Git Stashing Explained

In Git, stashing means temporarily saving your uncommitted changes without committing them. This allows you to switch branches or work on something else without losing progress.

- Save changes to stash

```
git stash
```

Saves uncommitted changes and reverts the working directory to the last commit.

- List all stashes

```
git stash list
```

Shows a list of saved stashes.

- Apply the last stash

```
git stash pop
```

Restores the last stashed changes and removes them from the stash list.

- Apply a specific stash

```
git stash apply stash@{n}
```

Restores a specific stash without removing it.

- Drop a specific stash

```
git stash drop stash@{n}
```

Deletes a specific stash.

- Clear all stashes

```
git stash clear
```

Removes all stashed changes permanently.

Example Use Case

- You're working on Feature A, but suddenly need to switch to Feature B.
- Instead of committing unfinished work, you run `git stash` to save your changes.
- You switch to Feature B, complete the work, and return to Feature A.
- Run `git stash pop` to bring back your saved changes and continue working.