1. git restore:

`git restore` is a command used to restore working directory files to their state from a specific commit or branch. It's useful for discarding changes or undoing modifications.

Syntax:

```
git restore [<options>] <file>...
```

Common use cases:

- **Restore a file to its last committed state**:

  ```
  git restore <file>
  ```

  This will discard any changes made to `<file>` and restore it to the version from the last commit.

- **Restore all files to their last committed state**:

  ```
  git restore .
  ```

  This restores all modified files in the working directory to the state of the last commit.

- **Restore a file from a specific commit**:

  ```
  git restore --source=<commit> <file>
  ```

  This restores a specific file from the specified commit.

- **Restore files with untracked changes** (use cautiously):

  ```
  bash
  Copy code
  git restore --staged <file>
  ```

  This will unstage the file without discarding the changes made to it in the working directory.

---

2. Pull Request (PR):

A **Pull Request (PR)** is a feature in GitHub and other Git-based platforms that allows developers to propose changes to a repository. It's used for code review and collaboration. When a developer wants to merge their changes from one branch to another (typically from a feature branch to `main` or `master`), they create a pull request.

Steps to create a pull request:

1. **Push your branch to the remote repository**:

```
git push origin <branch-name>
```

This uploads your local branch to the remote repository.

2. **Create a Pull Request**:
   - On GitHub, go to the repository page and click on the "Pull requests" tab.
   - Click "New Pull Request."
   - Select the branch you want to merge into (`main`, `develop`, etc.) and the branch you want to merge from.
   - Add a title and description for your pull request.
   - Submit the pull request for review.
3. **Review and Merge**:
   - Other contributors review the code.
   - After approval, the PR is merged into the target branch by clicking the "Merge" button.

---

3. .gitignore:

A `.gitignore` file is used to tell Git which files or directories to ignore when committing changes to the repository. This is typically used for files that are not relevant to version control (e.g., temporary files, build artifacts, dependency files).

How to create and use `.gitignore`:

1. **Create a `.gitignore` file** in the root of your repository.
2. **Add patterns of files and directories to ignore**. Examples:

   ```
   # Ignore all .log files
   *.log

   # Ignore node_modules directory
   node_modules/

   # Ignore all files in the temp folder
   temp/
   ```

3. **Ensure the .gitignore file is committed**: If you've already tracked files that you want to ignore, you need to remove them from Git's tracking:

   ```
   git rm --cached <file>
   ```

4. **Check that the files are being ignored** by running:

   ```
   git status
   ```

Summary:

- `git restore`: Used to restore files or undo changes.

- **Pull request**: A feature in Git hosting platforms for proposing and reviewing code changes.
- `.gitignore`: A file for specifying which files or directories Git should ignore in version control.