Git-Commands

March 13, 2024

ragu8

1 Initialization:

- `git init`: Initializes a new Git repository in the current directory.

2 Configuration:

- `git config`: Configures Git settings, either globally or per repository.
  - `git config --global [setting] [value]`: Sets a global configuration.
  - `git config [setting] [value]`: Sets a configuration for the current repository.
  - `git config --list`: Lists all Git configurations.

3 Cloning:

- `git clone [repository URL]`: Clones a repository from a remote source to your local machine.

4 Local Changes:

- `git add [file]`: Adds a file or directory to the staging area.
- `git commit -m "Commit message"`: Commits changes in the staging area with a message.
- `git diff`: Shows changes between commits, commit and working tree, etc.
- `git status`: Shows the status of changes as untracked, modified, or staged.

5 Branching:

- `git branch`: Lists all branches in the repository.
- `git branch [branch_name]`: Creates a new branch.
- `git checkout [branch_name]`: Switches to the specified branch.
- `git merge [branch_name]`: Merges the specified branch into the current branch.

6 Remote Repositories:

- `git remote`: Manages remote repositories.
  - `git remote add [name] [url]`: Adds a new remote repository.

  - `git remote -v`: Lists all remote repositories.
- `git fetch [remote]`: Fetches changes from a remote repository.
- `git pull [remote] [branch]`: Fetches and merges changes from a remote repository.
- `git push [remote] [branch]`: Pushes changes to a remote repository.

7 History:

- `git log`: Shows commit logs.
- `git show [commit]`: Shows information about a commit.
- `git blame [file]`: Shows who last modified each line of a file and when.

8 Undoing Changes:

- `git reset [file]`: Unstages changes in the staging area.
- `git checkout -- [file]`: Discards changes in the working directory.
- `git revert [commit]`: Creates a new commit that undoes changes introduced by a specific commit.

9 Tagging:

- `git tag [tag_name]`: Creates a new tag at the current commit.
- `git tag -l`: Lists all tags in the repository.

10 Collaboration:

- `git fetch`: Fetches changes from a remote repository.
- `git pull`: Fetches and merges changes from a remote repository.
- `git push`: Pushes changes to a remote repository.
- `git merge`: Merges changes from one branch to another.
- `git rebase`: Applies changes from one branch to another by reapplying each commit.

11 Submodules:

- `git submodule`: Manages submodules within the repository.

12 Miscellaneous:

- `git stash`: Temporarily shelves changes.
- `git fsck`: Verifies the integrity of the Git filesystem.
- `git grep`: Searches the contents of files in a Git repository.
- `git bisect`: Finds the commit that introduced a bug.
- `git revert --abort`: Cancels a revert operation.
- `git cherry-pick [commit]`: Applies the changes introduced by a specific commit onto the current branch.
- `git rebase -i [commit]`: Interactively rebase commits, allowing you to squash, edit, or reorder them.

- `git branch -d [branch]`: Deletes a specified branch.
- `git clean`: Removes untracked files from the working directory.
- `git config --unset [setting]`: Removes a configuration setting.
- `git remote set-url [remote_name] [new_url]`: Changes the URL of a remote repository.
- `git gui`: Opens a graphical user interface for Git.
- `git add -i`: Interactively choose which changes to stage.
- `.gitignore`: A file used to specify intentionally untracked files to ignore.
- `git commit --amend`: Amends the last commit with new changes or a new commit message.
- `git bisect start`: Starts the process of binary search to find the commit that introduced a bug.
- `git bisect good [commit]`: Marks a commit as good (bug-free).
- `git bisect bad [commit]`: Marks a commit as bad (contains the bug).
- `git bisect reset`: Ends the bisect process and returns to the original HEAD.
- `git archive [branch]`: Creates a tarball from a specified branch.