

# Git Basics

- **Version control** is a system that records changes to a file or set of files over time so that you can recall specific versions later.
- A **Repository** contains all of your project's files and each file's revision history.

## 3 States

### Modified / Un-tracked

- changes are made in a file but have not been committed.

### Staged

- a modified file is marked and is ready to get committed

### Committed

- a snapshot of the data is permanently stored in git directory.

## Creating new Git Repository

- `git init` initialises an empty git repository in the current directory. A “.git” file is created and hidden.

```
driptanil@driptanil in ~/Documents as 0 took 3s
└ git init learningGit
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/driptanil/Documents/learningGit/.git/
```

- `ls -a` shows all the files including the hidden files.

```
driptanil@driptanil in repo: learningGit on 0 master
└ ls -a
drwxr-xr-x - driptanil 12 Mar 16:13 .
drwxr-xr-x - driptanil 12 Mar 16:13 ..
drwxr-xr-x - driptanil 12 Mar 16:13 .git
```

- `touch hello.txt` will create a file named “hello.txt”.

```
driptanil@driptanil in repo: learningGit on 0 master took 30ms
└ touch hello.txt
```

## Git Status

- `git status` shows the status of the git repository.

```
driptanil@driptanil in repo: learningGit on ⚡ master [?] took 3ms
└ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    hello.txt

nothing added to commit but untracked files present (use "git add" to track)
```

## Git Add

- `git add <file>` stages the changes of the file.

```
driptanil@driptanil in repo: learningGit on ⚡ master [?] took 2ms
└ git add hello.txt

driptanil@driptanil in repo: learningGit on ⚡ master [+] took 28ms
└ git status
On branch master    I

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   hello.txt
```

## Git Commit

- `git commit -m "<commit_message>"` commits to all the staged changes. "-m" adds name to the commit.

```
driptanil@driptanil in repo: learningGit on ⚡ master [+] took 6ms
└ git commit -m "hello.txt file added"
[master (root-commit) 7bc219b] hello.txt file added
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 hello.txt
```

## Git Log

- `git log` will show all the history of all commits.

```
driptanil@driptanil in repo: learningGit on ⚡ master [!] took 4ms
└ git log
commit 7bc219b3a62a0a2d1ff326af6088aac593abd778 (HEAD → master)
Author: driptanil <driprecovery@gmail.com>
Date:   Sat Mar 12 16:34:36 2022 +0530

  hello.txt file added
```

## Git Restore

- `vim names.txt` allows editing the file in the console, when files are modified after they are needed to be staged again.

```
driptanil@driptanil in repo: learningGit on ⚡ master took 32ms
└ vim hello.txt

driptanil@driptanil in repo: learningGit on ⚡ master [!] took 18s
└ git status
On branch master
Changes not staged for commit:           []
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   hello.txt

no changes added to commit (use "git add" and/or "git commit -a")

driptanil@driptanil in repo: learningGit on ⚡ master [!] took 6ms
└ cat hello.txt
File: hello.txt
Hello, Git
```

- `git restore --staged <staged_file>` will remove staged changes for file from staged (only works after first commit).

```
driptanil@driptanil in repo: learningGit on ⚡ master [!] took 2ms
└ git add hello.txt

driptanil@driptanil in repo: learningGit on ⚡ master [+] took 6ms
└ git status hello.txt
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   hello.txt

driptanil@driptanil in repo: learningGit on ⚡ master [+] took 7ms
└ git restore --staged hello.txt

driptanil@driptanil in repo: learningGit on ⚡ master [!] took 7ms
└ git status hello.txt
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   hello.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

## Git Reset

- `git reset <commit_id>` will revert to the commit.

```

driptanil@driptanil in repo: learningGit on ⚡ master [!] took 9ms
└ git add hello.txt

driptanil@driptanil in repo: learningGit on ⚡ master [+] took 7ms
└ git commit -m "hello.txt modified"
[master 166201e] hello.txt modified
 1 file changed, 1 insertion(+)

driptanil@driptanil in repo: learningGit on ⚡ master took 5ms
└ git log
commit 166201e87cc9b688e159afb3ee71d3dc9a98d1ee (HEAD → master)
Author: driptanil <driprecovery@gmail.com>
Date:   Sat Mar 12 16:51:41 2022 +0530

  hello.txt modified

commit 7bc219b3a62a0a2d1ff326af6088aac593abd778
Author: driptanil <driprecovery@gmail.com>
Date:   Sat Mar 12 16:34:36 2022 +0530

  hello.txt file added

driptanil@driptanil in repo: learningGit on ⚡ master took 4ms
└ git reset 7bc219b3a62a0a2d1ff326af6088aac593abd778
Unstaged changes after reset:
M      hello.txt

driptanil@driptanil in repo: learningGit on ⚡ master [!] took 2ms
└ git log
commit 7bc219b3a62a0a2d1ff326af6088aac593abd778 (HEAD → master)
Author: driptanil <driprecovery@gmail.com>
Date:   Sat Mar 12 16:34:36 2022 +0530

  hello.txt file added

driptanil@driptanil in repo: learningGit on ⚡ master [!] took 4ms
└ git status hello.txt
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   hello.txt

no changes added to commit (use "git add" and/or "git commit -a")

```

## Git Stash

- `git stash` will neither commit the changes nor will delete all the changes, it just stores the changes somewhere else.

```
driptanil@driptanil in repo: learningGit on ⚡ master [!] took 29s
└ cat hello.txt
File: hello.txt
Hello, Git

driptanil@driptanil in repo: learningGit on ⚡ master [!] took 80ms
└ git status hello.txt
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   hello.txt

no changes added to commit (use "git add" and/or "git commit -a")

driptanil@driptanil in repo: learningGit on ⚡ master [!] took 2ms
└ git add hello.txt

driptanil@driptanil in repo: learningGit on ⚡ master [+] took 2ms
└ git stash
Saved working directory and index state WIP on master: 7bc219b hello.txt file added

driptanil@driptanil in repo: learningGit on ⚡ master [$] took 8ms
└ git log
commit 7bc219b3a62a0a2d1ff326af6088aac593abd778 (HEAD → master)
Author: driptanil <driprecover@gmail.com>
Date:   Sat Mar 12 16:34:36 2022 +0530

  hello.txt file added

driptanil@driptanil in repo: learningGit on ⚡ master [$] took 9ms
└ git status hello.txt
On branch master
nothing to commit, working tree clean

driptanil@driptanil in repo: learningGit on ⚡ master [$] took 2ms
└ cat hello.txt
File: hello.txt  <EMPTY>
```

- `git stash pop` will bring back all the changes stored somewhere else and will be staged.

```
driptanil@driptanil in repo: learningGit on ⚡ master [!] took 39s
λ cat hello.txt
File: hello.txt
Hello, Git & GitHub

driptanil@driptanil in repo: learningGit on ⚡ master [!] took 73ms
λ git status hello.txt
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   hello.txt

no changes added to commit (use "git add" and/or "git commit -a")

driptanil@driptanil in repo: learningGit on ⚡ master [!] took 2ms
λ git add hello.txt

driptanil@driptanil in repo: learningGit on ⚡ master [+] took 2ms
λ git stash
Saved working directory and index state WIP on master: 7bc219b hello.txt file added

driptanil@driptanil in repo: learningGit on ⚡ master [$] took 21ms
λ cat hello.txt
File: hello.txt  <EMPTY>

driptanil@driptanil in repo: learningGit on ⚡ master [$] took 70ms
λ git stash pop
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   hello.txt

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (ff7a5b08c55bd80d1ab3d822302e443721697614)

driptanil@driptanil in repo: learningGit on ⚡ master [!] took 18ms
λ cat hello.txt
File: hello.txt
Hello, Git & GitHub
```

- `git stash clear` will remove all the changes that haven't been staged or committed will be removed.

```

driptanil@driptanil in repo: learningGit on ⚡ master [!] took 18ms
└ cat hello.txt      ┌
File: hello.txt
Hello, Git & GitHub

driptanil@driptanil in repo: learningGit on ⚡ master [!] took 70ms
└ git status hello.txt
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   hello.txt

no changes added to commit (use "git add" and/or "git commit -a")

driptanil@driptanil in repo: learningGit on ⚡ master [!] took 9ms
└ git add hello.txt

driptanil@driptanil in repo: learningGit on ⚡ master [$+] took 2ms
└ git stash
Saved working directory and index state WIP on master: 7bc219b hello.txt file added

driptanil@driptanil in repo: learningGit on ⚡ master [$] took 20ms
└ cat hello.txt
File: hello.txt  <EMPTY>

driptanil@driptanil in repo: learningGit on ⚡ master [$] took 74ms
└ git stash clear

driptanil@driptanil in repo: learningGit on ⚡ master took 5ms
└ cat hello.txt
File: hello.txt  <EMPTY>

driptanil@driptanil in repo: learningGit on ⚡ master took 70ms
└ git status hello.txt
On branch master
nothing to commit, working tree clean

```

## Adding Origin

- `git remote add origin https://github.com/<username>/<repository_link>` will link the working directory to the GitHub repository.

```

└ driptanil@driptanil in repo: LearningGit on ⚡ master [+] took 3ms
└ git remote add origin git@github.com:driptanil/LearningGit

```

## Git Push

- `git push <repository-link>` or `git push origin` will push all the commits made in local git directory to remote repository. **Does not work, now token are used for authentication of user.** [GitHub Personal Access Token Documentation](#)

```

driptanil@driptanil in repo: learningGit on ⚡ master took 1s
└ git push https://github.com/driptanil/learningGit
Username for 'https://github.com': lol
Password for 'https://lol@github.com':
remote: Support for password authentication was removed on August 13, 2021. Please use a personal access token instead.
remote: Please see https://github.blog/2020-12-15-token-authentication-requirements-for-git-operations/ for more information.
fatal: Authentication failed for 'https://github.com/driptanil/learningGit/'

```

## GitHub Personal Access Token:

The screenshot shows the GitHub developer settings page. In the sidebar, 'Personal access tokens' is selected. The main area displays a table of generated tokens. One token is highlighted for the 'Data-Structures-Algorithm Repository':  
Last used within the last week  
Delete  
admin:enterprise, admin:gpg\_key, admin:org, admin:org\_hook,  
admin:public\_key, admin:repo\_hook, delete:packages, gist, notifications, repo,  
user, workflow, write:discussion, write:packages  
Expires on Sun, Apr 10 2022.

- Click on **Generate new token**
- Choose the permissions and validity of the token
- Copy the token (Store it somewhere safe)
- `git push https://<token>@github.com/<username>/<repository_link>` will push all the commits to the repository.

```
driptanil@driptanil in repo: learningGit on ⚡ master took 1s
└ git push "https://<token>@github.com/driptanil/LearningGit"
```

```
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 218 bytes | 218.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/driptanil/LearningGit
 * [new branch]      master → master
```

## SSH Keys:

1. Use `git clone git@github.com:<user>/<repository>.git` to clone repository.

The screenshot shows the GitHub repository page for 'driptanil/LearningGit'. The 'Code' tab is selected. On the right, there's a 'Clone' section with options for HTTPS, SSH, and GitHub CLI. The SSH URL is shown as `git@github.com:driptanil/LearningGit.git`.

2. Use `ssh-keygen -o` to generate SSH key using git.

```

driptanil@driptanil in ~
└─ ssh-keygen -o
Generating public/private rsa key pair.
Enter file in which to save the key (/home/driptanil/.ssh/id_rsa):
/home/driptanil/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/driptanil/.ssh/id_rsa
Your public key has been saved in /home/driptanil/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:N0wgrrdY3SbGnYbiDWuv9xlmTficCN5z1kaqAXrOsAQ driptanil@driptanil-80tl
The key's randomart image is:
+---[RSA 3072]----+
| . .
| . .
| . .
| . o * o
| E = S @ .
| * X O B =
| . O + O O o
| o B.o O .
| oo+.+
+---[SHA256]----+

```

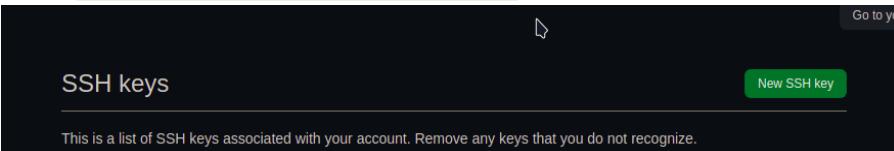
3. Use `cat ~/.ssh/id_rsa.pub` (this file contains the private SSH key) and select the contents and copy using `Ctrl + Shift + C`.

```

driptanil@driptanil in ~ took 1s
└─ cat ~/.ssh/id_rsa.pub
File: /home/driptanil/.ssh/id_rsa.pub

```

4. Open <https://github.com/settings/keys> (adding the private SSH key to GitHub), click on `new SSH keys`.



Paste the copied key in `key` section and add a `title`.

SSH keys / Add new

Title

Key

Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'

Add SSH key

5. Click on `Add SSH key`.

## Pulling Changes

- `git pull origin` will pull all the commits made in the remote repository to the local git directory.

## Cloning Repository

- `git clone <repository_link>` will download all the files in the remote repository to a local git directory.

# How to make contributions to the existing GitHub repository?

# Forking a Repository

- No one except the owner of the GitHub repository is allowed to make changes directly to the repository.
- **Fork** allows us to make a copy of the existing GitHub repository with our own ownership and in this repository, we are allowed to make changes directly.
- The original repository which has been forked is known as the Upstream URL.