1. What’s the difference Git & GitHub commands?

- **Git** → a **version control system** (runs locally)
- **GitHub** → a **remote hosting platform** for Git repositories (cloud + collaboration)

2. Basic Git Setup (One-Time)

```
git config --global user.name "Your Name"
git config --global user.email "you@example.com"
git config --list
```

3. Starting a Repository

Create a new repo locally

```
git init
```

Clone an existing repo (from GitHub)

```
git clone https://github.com/user/repo.git
```

4. Checking Repository Status

```
git status        # shows file state
git log           # commit history
git log --oneline # compact history
```

5. Tracking Changes

Add files

```
git add file.txt
git add .          # add everything
```

Commit changes

```
git commit -m "Meaningful commit message"
```

View differences

```
git diff           # unstaged changes
git diff --staged  # staged changes
```

---

6. Branching (Very Important)

Create & switch branches

```
git branch new-feature
git checkout new-feature
```

Or in one step:

```
git checkout -b new-feature
```

List branches

```
git branch
```

Merge branches

```
git checkout main
git merge new-feature
```

Delete branch

```
git branch -d new-feature
```

---

7. Remote Repositories (GitHub)

Add remote

```
git remote add origin https://github.com/user/repo.git
```

View remotes

```
git remote -v
```

Push code to GitHub

```
git push origin main
```

First push with upstream:

```
git push -u origin main
```

Pull updates

```
git pull origin main
```

---

8. Fetch vs Pull

```
git fetch    # downloads changes, doesn’t merge
git pull     # fetch + merge
```

---

9. Undoing Mistakes (Safely!)

Unstage a file

```
git reset file.txt
```

Undo last commit (keep changes)

```
git reset --soft HEAD~1
```

Discard local changes

```
git checkout -- file.txt
```

---

10. Stashing (Temporary Storage)

```
git stash
git stash list
git stash apply
git stash drop
```

---

11. Tagging Versions

```
git tag v1.0
git tag
git push origin v1.0
```

---

12. GitHub Collaboration Workflow

1. **Fork** repo

2. **Clone** fork
3. Create **feature branch**
4. Commit changes
5. **Push** branch
6. Open **Pull Request (PR)**

---

13. Common GitHub-Specific Actions

```
git pull upstream main   # sync fork with original repo
git push origin feature-branch
```

---

14. `.gitignore` (Very Important)

Example:

```
node_modules/
.env
*.log
```

---

15. Essential Git Commands Cheat Sheet

```
git init
git clone
git add
git commit
git push
git pull
git branch
git checkout
git merge
git stash
git reset
git log
```
