <branch></code>	<code>git push origin fix/patrick-remove-parasite-chars-from-config-file</code>
Delete a branch	<code>git branch -d <branch></code>	<code>git branch -d feat/raja/count-api-failures</code>
Integrate your feature branch to the main	<code>git merge <branch></code>	<code>git merge main</code>

Using branches, several developers are able to work together on the same code base, the same project.

git merge is not usually done manually, but is managed by your pull request system.

Getting out of (mild) troubles

Cancel uncommitted local changes	<code>git reset --hard && git pull remote <remote_branch></code>
---	--

Basic configuration

Set your name	<code>git config --global user.name "John Doe"</code>
Set your email address	<code>git config --global user.email johndoe@example.com</code>
Set your default editor	<code>git config --global core.editor emacs</code>



By **TME520** (TME520)
cheatography.com/tme520/tme520.com

Published 26th April, 2020.
 Last updated 2nd April, 2021.
 Page 1 of 2.

Sponsored by **ApolloPad.com**
 Everyone has a novel in them. Finish Yours!
<https://apollopad.com>

Dot files

<code>.gitignore</code>	Specifies intentionally untracked files that Git should ignore. Files already tracked by Git are not affected.
<code>.gitattributes</code>	Gives attributes (end of line type, diff type, merge type...) to certain files.

Examples of `.gitignore` files: [click here](#)

Examples of `.gitattributes` files: [click here](#)

Basic workflow

Clone a repo	<code>git clone https://github.com/TME520/etm.git</code>
Create a new branch	<code>git checkout -b conf/you/add-api-endpoints-to-monitoring</code>
Check repo status and current branch	<code>git status</code>
Add changes to next commit (track files)	<code>git add -A</code>
Commit with a message	<code>git commit -m "Sometimes dogs are grey"</code>
Refresh current local branch with remote branch	<code>git pull origin development</code>
Push changes to remote	<code>git push origin conf/you/add-api-endpoints-to-monitoring</code>

git push uploads your commits to the remote repository.

git pull is a combination of `git fetch` and `git merge`. It gets the updates from remote repository and applies the latest changes to your local.

Commits

Add one file to a future commit	<code>git add <file></code>	<code>git add benchm ark.c</code>
Add all files to a future commit	<code>git add -A</code>	
Commit with a message	<code>git commit -m "<message>"</code>	<code>git commit -m "Initial commit"</code>
Change the message of the latest commit	<code>git commit --amend -m "<new_message>"</code>	<code>git commit --amend -m "Beautiful commit"</code>
Cancel a commit	<code>git log --oneline</code> <code>git revert <commit_id></code>	

Git commits are checkpoints in the development process which you can go back to later if needed.

Git commit saves your changes only locally.

git revert won't delete the commit, it will instead create a new one cancelling the other.

