```
git commit -a -m "comment"
```

```
git commit -v
```

```
git diff --staged
```

```
git diff
```

List them in `.gitignore`. The syntax accepts glob patterns, leading & trailing slashes, and !.

Modified

Staged

```
git add [filename ...]
```

```
git status
```

```
git status
```

```
git status
```

```
git clone [url]
```

```
git add [filename ...]
```

```
git init
```

A repository with no working directory.

ssh.

```
git clone ssh://user@server:project.git
```

is the same as

```
git clone user@server:project.git
```

Clone a normal one, then:

```
git clone --bare my_project my_project.git
```

The OS's read/write permissions on the `.git` bare repo directory.

Go into the `.git` repo on the server and run:

```
git init --shared
```

Create a single "git" user and add all the devs' keys to `~/.ssh/authorized_keys`.

```
cd /opt/git
mkdir project.git
cd project.git
git --bare init
```

```
#on John's computer
cd myproject
git init
git add .
git commit -m 'initial commit'
git remote add origin git@gitserver:/opt/git/myproject.git
git push origin master
```

Change its shell to `git-shell` in `etc/passwd`.

```
git:x:1000:1000::/home/git:/bin/sh
                                :/usr/bin/git-shell
```

Create a webserver with the repo as webroot. Then add a post-update hook from the samples. People can then clone over http.

Use GitWeb, a CGI script that comes with Git.

Gitosis.

Gitolite.

```
git diff --check
```

On Windows clients:

```
git config --global core.autocrlf true
```

Converts LF to CRLF at checkout.

On Unix clients & on server:

```
git config --global core.autocrlf input
```

Converts CRLF to LF on commit.

On server:

```
git config --system receive.fsckObjects true
```