TortoiseGit Cheat Sheet

A comprehensive cheat sheet for using TortoiseGit, a Windows shell extension for Git, covering essential operations, commands, and best practices for version control.

Basic Operations

Cloning a Repository

Cloning downloads a remote repository to your local machine.

1. Right-click in the folder where you want to store the repository.
2. Select `Git Clone...`.
3. Enter the URL of the remote repository.
4. Choose the directory for the local repository.
5. Click `OK`.

Example: Cloning from GitHub:

Repository URL: `https://github.com/user/repo.git`
Directory: `C:\Users\User\Documents\repo`

Committing Changes

Committing saves changes to your local repository.

1. Right-click on the file(s) or folder you want to commit.
2. Select `Git Commit -> "master"...`.
3. Enter a descriptive commit message.
4. Click `Commit`.

Best practice: Write clear, concise commit messages explaining *why* the changes were made, not just *what* was changed.

Pushing Changes

Pushing uploads your local commits to the remote repository.

1. Right-click on the folder of your local repository.
2. Select `TortoiseGit -> Push...`.
3. Ensure the correct remote and branch are selected.
4. Click `OK`.

Important: Make sure you have committed your changes before pushing.

Branching and Merging

Creating a Branch

Branching allows you to work on new features or bug fixes in isolation.

1. Right-click on the folder of your local repository.
2. Select `TortoiseGit -> Create Branch...`.
3. Enter the name of the new branch.
4. Click `OK`.

Tip: Choose descriptive branch names related to the feature or issue you're addressing (e.g., `feature/new-login`, `bugfix/issue-123`).

Switching Branches

Switching Branches moves your working directory to a different branch.

1. Right-click on the folder of your local repository.
2. Select `TortoiseGit -> Switch/Checkout...`.
3. Select the branch you want to switch to.
4. Click `OK`.

Note: Ensure you have committed or stashed any uncommitted changes before switching branches to avoid conflicts.

Merging Branches

Merging integrates changes from one branch into another.

1. Switch to the target branch (e.g., `master`).
2. Right-click on the folder of your local repository.
3. Select `TortoiseGit -> Merge...`.
4. Select the branch you want to merge into the current branch.
5. Click `OK`.

Conflict Resolution: If conflicts arise during the merge, TortoiseGit will prompt you to resolve them. Use a merge tool (e.g., TortoiseMerge) to compare and edit the conflicting files.

Advanced Features

Stashing Changes

Stashing temporarily shelves changes you've made so you can work on something else, and then come back and re-apply them later.

1. Right-click on the folder of your local repository.
2. Select `TortoiseGit -> Stash Save...`.
3. Enter a name or description for the stash.
4. Click `OK`.

To re-apply stashed changes: Right-click on the folder, select `TortoiseGit -> Stash Pop...`, and choose the stash you want to apply.

Reverting Changes

Reverting undoes changes to a specific file or commit.

- **Revert a File:** Right-click on the file, select `TortoiseGit -> Revert...`.
- **Revert a Commit:** Use `TortoiseGit -> Show Log`, right-click on the commit to revert, and select `Revert this commit`.

Caution: Reverting a commit creates a *new* commit that undoes the changes. It doesn't erase the original commit history.

Resolving Conflicts

Resolving Conflicts occurs when Git cannot automatically merge changes from different branches.

1. Identify conflicted files (marked with a conflict icon).
2. Right-click on the conflicted file and select `TortoiseGit -> Edit Conflicts`.
3. Use the merge tool to compare and resolve the differences.
4. Mark the file as resolved after making the necessary changes.
5. Commit the resolved file.

Tip: Communicate with your team to understand the changes and agree on the best way to resolve conflicts.

Troubleshooting

Common Issues

Problem: Cannot push changes due to remote changes.

Solution: Pull the remote changes first using `TortoiseGit -> Pull...` then try pushing again. Resolve any conflicts if necessary.

Problem: Working directory is dirty (uncommitted changes).

Solution: Commit or stash your changes before switching branches or performing other operations.

Error Messages

Message: "fatal: refusing to merge unrelated histories"

Cause: Attempting to merge branches with completely different histories.

Solution: Use the `--allow-unrelated-histories` option (though this may indicate a larger issue in your workflow).

Message: "Your branch is behind 'origin/master'"

Cause: Your local branch is out of sync with the remote.

Solution: Pull the latest changes from the remote.

Useful Commands (Git Bash)

- `git status`: Shows the status of your working directory.
- `git log`: Displays the commit history.
- `git diff`: Shows the differences between commits, branches, etc.
- `git branch`: Lists, creates, or deletes branches.
- `git remote -v`: Shows the remote repositories.