Git Command Cheatsheet

Stage All Changes and Commit with a Message

```
git add .
git commit -m "Add meaningful description of the changes made"
```

Move Commits to the Correct Branch

```
# Create and switch to the correct branch
git checkout -b correct-branch

# Go back to the wrong branch
git checkout wrong-branch

# Reset to remove the unwanted commits (N = number of commits)
git reset --hard HEAD~N

# Push reset changes (if needed)
git push --force
```

Create a New Branch and Push to GitHub

```
git checkout -b feature-branch-name

# Make changes...

git add .
git commit -m "Describe the changes"
git push origin feature-branch-name
```

Contribute to an Open Source Project on GitHub

```
1. Fork the repository on GitHub.
2. Clone your fork:
   git clone https://github.com/your-username/project-name.git
   cd project-name
3. Create a new branch:
   git checkout -b feature-description
4. Make changes and commit:
   git add .
   git commit -m "Your message"
   git push origin feature-description
5. Create a Pull Request on GitHub.
```

Resolve Merge Conflicts

```
git fetch origin
git merge origin/main

# Resolve conflict markers in files, then:
git add .
```

Git Command Cheatsheet

```
git commit -m "Resolved merge conflicts"
```

Create a Branch from the Latest Main

```
git checkout main
git pull origin main
git checkout -b feature-branch-name
```

Revert to a Specific Commit

```
# Option 1: Discard all commits after a specific commit
git reset --hard <commit-hash>
git push origin HEAD --force

# Option 2: Revert changes without deleting history
git revert <commit-hash>..HEAD
```

Restore a Deleted File After Commit

```
# From the previous commit
git checkout HEAD~1 -- path/to/deleted_file
git add path/to/deleted_file
git commit -m "Restore accidentally deleted file"

# From a specific commit
git checkout <commit-hash> -- path/to/deleted_file
```