10 Most Commonly Used Git Commands

FRC Team 3512, Spartatroniks

1. Cloning a remote repository

```
git clone <URL> <folder name>
```

- This copies the repository at the given URL to the current directory.
  - Example: `git clone https://github.com/Team3512/DriverStationDisplay.git`
  - Example: `git clone git@github.com:Team3512/DriverStationDisplay.git`

- If <folder name> is specified the repository's folder is given a different name.
  - Example: `git clone https://github.com/Team3512/DriverStationDisplay.git DSDisplay`

2. Adding files to the staging area

```
git add <file name>
```

- This adds new, untracked files as well as changes to already tracked ones.
  - Example: `git add OperatorControl.cpp`

3. Removing files from the staging area

```
git rm [--cached] <file name>
```

- This deletes files and stops them from being tracked.
  - Example: `git rm UnneededFile.hpp`

- If the "--cached" option is provided, it just removes the specified file's changes from the staging area.
  - Example: `git rm –cached OperatorControl.cpp`

4. Creating a commit in a local repository

```
git commit -m "<commit message>"
```

- This commits changes currently in the staging area.
  - Example: `git commit -m "Added new feature x"`

- If "git commit" is called without the -m flag, an editor will be opened for entering a commit message.

5. Pushing commits to a remote repository

```
git push [<branch name>]
```

- This pushes commits from a branch to the corresponding branch on a remote repository.
  - Example: `git push myBranch`

- If <branch name> isn't specified, the branch that is currently checked out will be used.

6. Pulling commits from a remote repository

```
git pull [<remote branch>]
```

- This pulls commits from remote repository's branch and merges them into the current branch.
  - Example: `git pull origin/master`
- If <remote branch> isn’t specified, git will attempt to pull changes from a remote branch with the same name as the one currently checked out.

7. Creating a branch

```
git branch <branch name>
```

- This creates a new branch starting at the current commit.
  - Example: `git branch newFeature`

8. Deleting a branch

```
git branch -d <branch name>
```

- This deletes the branch <branch name>.
  - Example: `git branch -d mergedFeature`
- This command won’t work if that branch is currently checked out.

9. Checking out a branch

```
git checkout <branch name>
```

- This checks out <branch name>.
  - Example: `git checkout otherBranch`
- The default branch when a new project is created is called “master”.

10. Merging a branch

```
git merge <branch name>
```

- This pulls commits from the specified branch and merges them into the branch currently checked out.
  - Example: `git merge newFeature`
- In some cases, one may want to delete the branch after merging it (see 8).