

# **Git & GitHub Interview Q&A; with Hands-on Commands**

**For Freshers & Developers**

## Part 1: Git & GitHub Interview Questions and Answers

**Q:** What is Git?

**A:** Git is a distributed version control system that tracks changes in source code and allows multiple developers to collaborate efficiently.

**Q:** What is GitHub?

**A:** GitHub is a cloud-based platform that hosts Git repositories and provides collaboration features like pull requests and issue tracking.

**Q:** Difference between Git and GitHub?

**A:** Git is a version control tool, while GitHub is a hosting service for Git repositories.

**Q:** What is a repository?

**A:** A repository is a storage location where project files and their history are maintained.

**Q:** What is commit?

**A:** A commit saves the current changes to the repository with a message.

**Q:** What is branching?

**A:** Branching allows parallel development without affecting the main code.

**Q:** What is merge?

**A:** Merge combines changes from one branch into another.

**Q:** What is pull request?

**A:** A pull request is a request to merge code changes after review.

**Q:** What is clone?

**A:** Clone creates a local copy of a remote repository.

**Q:** What is staging area?

**A:** The staging area holds changes before committing.

## Part 2: Hands-on Git Command Examples

Initialize a repository:

```
git init
```

Check repository status:

```
git status
```

Add files to staging:

```
git add .
```

Commit changes:

```
git commit -m 'Initial commit'
```

Create a new branch:

```
git branch feature-branch
```

Switch branch:

```
git checkout feature-branch
```

Merge branch:

```
git merge feature-branch
```

Add remote repository:

```
git remote add origin
```

Push changes:

```
git push origin main
```

Pull changes:

```
git pull origin main
```

View commit history:

```
git log
```

Resolve merge conflict:

Edit file → git add → git commit

## Part 3: Best Practices

- Write meaningful commit messages.
- Commit small and logical changes.
- Use branches for new features.
- Pull latest changes before pushing.
- Review code before merging.