

Git :-

Configuring Git

```
git config --global user.name "My Name"
```

```
git config --global user.email "someone@email.com"
```

```
git config --list
```

untracked

new files that git doesn't yet track

modified

changed

staged

file is ready to be committed

unmodified

unchanged

```
git add fileName.cpp
```

```
git add dir
```

```
git add .
```

Add & Commit

add - adds new or changed files in your working directory to the Git staging area.

```
git add <- file name ->
```

commit - it is the record of change

```
git commit -m "some message"
```

Cloning Private Repository from Github Using Personal Token

1. Click on User Icon
2. Go to **"Settings"**:
3. Scroll down then go to **"Developer settings"**:
4. Press **"Personal access tokens"** then press **"Generate new token"**:
5. Fill **"Note"** then check **"repo"**:
6. Scroll down then press **"Generate token"**:
7. Finally, a **PAT** is generated:
8. `git clone https://<Personal Token>@github.com.....`

Merging Code



Way 1

`git diff <- branch name->` (to compare commits, branches, files & more)

`git merge <- branch name->` (to merge 2 branches)

Way 2

Create a PR

Pull Request

It lets you tell others about changes you've pushed to a branch in a repository on GitHub.

Pull Command

`git pull origin main`

used to fetch and download content from a remote repo and immediately update the local repo to match that content.

Resolving Merge Conflicts

An event that takes place when Git is unable to automatically resolve differences in code between two commits.

<p>This is a new repo.</p>

Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes

<<<<<< HEAD (Current Change)

<p>This is a new feature (dropdown)</p>

=====

<p>This is a new feature (button)</p>

>>>>>> main (Incoming Change)



Resolve in Merge Editor

Undoing Changes

Case 1 : staged changes

```
git reset <- file name ->
```

```
git reset
```

Case 2 : committed changes (for one commit)

```
git reset HEAD~1
```

Case 3 : committed changes (for many commits)

```
git reset <- commit hash ->
```

```
git reset --hard <- commit hash ->
```

Fork

A fork is a new repository that shares code and visibility settings with the original “upstream” repository.

Fork is a rough copy.