Ultimate Git Commands Reference

Advanced Git Internals and Features

Git Internals Overview

Git is a distributed version control system that uses snapshots and SHA-1 hashes.

Key Concepts:

- Repository (.git): Stores all version history and config.

- Working Directory: The actual files you're editing.

- Staging Area (Index): Prepares changes to be committed.

- HEAD: Points to the latest commit on the current branch.

Git Object Model

Git stores 4 object types:

1. Blob - Stores file data.

2. Tree - Directory listing.

3. Commit - Snapshot of tree with metadata.

4. Tag - Named pointer to a commit.

Each object is identified by a SHA-1 hash.

Ref and HEAD Explained

- HEAD points to the current branch reference.

- A branch is a pointer to a commit.

Use `cat .git/HEAD` to see where HEAD points.

Detached HEAD: Happens when HEAD points to a commit instead of a branch.

Reflog

git reflog

```
# Shows where HEAD and references have been, useful for recovering lost commits.
```

Example:

```
git reset --hard HEAD@{2}
```

Git Hooks

Git can run custom scripts on certain events.

Located in `.git/hooks`.

Examples:

- pre-commit: Runs before commit.
- post-merge: Runs after merge.

Sparse Checkout

Allows partial cloning of a repository.

```
git clone --filter=blob:none --no-checkout <repo>
cd <repo>
git sparse-checkout init --cone
git sparse-checkout set <directory>
```

Submodules

Use when including another Git repo inside a repo.

```
git submodule add <repo_url> <path>
git submodule update --init --recursive
```

Worktrees

Worktrees allow multiple working directories for a single repo.

Ultimate Git Commands Reference

```
git worktree add ../feature-dir feature-branch
```

Bisecting Bugs

```
git bisect start
git bisect bad
git bisect good <commit>
# Git checks out commits between and narrows down buggy commit.
```

Advanced Tagging

Create signed tags with GPG:

```
git tag -s v1.0 -m "Signed release"
git verify-tag v1.0
```

Custom Aliases

```
git config --global alias.co checkout
git config --global alias.st status
# Allows use of `git co` instead of `git checkout`
```