

# Git Stash Cheat Sheet

## Common Use Cases:

- You're working on something but need to switch branches quickly.
- You want to pull the latest changes but have local modifications that aren't ready.

## Basic Commands

### 1. Stash your changes

`git stash`

- Saves your *\*tracked\** modified files and reverts them to the last commit state.
- Untracked files (new files) won't be stashed by default.

### 2. Include untracked files (optional)

`git stash -u`

- Includes untracked (git status -> red files).

### 3. View the stash list

`git stash list`

- Shows a list of all stashed changes like:  
stash@{0}: WIP on main: abc123 Commit message

### 4. Apply the last stash

`git stash apply`

- Applies the latest stash to your working directory without removing it from the stash list.

### 5. Apply and remove the stash

`git stash pop`

- Applies the latest stash and deletes it from the stash list.

# Git Stash Cheat Sheet

## 6. Apply a specific stash

```
git stash apply stash@{1}
```

## 7. Drop (delete) a specific stash

```
git stash drop stash@{0}
```

## 8. Clear all stashes

```
git stash clear
```

## Example Workflow

```
# make some changes in files
```

```
git stash          # save changes
```

```
git switch other-branch
```

```
# do work on other-branch
```

```
git switch main
```

```
git stash pop      # bring back saved changes
```