

Git & Github Note

Personal Notes of Md Jihad Alom - For Educational Use Only

Author: Md Jihad Alom | Github: <https://github.com/mdjihadalom>

1. Git Setup:

Git Config (User Setup):

Git ব্যবহার শুরু করার আগে আপনার নাম এবং ইমেইল কনফিগার করা উচিত।

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

```
git config --global user.email "youremail@example.com"
```

2. Repository তৈরি ও ক্লোন করা:

নতুন Git Repository তৈরি করা:

একটি নতুন প্রজেক্ট শুরু করার জন্য এটি ব্যবহার করা হয়।

```
git init
```

GitHub থেকে Repository Clone করা:

GitHub থেকে কাজের কপি নিতে এই কমান্ড ব্যবহার করুন।

```
git clone https://github.com/username/repo.git
```

3. ফাইল ট্র্যাক করা ও কমিট করা:

ফাইল staging area-তে যোগ করা:

স্টেজিং এরিয়াতে ফাইল যোগ করতে এই কমান্ড ব্যবহার করুন।

```
git add <filename> # নির্দিষ্ট ফাইল যোগ করা
```

```
git add . # সমস্ত পরিবর্তন যোগ করা
```

কমিট করা:

ফাইলের পরিবর্তন সেভ করতে এবং একটা মেসেজ দিতে এই কমান্ড ব্যবহার করুন।

```
git commit -m "Your commit message"
```

Git & Github Note

কমিটের অবস্থা চেক করা:
পরিবর্তনের অবস্থা দেখার জন্য।

```
git status
```

শেষ কমিট চেক করা:
পূর্ববর্তী কমিটের হিস্ট্রি দেখতে।

```
git log
```

4. রিমোট রিপোজিটরি সাথে কাজ করা:

রিমোট রিপোজিটরি সংযুক্ত করা:
GitHub-এ রিমোট রিপোজিটরি সংযুক্ত করতে।

```
git remote add origin https://github.com/username/repo.git
```

প্রকল্প পুশ করা (**GitHub**-এ কোড আপলোড করা):
GitHub-এ কোড আপলোড করতে এই কমান্ড ব্যবহার করুন।

```
git push -u origin main
```

রিমোট থেকে পরিবর্তন আনানো (**Pull** করা):
রিমোট রিপোজিটরি থেকে সর্বশেষ আপডেট নেওয়ার জন্য।

```
git pull origin main
```

5. ব্রাঞ্চ ব্যবহার:

নতুন ব্রাঞ্চ তৈরি করা:
নতুন ফিচার বা বাগ ফিক্সের জন্য ব্রাঞ্চ তৈরি করুন।

```
git branch new-branch
```

নতুন ব্রাঞ্চে সুইচ করা:
তৈরি করা ব্রাঞ্চে সুইচ করতে।

```
git checkout new-branch
```

Git & Github Note

একই সাথে নতুন ব্রাঞ্চ তৈরি এবং সুইচ করা:
একই কমান্ডে ব্রাঞ্চ তৈরি এবং সুইচ করতে।

```
git checkout -b new-branch
```

ব্রাঞ্চ মার্জ করা:
কাজ শেষ হওয়ার পর নতুন ব্রাঞ্চ মেন ব্রাঞ্চে মার্জ করতে।

```
git merge new-branch
```

ব্রাঞ্চার তালিকা দেখা:
আপনার বর্তমান ব্রাঞ্চার তালিকা দেখতে।

```
git branch
```

6. পরিবর্তন গুলি পূর্বাবস্থায় ফেরানো:

Unstaged পরিবর্তন **Undo** করা:
স্টেজ না করা পরিবর্তনগুলো ফিরে পেতে।

```
git restore <filename>
```

Staged পরিবর্তন **Undo** করা:
স্টেজ করা পরিবর্তনগুলো ফিরিয়ে নিতে।

```
git reset HEAD <filename>
```

সব পরিবর্তন **Undo** করা:
সমস্ত পরিবর্তন পূর্বাবস্থায় ফিরিয়ে নিতে।

```
git reset --hard
```

7. ফাইল ইতিহাস দেখা:

নির্দিষ্ট ফাইলের ইতিহাস দেখা:
একটি নির্দিষ্ট ফাইলের পূর্ববর্তী কমিটের ইতিহাস দেখার জন্য।

```
git log <filename>
```

কোনো পরিবর্তনের পরে কোডের কী পরিবর্তন হয়েছিল তা দেখা:
কোডের পরিবর্তন দেখতে।

Git & Github Note

git diff

8. রিমোট রিপোজিটরি সাথে সংযোগ বিচ্ছিন্ন করা:

রিমোট রিপোজিটরি মুছে ফেলা:

রিমোট রিপোজিটরি মুছে ফেলার জন্য।

git remote remove origin

9. Git স্ট্যাটাস ও তথ্য:

Git Status চেক করা:

ফাইলের বর্তমান অবস্থা এবং স্টেজিং এরিয়া দেখতে।

git status

Git Configuration দেখার জন্য:

কনফিগার করা সেটিংস দেখতে।

git config --list

10. লগ দেখার কমান্ড:

Commit History দেখা:

সমস্ত কমিটের ইতিহাস দেখতে।

git log

সংক্ষেপে কমিট হিস্ট্রি দেখার জন্য:

ছোট আকারে কমিট ইতিহাস দেখার জন্য।

git log --oneline

গ্রাফিক্যাল লগ দেখার জন্য:

গ্রাফিক্যাল ফর্ম্যাটে লগ দেখতে।

git log --graph --oneline --all

Git & Github Note

আরও কিছু গুরুত্বপূর্ণ কমান্ড:

ফাইল/ফোল্ডার সরানো বা নাম পরিবর্তন করা:
ফাইল বা ফোল্ডারের নাম পরিবর্তন বা সরানোর জন্য।

```
git mv <old-filename> <new-filename>
```

স্টেজড ফাইলের সাথে নতুন পরিবর্তন যোগ করা:
পূর্ববর্তী স্টেজড ফাইলগুলোর সাথে নতুন পরিবর্তন যোগ করতে।

```
git add -u
```

রিমোট ব্রাণ্শের সাথে সম্পর্কিত অবস্থা দেখার জন্য:
রিমোট ব্রাণ্শের অবস্থা দেখতে।

```
git remote show origin
```

Git-এ ট্যাগ ব্যবহার করা:
নির্দিষ্ট ট্যাগ ব্যবহার করতে।

```
git tag <tagname>
```

ট্যাগ দেখতে:
আপনার Git রিপোজিটরির ট্যাগ দেখতে।

```
git tag
```

ট্যাগ পুশ করা:
ট্যাগ পুশ করতে।

```
git push origin <tagname>
```

1. Navigation Commands

- `pwd` - Current directory path
- `ls` - List files and folders
- `cd <folder>` - Change directory
- `cd ..` - Go back one folder

Git & Github Note

2. File and Folder Management

- `touch <file>` - Create a new file
- `mkdir <folder>` - Create a new folder
- `cp <source> <destination>` - Copy files/folders
- `mv <source> <destination>` - Move or rename files
- `rm <file>` - Delete a file
- `rm -r <folder>` - Delete a folder

3. Viewing Files

- `cat <file>` - Show file content
- `nano <file>` - Edit files in terminal

4. System Information

- `date` - Show date and time
- `whoami` - Show current user
- `clear` - Clear the terminal

5. Exiting Terminal

- `exit` - Close the terminal