



## Checking & Tracking

See what's changed, staged, or untracked.

### ● CHECK STATUS

```
git status
```

### HISTORY GRAPH

```
git log --oneline --graph --decorate
```

### SHOW UNSTAGED CHANGES

```
git diff
```

### SHOW STAGED CHANGES

```
git diff --staged
```



## Adding & Committing

Stage files and save version history.

### ● STAGE EVERYTHING

```
git add .
```

### STAGE SPECIFIC FILE

```
git add <file>
```

### COMMIT WITH MESSAGE

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

### ADD & COMMIT (TRACKED)

```
git commit -am "message"
```





## Working with Remote

Sync your local code with GitHub.

- FETCH & MERGE

```
git pull origin main
```

PUSH TO GITHUB

```
git push origin main
```

CLONE REPOSITORY

```
git clone <url>
```



## Branching

Manage separate lines of development.

### ● LIST BRANCHES

```
git branch
```

### CREATE NEW BRANCH

```
git branch <name>
```

### SWITCH BRANCH

```
git checkout <name>
```

### CREATE & SWITCH

```
git checkout -b <name>
```

### MERGE BRANCH

```
git merge <name>
```



## Fixing & Undoing

Revert or stash changes safely.

### ● DISCARD CHANGES

```
git restore <file>
```

#### UNSTAGE FILE

```
git reset HEAD <file>
```

#### RESET (DESTRUCTIVE)

```
git reset --hard
```

#### SAVE TEMP CHANGES

```
git stash
```

#### APPLY TEMP CHANGES

```
git stash pop
```



## Condensed Daily Flow

Routine for saving and syncing work.

- 1. CHECK

```
git status
```

- 2. STAGE ALL

```
git add .
```

- 3. COMMIT

```
git commit -m "update"
```

- 4. PULL LATEST

```
git pull origin main
```

- 5. PUSH CHANGES

```
git push origin main
```