

# *skillcrush*

# TROUBLESHOOTING GIT

## *cheatsheet*

This cheatsheet provides commands to solve common troubleshooting situations with Git including locating changes, undoing commits, and untracking files and directories.

---

## COMPARISON COMMANDS

---

Comparison commands allow you to spot differences between commits and branches. You'll use these commands to locate and fix undesired commits and changes.

### **git diff**

Shows changes between your working directory and the last commit.

```
git diff
```

---

### **git diff --staged**

Shows changes between your last commit and the files that you staged to be committed.

```
git diff --staged
```

---

### **git diff <branch\_name>..<branch\_name>**

View differences between two branches.

```
git diff main..contact-info
```

---

### **git diff <commit hash>..<commit hash>**

View differences between two commits.

```
git diff c9de0a..779785
```

---

## UNDO COMMANDS

---

Undo commits, change commit messages, and stop a merge using these undo commands.

### **git reset --hard <commit hash>**

Permanently deletes the commits and any uncommitted changes on the branch. Use with caution!

```
git reset --hard 25fg94
```

---

### **git reset --soft <commit hash>**

Rewinds commits to the staging area. You can choose to modify or commit the changes.

```
git reset --soft 689vft
```

---

### **git reset --mixed <commit hash>**

Rewinds commits to before staging. You can choose to modify, add, or save the commit.

```
git reset --mixed v56et2
```

---

### **git revert <commit hash>**

Reverse the changes made to the commit without changing the project history.

```
git revert t78y90
```

---

### **git commit --amend -m**

Change the most recent commit message to a new message.

```
git commit --amend -m "Updated email and phone number"
```

---

### **git checkout <file\_name>**

Rewind a file to match the file version of the last commit.

```
git checkout about.html
```

---

## **git merge --abort**

Stop a merge that's in progress.

```
git merge --abort
```

---

## **UNTRACK COMMANDS**

---

When a file is **untracked**, Git will stop tracking any updates to the file. You'll untrack a file or directory that you no longer want to commit to your repo.

## **git rm --cached <file\_name>**

Untrack a file.

```
git rm --cached experiments.html
```

---

## **git rm -cached <directory\_name>**

Untrack a directory.

```
git rm --cached client_test_site
```