Code . দিয়ে vs code open হবে।

Git checkout –f

সবকিছু ডিলিট করলেও একদম সবশেষ commit অবস্থায় নিয়ে আসবে।

Git log

Git log --oneline

Git log -p -1

এখানে ১ দিয়ে লাস্ট একটা কমিট দেখাবে ২ দিলে ২ টা দেখাবে।

কমিট করলে আর ডিফারেন্স দেখা যায় না। এটা ‍stage area এর জন্য কাজ করে।

Git diff

Git add -A দিয়ে সব কিছু অ্যাড করা হবে। file, folder

Git rm test.txt –f {এই ফাইলটি একদম পিসি থেকে ডিলেট হয়ে যাবে}

-f না দিলেও হবে।

গিট এই কোড এর জন্য delete করে তাকে stage এ নিয়ে যাবে। তারপর আমাদের কে commit করতে হবে।

তবে এটা stage area এর জন্য।

.gitignore { এটাকে কমিট করে দিতে হবে। এই ফাইল টাকে গিট ইগনোর করবে।

কোন folder কে ignore করতে : img/

কোন ফাইল কে ignore করতে : শুধু ফাইল এর নাম লিখলেই হবে।

আরো কিছু : \*.jpg বা \*.png বা \*.js

কোন ফাইলকে রিনেম করার জন্য

Git mv file name rename file name

File name: ফাইল এর নাম

Rename file name: পরিবর্তন হয়ে যেই নাম হবে।

git rm --chach filename গিট এই ফাইলকে ট্রাক করবে না।

Git branch ব্যান্চ দেখার জন্য

Git branch mini {মাস্টার এর কপি}

Git checkout –b main {create a branch and this time is switch this branch}

Git checkout mini { Switch to mini branch }

Git checkout master মাস্টার এর আসলে মিনি এ করা পরিবর্তন দেখা যাবে না।

আবার মিনিতে চেকআউট master branch এ গেলে মিনিতে করা পরিবর্তন দেখা যাবে।

এখানে একটি কথা বলে রাখতে হবেঃ চেক আউট এর ফলে আমার ডিলিট করলে চলে যাবে, আপডেট করলে আপডেট হবে তবে আমি যেই যেই checkout এর যেতে চাই তাতেই যেতে পারবো।

master এ থেকে

Git merge master –m ”this is comment”

Git merge mini

$ git push origin master

git push -u origin master

**SSH**

1. git remote add origin git@github.com:Ismail-MIah/practice2.git
2. এই লিংক এর সাহায্যে আমার git hub এর সাথে git connect করবো।
3. এই লিংকে আমরা origin নামে চিনবো।
4. Git remote –v { Origin link }
5. Git push origin master . { কাকে push করবোঃ origin, কোন branch কেঃ master}
6. এখন আামার read write permission টা দিতে হবে।
7. Github 🡪 seting 🡪 ssh and gpg keys 🡪 { Generating a new SSH key}
8. ssh-keygen -t ed25519 -C "*your\_email@example.com*" মেইল টা চেন্জ করে git এ পেস্ট করে enter করলে 🡪
9. Enter file in which to save the key (path): ………. এখানে যে কোন নাম দিলেই হবে ।তবে আমি git দিবো। তার পর তিন বার enter দিবো।
10. প্রোসেস আইডি তৈরি করার কোড ঃ eval "$(ssh-agent -s)"
11. ssh-add ~/ git যে ফাইলের কিছুক্ষণ পূর্বে নাম দিয়েছিলাম সেই ফাইল লোকেশন দিতে হবে
12. ssh key git .pub ফাইলে থাকে। তা github এ দিয়ে দিবো।
13. এবং একটা title দিবো। ‍save করতে হবে।
14. এখন git and git.pub ফাইল দুটোকে add ও commit করে দিতে হবে। কারণ এগুলো আনট্র্যাক অবস্থায় আছে।
15. এখন git and git.pub ফাইল দুটোকে add ও commit করে দিতে হবে। কারণ এগুলো আনট্র্যাক অবস্থায় আছে।
16. মাস্টার branch এ থেকে ঃ git push –u origin master.
17. ফাইলগুলো এখন github এ আপলোড হবে। {শুধু master branch } অন্য ব্যান্চ গুলো আপলোড করার জন্য প্রথমে checkout করে সেই branch এ যেতে হবে। যেমন ঃ mini branch
18. Git checkout mini
19. git push –u origin mini.
20. অন্য অন্য branch গুলো upload একই ভাবে করতে হবে।

আমার কম্পিউটারে lagecy format সাপোর্ট করে না।এটা lagecy format এর জন্য না। নিচে দেখ:-

]Github এ একটা repository তৈরি করে তাকে আমাদের করা প্রজেক্ট এর সাথে যুক্ত করবো।

নতুন repository তৈরি করার সময় কি কি করতে হবে তা github বলে দেয়।

Git remote add origin (urL)

Git remote –v

Git push origin master

Git branch mini এটা একটা ব্যান্চ তৈরি করবে । তবে তাতে সুইজ করবে না।

git checkout -b main এটা একটা ব্যন্চ তৈরি করবে এবং তাতে সুইজ কারবে।

branch condition:

modification -> stage -🡪 branch

কোন মডিফাইকে stage এ নিলে সেটা ওই branch এর অংশ হয়ে যাবে। এক্ষেত্রে checkout ব্যবহার করা ভাল।

Mini branch ke master branch er shathe marge krar jno first checkout master branch. Then use this code : git merge mini -m "I can marge mini "

\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Git checkout -- filename কেন ব্যবহার করা হয়?

ফাইল modify করলে যদি মনে হয় এই modify এর কোন প্রয়োজন নেই। তখন উক্ত code ব্যবহার করে head commit এর অবস্থায় ফাইলকে নিয়ে যাওয়া যায়।

শুরু সেই ফাইল যেটা আমি modify করতে চাচ্ছি না।

পুরো repository কে শেষ কমিট এ না নিয়ে শুধু ওই ফাইল কে নিবে।

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **git config –global user.name “ismail-miah”** | | git config –global user.email “miahismail654@gmail.com” | | Git config --list | | git config –global user.name “raju” | | Git config --list | | pwd | | ls / ls -a | |
| |  | | --- | | **git init** | | git init হলে .git আবসে | | Touch day1.txt বা echo > | | Open day1.txt bash command not found | | Cat day1.txt ফাইলিএর text দেখা যায়। | | Git restore day1.txt state এ নেওয়ার আগে। | | Git add | | Git add -A | | Git add . | |
| |  | | --- | | **Git add \*.js** | | Git add \*\*/\*.js | | Git diff | | Git commit –m “…………………….” | | Git status | | Git log | |
| |  | | --- | | **Git reset –soft HEAD^**  **Recent commit is uncommit** | | Git reset HEAD^  Recent commit is uncommit and remove staging area. | | Git reset –hard HEAD^  Recent commit is uncommit and file back first code. | | Git diff id1 id2 দুইটা কমিট এর মধ্যে পার্থক্য  Id1 : লিস্টের নিচের কমিট এর id  Id2 : লিস্টের উপরের কমিট এর id | | Git diff - - staged staged করার পরের diff দেখার জন্য | | Credential Manager | | clip < ~/.ssh/id\_rsa.pub এই কোড দ্বারা ssh key copy হয়ে clipboard এর  আসবে। | | tail ~/.ssh/id\_rsa.pub এই কোড দ্বারা ‍ssh key bit bash এ আসবে। | | rm-rf .gt এই working directory আর git repository হিসাবে থাকবে না। | |
|  |
| |  | | --- | | **Git log --oneline** | | Git show | | Git show id {কমিটে কি চেন্জ হয়েছে তা দেখায় }  Id: কোথায় পাব?  Git log –oneline দিয়ে id নিয়ে নিবো। | | Git checkout id | | চেক আউট ব্যবহার করা হয় শুধু সেফ ভাবে পূর্বের কমিট গুলো read করার জন্য। | | Git checkout master  এটা masterbranch এ নিয়ে আসেবে। | | Git checkout –d branchname delete a branch  {মার্জ না করলে কোড কাজ করবে না।} | |
| |  | | --- | | **git remote** | | Git remote add origin (url --- github থেকে) | | Git remote -v | | Git clone (url) best way for github to local | | Git clone (url) file new name. | | Clone🡪 fetch 🡪 pull | | Git branch –D branchName এভাবে কোন branch কে delete করতে হয়। | | আমাদের তৈরি কোড টি ডিফিকাল মনে হচ্ছে তাই চাচ্ছি অন্য কোন নিয়মে কোড  করার জন্য তবে পূর্বের কোড গুলো হারাতে চাচ্ছি না।  এই জন্য git stash | | Git stash pon বা ‍apply সবশেষ stash ‍show করবে। | | Git stash list | | Git stash pon লিস্টে থেকে যেটাতে যেতে চাই সেই stash@{1} | | Git clean –f যতগুলো untrack file আছে তা delete হবে। | | Git commit –a –m “comment” ‍staging area কে ফাকিঁ দিয়ে কমিট করবে। | | Git log –p দিয়ে কমিট এর ডিফারেন্স দেখা যায়। | | Git log –p -3 | | Git log –stat শর্টকাটে বলবে কি কি কমিট করা হয়েছে। | | Git log –pretty=full | | Git log –pretty=short | | Git log –pretty=oneline not working | | git log --since=2.weeks দুই সপ্তাহে যা কমিট করা হয়েছে। | | git log --since=2.months দুই মাসে যা কমিট করা হয়েছে। | | Git scm userfull options for git log formate | | Git log –pretty=format: “%h --%an” | | Git commit –amend কমিট এর ম্যাজেস এডিট করার জন্য। | | git config –global alias.last ‘log –p -1’  এখন git last লিখলে git log –p -1 এর কাজ করবে। | | git branch –v {show all branch last commit} | | git branch --merged --> already merged branchs. | | git branch --no --merged --> not already merged branchs. | | কোন branch ডিলিট করার জন্য  git branch --d mimi --> এই ব্যান্চ মার্জ করা না হলে গিট মনে করবে কেউ ভল  করে উক্ত branch কে ডিলিট করতে চাচ্ছে।  message: git error: if mini is not merge  git branch --D mimi এখন মার্জ করা না হলেও ডিলিট হবে। | | কোন branck কে github এ অন্য নামে push করার কোড:  git push origin bugfix:mybugfix | | github থেকে কোন branch কে delete করার জন্য কোড:  git push –d origin branchName | |  | |  | |  | |
|  |
|  |

প্রথমে github রিপোজিটরি তৈরি করে সেটা (https) link copy করে deshktop এ যে কোন নাম দিয়ে clone করবো।

যেমন: gitbash দিয়ে working directory desktop এ যাবো।

তারপার পর কোড: git clone link

Ssh key কোন ফাইলে নিবো না। পাসওয়াড় দিবো না।

তিনটি enter দিবো।

1. ssh-add ~/ git যে ফাইলের কিছুক্ষণ পূর্বে নাম দিয়েছিলাম সেই ফাইল লোকেশন দিতে হবে।

এই ফাইল তৈরি করবো না। আর লোকেশন ও দিবো না।

**ssh-add ~/.ssh/id\_rsa**

**এটা দিলে identity add হবে।**

tail ~/.ssh/id\_rsa.pub

github issues কেন ব্যবহার করা হয়?

New issues --> name --> description ---> assignes --> bug--> submit

যখন issues submit করবো তখন তার একটা id আসবে।

কোড এ গিয়ে issues solve কিভাবে করবো?

commit করার সময় সাথে issues id দিলে git বুঝবে issues solve করা হয়েছে।

fast forward merge