Git Reference

Set global configuration (only needs to be done once per machine):

```
git config --global user.name "Your Name"
git config --global user.email "you@some.domain"
git config --global color.ui "auto"
git config --global core.editor "your_editor"
```

Initialize the current working directory as a repository:

```
git init
```

Display the status of the repository:

```
git status
```

Add specific files to the staging area:

```
git add filename_1 filename_2
```

Add all modified files in the current directory and its sub-directories to the staging area:

```
git add -A .
```

Commit changes in the staging area to the repository's history: (Without -m and a message, this command runs a text editor.)

```
git commit -m "Some message"
```

View the history of the repository:

```
git log
```

Display differences between the current state of the repository and the last saved state:

```
git diff
```

Display differences between the current state of a particular file and the last saved state:

```
git diff path/to/file
```

Display differences between the last saved state and what's in the staging area:

```
git diff --staged
```

Compare files to the previously saved state:

```
git diff HEAD~1 path/to/file
```

Compare files to an earlier saved state:

```
git diff HEAD~27 path/to/file
```

Compare files to a specific earlier state:

```
git diff 123456 path/to/file
```

Erase changes since the last save:

```
git reset --hard HEAD
```

Restore file to its state in a previous revision:

```
git checkout 123456 path/to/file
```

Add a remote to a repository:

```
git remote add nickname remote_specification
```

Display a repository's remotes:

```
git remote -v
```

Push changes from a local repository to a remote (assuming `master` already exists in the remote):

```
git push nickname master
```

Push changes from a local repository to a remote (if `master` doesn't yet exist in the remote):

```
git push -u nickname master
```

Pull changes from a remote repoisitory:

```
git pull nickname master
```

Note: `master` may be replaced with another branch name by more advanced users.

Clone a remote repository:

```
git clone remote_specification
```

Markers used to show conflict in a file during a merge:

```
<<<<<<< HEAD
lines from local file
=======
lines from remote file
>>>>>>> 1234567890abcdef1234567890abcdef12345678
```