

Markdown documentation code of **push assembly line on github**

Git and GitHub are related but distinct tools used in the field of version control for software development. Here's a breakdown of their differences:

Git

- Git is a distributed version control system (DVCS) that allows developers to track changes in their source code over time.
- It was created by Linus Torvalds in 2005 and is open-source.
- Git operates locally on a developer's machine, providing a command-line interface and a set of commands to manage repositories.
- It allows developers to create branches, make commits, merge changes, and revert to previous versions easily.

- Git does not require a network connection to operate; it functions offline and provides fast performance.

GitHub

- GitHub is a web-based hosting service for Git repositories.
- It was created in 2008 and is owned by Microsoft.
- GitHub provides a platform for collaboration, sharing, and hosting Git repositories on the web.
- It offers additional features such as issue tracking, pull requests, code reviews, project management tools, and more.
- GitHub enables multiple developers to work together on the same project, managing their codebase and merging changes seamlessly.

- It also provides features for forking repositories, contributing to open-source projects, and hosting documentation.

In summary, Git is the version control system itself, whereas GitHub is a hosting platform built around Git, adding web-based collaboration and additional features. Git can be used without GitHub, but GitHub relies on Git as its underlying version control system.



Git

Software

Version control

Maintained by Linux

Open-Source

No user management

Locally installed

Minimal external tool
configuration

Little to no competition



GitHub

Service

Git repository hosting

Maintained by Microsoft

Free or paid membership

Built-in user management

Hosted on the web

Active marketplace for
tool integration

High competition

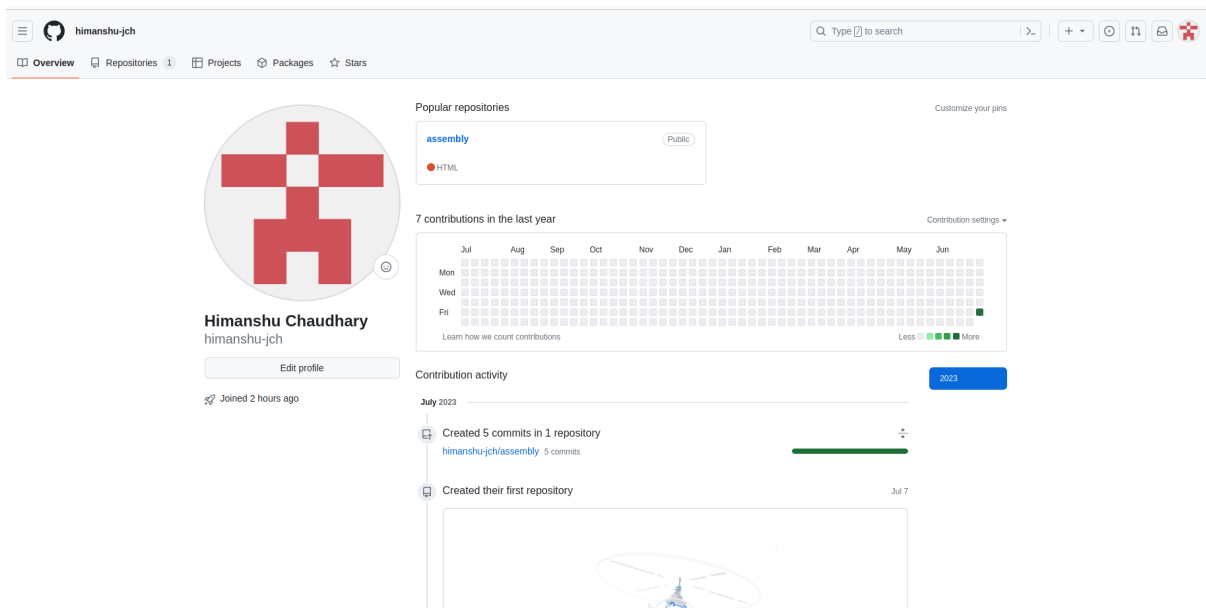
To push an assembly line code to GitHub, follow these step-by-step instructions

Step 1. First, install GitHub and create an account on GitHub.

- `sudo apt install git`
- `git --version`

```
himanshu@123:~$ sudo apt install git
[sudo] password for himanshu:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  app-install-data-partner bsdmainutils crda g++-9 gcc-10-base
  gir1.2-clutter-1.0 gir1.2-clutter-gst-3.0 gir1.2-cogl-1.0
  gir1.2-coglpango-1.0 gir1.2-gnomebluetooth-1.0 gir1.2-gtkclutter-1.0
  gnome-getting-started-docs gnome-screenshot grim ippusbxd libamtk-5-0
  libamtk-5-common libasn1-8-heimdal libboost-date-time1.71.0
  libboost-filesystem1.71.0 libboost-iostreams1.71.0 libboost-locale1.71.0
  libboost-thread1.71.0 libbrotli1.0.7 libcamel-1.2-62 libcbor0.6 libcdio18
  libcmis-0.5-5v5 libcroco3 libdns-export1109 libdataserver-1.2-24
  libdataserverui-1.2-2 libextutils-pkgconfig-perl libfuse2
  libgdk-pixbuf-xlib-2.0-0 libgdk-pixbuf2.0-0 libgsoap-2.8.91
  libgssapi3-heimdal libgupnp-1.2-0 libhandy-0.0-0 libhcrypto4-heimdal
  libheimbase1-heimdal libheimntlm0-heimdal libhogweed5 libhx509-5-heimdal
  libicu66 libidn11 libisl22 libjson-c4 libjuh-java libjurt-java
  libkrb5-26-heimdal libldap-2.4-2 liblibreoffice-java libllvm12 libmozjs-68-0
  libmpdec2 libneon27-gnutls libnettle7 libntfs-3g883 liborcus-0.15-0
  libperl5.30 libphonenumbers7 libpoppler97 libprotobuf17 libpython3.8
  libpython3.8-minimal libpython3.8-stdlib libqpdf26 libraw19
  libreoffice-style-tango libridl-java libroken18-heimdal libsane libsnmp35
  libssl1.1 libstdc++-9-dev libtepl-4-0 libtracker-control-2.0-0
  libtracker-miner-2.0-0 libtracker-sparsql-2.0-0 libunoloader-java libvpx6
```

```
himanshu@123:~$ git --version
git version 2.34.1
```



Step 2. **Create a new repository:**

A repository is a place where you can store and manage your code. To create a new repository, click on the "New" button on the left side of your GitHub dashboard. Provide a name for your repository, a brief description, and choose whether you want it to be public (visible to everyone) or private (accessible only to you or collaborators). You can also initialise the repository with a README file, which is a good practice to provide some basic information about your project.

Step 3. **Clone the repository:**

Cloning a repository means creating a local copy on your computer. To clone your repository, click on the green "Code" button on your repository page and copy the repository's URL. Open a terminal or Git Bash on your computer, navigate to the directory where you want to clone the repository, and use the following command:

- `git clone <repository-url>`

```
himanshu@123:~$ ls
assembly  Documents  Pictures      ricci.txt  Videos
desktop   Downloads  preview.plantuml  snap      'VirtualBox VMs'
Desktop   Music      Public        Templates

himanshu@123:~$ cd assembly/
himanshu@123:~/assembly$ ls
a.png      Assembly.pdf  chain.png     efficiency.png  productivity.png
Assembly.html  Assembly.png  cost.png      faster.png      quality.png
Assembly.jpeg  autonomy.png  dependency.png  limited.png     time.png
Assembly.md    car.png       divide.png     monotonous.png  work.png
```

```
himanshu@123:~/assembly$ git init
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/himanshu/assembly/.git/
```

```
himanshu@123:~/assembly$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Assembly.html
    Assembly.jpeg
    Assembly.md
    Assembly.pdf
    Assembly.png
    a.png
    autonomy.png
    car.png
    chain.png
    cost.png
    dependency.png
    divide.png
    efficiency.png
    faster.png
    limited.png
    monotonous.png
    productivity.png
    quality.png
```

Step 4. Add file and check status:

- git add
- git status

```
himanshu@123:~/assembly$ git add .
himanshu@123:~/assembly$
himanshu@123:~/assembly$
himanshu@123:~/assembly$
himanshu@123:~/assembly$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   Assembly.html
    new file:   Assembly.jpeg
    new file:   Assembly.md
    new file:   Assembly.pdf
    new file:   Assembly.png
    new file:   a.png
    new file:   autonomy.png
    new file:   car.png
    new file:   chain.png
    new file:   cost.png
    new file:   dependency.png
    new file:   divide.png
    new file:   efficiency.png
    new file:   faster.png
    new file:   limited.png
    new file:   monotonous.png
    new file:   productivity.png
```

Step 5. **Set up Git:**

Before you can start committing code to your repository, you need to set up Git on your local

machine. Download and install Git from the official website (<https://git-scm.com>). Once installed, open a terminal or Git Bash and configure your Git username and email using the following commands:

- `git commit -m "first commit"`
- `git config --global user.name "Your Name"`
- `git config --global user.email "your-email@example.com"`

```
himanshu@123:~/assembly$ git commit -m "first commit"
Author identity unknown

*** Please tell me who you are.

Run

    git config --global user.email "you@example.com"
    git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'himanshu@123.(none)')
himanshu@123:~/assembly$
```

```
himanshu@123:~/assembly$ git config --global user.email "jaathimanshu847@gmail.com"
himanshu@123:~/assembly$
himanshu@123:~/assembly$
himanshu@123:~/assembly$
himanshu@123:~/assembly$ git config --global user.name "himanshu-jch"
himanshu@123:~/assembly$
himanshu@123:~/assembly$
himanshu@123:~/assembly$ git config user.name
himanshu-jch
himanshu@123:~/assembly$
himanshu@123:~/assembly$
himanshu@123:~/assembly$
himanshu@123:~/assembly$ git config user.email
jaathimanshu847@gmail.com
```

```
himanshu@123:~/assembly$ git commit -m "first commit"
[master (root-commit) 628c1ae] first commit
20 files changed, 554 insertions(+)
create mode 100644 Assembly.html
create mode 100644 Assembly.jpeg
create mode 100644 Assembly.md
create mode 100644 Assembly.pdf
create mode 100644 Assembly.png
create mode 100644 a.png
create mode 100644 autonomy.png
create mode 100644 car.png
create mode 100644 chain.png
create mode 100644 cost.png
create mode 100644 dependency.png
create mode 100644 divide.png
create mode 100644 efficiency.png
create mode 100644 faster.png
create mode 100644 limited.png
create mode 100644 monotonous.png
create mode 100644 productivity.png
create mode 100644 quality.png
create mode 100644 time.png
create mode 100644 work.png
```

Step 6. Add the git remote origin and push branch.

```
himanshu@123:~/assembly$ git remote add origin https://github.com/himanshu-jch/assembly.git
himanshu@123:~/assembly$
himanshu@123:~/assembly$
himanshu@123:~/assembly$ git push -u origin master
Username for 'https://github.com': himanshu-jch
Password for 'https://himanshu-jch@github.com':
remote: Support for password authentication was removed on August 13, 2021.
remote: Please see https://docs.github.com/en/get-started/getting-started-with-git/about-remote-repositories#cloning-with-https-urls for information on currently recommended modes of authentication.
fatal: Authentication failed for 'https://github.com/himanshu-jch/assembly.git/'
himanshu@123:~/assembly$ git push -u origin master
Username for 'https://github.com': himanshu-jch
Password for 'https://himanshu-jch@github.com':
remote: Support for password authentication was removed on August 13, 2021.
remote: Please see https://docs.github.com/en/get-started/getting-started-with-git/about-remote-repositories#cloning-with-https-urls for information on currently recommended modes of authentication.
fatal: Authentication failed for 'https://github.com/himanshu-jch/assembly.git/'
```

- If you are unable to push to GitHub while following this step, please follow the instructions below.

Repositories

Codespaces

Packages

Copilot

Pages

Saved replies

Security

Code security and analysis

Integrations

Applications

Scheduled reminders

Archives

Security log

Sponsorship log

Developer settings

Don't specify

URL

Social accounts

Link to social profile

Link to social profile

Link to social profile

Link to social profile

Company

You can @mention your company's GitHub organization to link it.

Location

Display current local time

Other users will see the time difference from their local time.

All of the fields on this page are optional and can be deleted at any time, and by filling them out, you're giving us consent to share this data wherever your user profile appears. Please see our [privacy statement](#) to learn more about how we use this information.

Update profile

Contributions & Activity

Make profile private and hide activity

Enabling this will hide your contributions and activity from your GitHub profile and from social features like followers, stars, feeds, leaderboards and releases.

Include private contributions on my profile

Your contribution graph, achievements, and activity overview will show your private contributions without revealing any repository or organization information. [Read more.](#)

Update preferences

Settings / Developer Settings

Q Type to search

GitHub Apps

OAuth Apps

Personal access tokens

GitHub Apps

New GitHub App

Want to build something that integrates with and extends GitHub? [Register a new GitHub App](#) to get started developing on the GitHub API. You can also read more about building GitHub Apps in our [developer documentation](#).

© 2023 GitHub, Inc.

[Terms](#)

[Privacy](#)

[Security](#)

[Status](#)

[Docs](#)

[Contact GitHub](#)

[Pricing](#)

[API](#)

[Training](#)

[Blog](#)

[About](#)

Settings / Developer Settings

Q Type to search

GitHub Apps

OAuth Apps

Personal access tokens

Fine-grained tokens

Tokens (classic)

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API](#) over Basic Authentication.

Note

What's this token for?

Expiration *

<input type="checkbox"/> read:enterprise	Read enterprise profile data
<input type="checkbox"/> audit_log	Full control of audit log
<input type="checkbox"/> read:audit_log	Read access of audit log
<input type="checkbox"/> codespace	Full control of codespaces
<input type="checkbox"/> codespace:secrets	Ability to create, read, update, and delete codespace secrets
<input type="checkbox"/> project	Full control of projects
<input type="checkbox"/> read:project	Read access of projects
<input type="checkbox"/> admin:gpg_key	Full control of public user GPG keys
<input type="checkbox"/> write:gpg_key	Write public user GPG keys
<input type="checkbox"/> read:gpg_key	Read public user GPG keys
<input type="checkbox"/> admin:ssh_signing_key	Full control of public user SSH signing keys
<input type="checkbox"/> write:ssh_signing_key	Write public user SSH signing keys
<input type="checkbox"/> read:ssh_signing_key	Read public user SSH signing keys

Personal access tokens (classic)

Generate new token ▼

Revoke all

Tokens you have generated that can be used to access the [GitHub API](#).

github — *admin:enterprise, admin:gpg_key, admin:org, admin:org_hook, admin:public_key, admin:repo_hook, admin:ssh_signing_key, audit_log, codespace, delete:packages, delete_repo, gist, notifications, project, repo, user, workflow, write:discussion, write:packages*

⚠ This token has no expiration date.

Last used within the last week

Delete

Step 7. Push changes to GitHub:

After committing your changes, you can push them to your GitHub repository. Use the following command:

- `git push -u origin main`

```

himanshu@123:~/assembly$
himanshu@123:~/assembly$ git add README.md
himanshu@123:~/assembly$
himanshu@123:~/assembly$ git commit -m "first commit"
[main e0daf0e] first commit
2 files changed, 137 insertions(+)
create mode 100644 README.md
create mode 100644 assembly

```

```

himanshu@123:~/assembly$ git push -u origin main
Username for 'https://github.com': himanshu-jch
Password for 'https://himanshu-jch@github.com':
Enumerating objects: 33, done.
Counting objects: 100% (33/33), done.
Delta compression using up to 4 threads
Compressing objects: 100% (31/31), done.
Writing objects: 100% (33/33), 6.51 MiB | 1.79 MiB/s, done.
Total 33 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), done.
To https://github.com/himanshu-jch/assembly.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.

```

- And check your profile repositories .

The screenshot shows the GitHub interface for a repository named 'assembly' owned by 'himanshu-jch'. The repository is public and has 5 commits. The file list shows various files including README.md, Assembly.html, Assembly.jpeg, Assembly.md, Assembly.pdf, Assembly.png, a.png, assembly.txt, autonomy.png, car.png, chain.png, cost.png, dependency.png, divide.png, efficiency.png, faster.png, and limited.png, all marked as 'first commit'. The right sidebar contains sections for 'About', 'Releases', 'Packages', 'Languages' (showing HTML at 100.0%), and 'Suggested Workflows'.

