

# Getting Started with Git and GitHub

## Module 2 Cheat Sheet: Git Commands and Managing GitHub Projects

Package/Method	Description	Code Example
<b>git add</b>	Used to move changes from the working directory to the staging area	<code>git add sample.md</code>
<b>git add .</b>	Allows to move the changed files into the staging area on GitHub repositories	<code>git add .</code>
<b>git am</b>	Used to apply patches emailed to the repository	<code>git am &lt; patchfile.patch</code>
<b>git branch</b>	Allows to create an isolated environment within the repository to make changes	<code>git branch &lt;new-branch&gt;</code>
<b>git checkout</b>	Allows to see and change existing branches	<code>git checkout &lt;existing-branch&gt;</code>
<b>git checkout main</b>	Allows to switch to the main branch	<code>git checkout main</code>
<b>git clone</b>	Allows to create a copy of the remote repository	<code>git clone &lt;repository-url&gt;</code>
<b>git commit</b>	Allows you to take staged snapshots if changes and commit them to the project	<code>git commit -m "Your commit message here"</code>
<b>git config --global user.email</b>	<p>Example 1: Sets a global email configuration for Git</p> <p>Example 2: Sets a global username configuration for Git</p>	<p>Example 1:</p> <pre>git config --global user.email "your.email@example.com"</pre> <p>Example 2:</p> <pre>git config --global user.name "Your Name"</pre>
<b>git daemon</b>	Used to allow anonymous download from the repository	<code>git daemon --reuseaddr --verbose</code>
<b>git diff</b>	Helps others to review your code to identify and compare the changes	<code>git diff example.txt</code>
<b>git fetch</b>	Used to transfer the changes from the remote repo to your local repo	<code>git fetch &lt;options&gt; &lt;remote name&gt; &lt;branch name&gt;</code>
<b>git fetch upstream/master</b>	Used to grab upstream branches	<code>git fetch upstream master:upstream-master</code>
<b>git format-patch</b>	Generates or prepares e-mail submission if you adopt Linux kernel-style public forum workflow	<code>git format-patch -n &lt;number_of_commits&gt;</code>
<b>git http-backend</b>	Provides a server-side implementation of Git-over-HTTP, allowing both fetch and push services	<pre>git clone --bare /path/to/repos/myrepo.git cd myrepo.git git update-server-info</pre>
<b>git init</b>	Used to clone an existing repository	<code>git init &lt;directory&gt;</code>
<b>git instaweb</b>	Allows to set up web front-end to Git repositories	<code>git instaweb -p 8080</code>
<b>git log</b>	Enables to browse previous changes to a project	<code>git log -p filename</code>
<b>git merge</b>	Used to merge changes in the active branch into another branch	<code>git merge feature_branch</code>
<b>git merge upstream/master</b>	Merges changes from the 'upstream/master' branch to the current branch	<code>git merge upstream/master</code>
<b>git pull</b>	Used to transfer the changes from the remote repo to your local repo, and merge them to a branch	<code>git pull origin main</code>
<b>git pull downstream</b>	Pulls changes from a downstream repository, specifically from the master branch of that repository	<code>git pull downstream main</code>

Package/Method	Description	Code Example
<b>git pull upstream</b>	Pulls changes from the "upstream" repository into the current branch	<code>git pull upstream main</code>
<b>git push</b>	Used to push all the committed changes into the repository	<code>git push origin your_branch_name</code>
<b>git remote</b>	A command to manage a set of tracked repositories	<code>git remote add upstream https://github.com/original/repo.git</code>
<b>git remote add origin &lt;URL&gt;</b>	Adds a remote repository named "origin" with the specified URL	<code>git remote add origin https://github.com/yourusername/your-repo.git</code>
<b>git remote add upstream</b>	Adds the original repository as a new remote repository labeled upstream	<code>git remote add upstream https://github.com/original/repo.git</code>
<b>git remote rename</b>	The git remote rename command is followed by the name of the remote repository(origin) you want to rename and the new name(upstream) you want to give it	<code>git remote rename origin new-origin</code>
<b>git remote -v</b>	Allows to view the remotes associated with the local repository	<code>git remote -v</code>
<b>git request-pull</b>	<p>Example 1: Creates a summary of changes for your upstream to pull</p> <p>Example 2: Generates a summary of pending changes for an email request</p>	<p>Example 1:</p> <pre>git request-pull origin/main your-branch</pre> <p>Example 2:</p> <pre>git request-pull &lt;base&gt; &lt;head&gt; &lt;repository&gt;</pre>
<b>git rerere</b>	Reuses recorded resolution of previously resolved merge conflicts	<pre>git rerere git rerere diff</pre>
<b>git reset</b>	Undoes changes that were made to the files in your working directory	<code>git reset HEAD~1</code>
<b>git revert</b>	Used to undo botched commits	<code>git revert HEAD</code>
<b>git send-email</b>	<p>Example 1: Sends your email submission without corruption by your MUA</p> <p>Example 2: Sends a collection of patches as emails</p>	<p>Example 1:</p> <pre>git send-email --to=recipient@example.com path/to/patchfile.patch</pre> <p>Example 2:</p> <pre>git send-email --to recipient@example.com patches/*.patch</pre>
<b>git-shell</b>	Used as a restricted login shell for shared central repository users	<code>sudo usermod -s /usr/bin/git-shell gituser</code>
<b>git status</b>	Allows to see the state of your working directory and the staged snapshot of the changes	<code>git status</code>
<b>git version</b>	Displays the current Git version installed on your system	<code>git --version</code>
<b>git web</b>	Provides a web front-end to Git repositories	<code>git instaweb --port=8080</code>



# Skills Network