

গিট ইন্সটল করার পর সবার আগে গিটের কনফিগারেশন করা উচিত। তো সেইসব কমান্ডগুলো কি?(আমরা বোঝার সুবিধার্থে সব কমান্ডের শুরুতে হ্যাশ “#” দিব,কমান্ড লাইনে যেন আবার ‘#’ লিখে ফেল না।)

#git config --global user.name “user_name” (গিটহাব account এর নাম)

#git config --global user.email “user_email” (গিটহাব যে মেইল আইডি দিয়ে খুলা হয়েছে।)

উপরের কাজগুলো শুধুমাত্র একবার করলেই চলবে। কিন্তু যদি অন্য কারো অ্যাকাউন্ট এ গিয়ে কাজ করতে হয় তবে এই কাজ আবারও করতে হবে। অর্থাৎ আবার তার নাম এবং মেইল আইডি।

এবারে আসি গিট কন্ট্রোলিং এর ব্যাপারে। সবার প্রথমে গিটহাবে একটা রিপোসিটোরি খুলতে হবে। সেটাতে পছন্দসই নাম দিতে হবে। ঐ রিপোসিটোরিতে গেলে যে HTTP লিংক আছে সেটি কিন্তু আমাদের পরের বেলা লাগবে, তাই সবচেয়ে ভাল হয় ঐ লিংকটি একটি .txt ফাইলে কপি করে রাখি। এটিতো গেলো অনলাইনের কাজ এবারে আসি লোকালি কি কি কাজ করতে হবে। এবারে কিছু কাজে সিকুয়েন্স মেইন্টেন করা করতে হবে।

- প্রথমে একটা ফোল্ডার নিতে হবে। যেটিতে আমাদের কোড/ফাইল/ফোল্ডার সমূহ থাকবে বা আমরা গিটে আপলোড করে রাখব।
- ঐ ফোল্ডারে আমাদের কমান্ড-উন্ডো আনতে হবে। সেক্ষেত্রে যদি কেউ উন্ডোজ ইউজার হয় তাহলে shift+(right-click) করলেই দেখা যাবে (open command window here) সেখানে কমান্ড দিতে হবে #git init এই কাজটি আমাদের সম্পূর্ণ কাজের জন্য একবার করলেই যথেষ্ট। আসলে গিট ইনিশিয়ালাইজ করা। এই কমান্ডের মাধ্যমে আমাদের সিস্টেম বুঝতে পারে এটা পরে গিটের সাথে অ্যাড করা হবে।
- এখন ঐ ফোল্ডারে গিয়ে আমাদের পছন্দ মত ফাইল/ফোল্ডার নেই। আমরা চাইলে অন্য ফোল্ডার থেকে কপি করেও আনতে পারি। কিন্তু সাবধান ঐ ফোল্ডার/ফাইলটি যেন কমিট করা না থাকে। (কমিট!! এটা আবার কি? কিছুক্ষণ পরেই জানতে পারব।)
- ঐ ফোল্ডার বা ফাইলটি গিটে অ্যাড করতে হবে যা লোকালি হবে। অ্যাড করার কমান্ড
#git add file.txt (file.txt বলতে আমি ফাইলের নাম এবং .txt বলতে আমি ঐ ফাইলের এক্সটেনশন বুঝাচ্ছি।)
#git add folder_name (আমি যদি ফাইল না নিয়ে ফোল্ডার নেই তখন এভাবে লিখলেই হবে)
(একটা কথা বলে রাখা ভাল, আমরা গিটের ফোল্ডারে যাই নেই না কেন সেটা যদি গিটে অ্যাড না করি বা git add কমান্ড টি না দেই তবে সেটা অনলাইনের মাধ্যমে অ্যাড করতে পারব না।)
(গিট অ্যাড করার আরো সিস্টেম রয়েছে। সব এক্সট্রা ইনফো নিচে হয়েছে। আপাতত এতটুকু শিখলেই চলবে।)
- এবারে যে জিনিসটা নিয়ে কথা বলব সেটি হল গিট কমিট করা। আমরা খুব ভাল লেগেছে এই জিনিসটি দেখে। কমিট ঠিক কতখানি গুরুত্বপূর্ণ জিনিস সেটি শুরুর দিকে বোঝা যাবে না যদি না আমরা নিয়মিতভাবে গিট নিয়ে কাজ না করি। কমিট হল একটি ফাইলে কি কাজ করা হয়েছে বা ঐ ফাইল/ফোল্ডার এ আমি কি নিয়ে লিখছি সেটা নিয়ে। আমরা অনেক সময় ফাইলের পরিবর্তন নিয়ে আসি,যেটা পরে ভুলে যাই যা কমিটে লিখে রাখা যায়, এমনকি কমিট করার সময় আসলে কি কি পরিবর্তন করেছি সেটিও দেখা যায় (এই কথাগুলো মাথার উপর দিয়ে গেলে খুব বেশী অবাক হব না)।কমিট করার জন্য কমান্ড
#git commit -m “commit_messages” file_name (file কমিট করার জন্য)
#git commit folder_name “commit_messages” (ফোল্ডার কমিট করার জন্য)
(শুধু যে অ্যাড করলেই খেল খতম তা নয় কিন্তু, কমিট করতে হবে। কমিট না করলে ঐ ফাইল/ফোল্ডারটি গিটহাবে আপলোড করতে পারবে না।)
- ফাইল বানালাম, ফাইলে কাজ করলাম, ফাইল অ্যাড করলাম, ফাইল কমিট করলাম। বাকি থাকল আপলোড করা, মানে অনলাইনে স্টোর করা। যেটির জন্য এত কাহিনি করলাম আমরা। আর এই জন্য যে কাজটি করব আমরা
#git remote add origin “http_link” (এটি হল সেই http_link যেটার কথা আমি একবারে শুরুতে বলেছিলাম .txt ফাইলে সেভ করে রাখতে বলেছিলাম। মাথায় রেখ লিংকটি কিন্তু “” ইনভার্টেড কমা মার্কের মাঝে রাখতে হবে।)
- এবারে পুশ বা আপলোড করার পালা এটির জন্য যা করতে হবে
#git push -u origin master (অনেক সময় এই কমান্ডে কাজ নাও করতে পারে, প্রকৃতপক্ষে এই কমান্ডের কাজ করা উচিত কিন্তু নতুন হিসেবে অনেকের ভুল হয়। কি ভুল, কোথায় ভুল তা আর বুঝে উঠতে পারে না।
সেক্ষেত্রে আরো কয়েকটি ভিন্ন কমান্ড আছে:
#git push origin master --force অথবা
#git push -u --force origin master
মাথায় রেখ এই কমান্ডগুলো সচরচর ব্যবহার করবে না। উপরের কমান্ডটি না কাজ করলে তবেই এটি করবে।)
- এবারে user_name মানে যেনামে গিট অ্যাকাউন্ট খোলা হয়েছে এবং password দেয়ার পালা।

ব্যস হয়ে গেল। আমাদের প্রথম ধাপ শেষ। আরো কিছু কাহিনী আছে।

১। কোন ফোল্ডারের গিটের সব ইতিহাস দূর করে দেয়ার একখান কমান্ড

#rm -rf .git

২। কোন ফাইল যদি কমিট করা না হয় বা অ্যাড করা না হয় সেটা দেখার জন্য কমান্ড

git status (কোন ফাইল অ্যাড বা কমিট করা না থাকলে বলে দিবে)

৩।গিট ক্লোন করার জন্য কমান্ড

- `#git clone http_link` (মাথায় রাখি এখানে কোন "" দেয়া লাগবে না)
- সব রকমের এডিট আগের মতি করা যাবে। শুধুমাত্র কমিট এর বেলায় অন্য কথা প্রযোজ্য। আমরা যেসব ফাইলের/ফোল্ডারে চেঞ্জ আনব শুধুমাত্র সেসবেই কমিট করতে পারব। আর তানা হলে পারব না।
- গিটে কি কি পরিবর্তন আনা হয়েছে আগের সাথে সেটার জন্য কমান্ড
`#git diff`
কি কি পরিবর্তন এসেছে সব দেখাবে দিন-তারিখ সহ।
- এবারে পুশ করার জন্য আগের কমান্ডের থেকে একটু ভিন্ন কমান্ড দিতে হবে।
`#git push -u origin master` (আলাদা করে রিপো লিংক দিতে হবে না।)

৪। সব ফাইল একসাথে কমিট করার টেকনিক। এটি করলে কিন্তু সব ফাইল/ফোল্ডারে একই রকম কমিট হবে যেসবে চেঞ্জ আসছে যেটা ভাল কিছু নয়, তবুও কাজ করার সুবিধা এনে দেয়।

```
# git commit -am "commit message"
```

৫। গিটের আগের সব ইতিহাস দেখার জন্য কমান্ড। মানে কখন কি হইছে, কি কমিট করা হইছে, কি পরিবর্তন আসছে সেসব
`#git log`

৬। যদি পার্টিকুলার কোন ফাইলের কমিট ইতিহাস দেখতে হয় তবে `###` এই কমান্ডের সময় ফাইলের যে হ্যাশ কোড থাকে সেটার সামনের দিকে ৫-৭ টা লেটার কপি করে কমান্ড দিলে সব দেখা যাবে।

```
#git show f9ab3455
```

৭। ধরা যাক, আমার কোডে যখন আমি তৃতীয় কমিট করেছিলাম, তখন আসলে কোডের কি অবস্থা ছিল যদি সেটা দেখতে চাই তবে আমায় আগে গিট লগ থেকে ঐ ফাইলের হ্যাশ ভ্যালু বের করতে হবে তারপর কমান্ড দিতে হবে

```
#git checkout has_value
```

এখন আমাদের পুরো প্রজেক্ট কপি করতে চাই, তবে সেটিও করতে পারি, এটা কিভাবে হবে সেটা এখনো শিখি নাই। তবে কি কি ফাইল/ফোল্ডার ছিল সেটি দেখার জন্য `"ls"` দিলেই হবে। আবার কোন ফাইলে আসলে কি কি ছিল সেটা দেখতে হলে কমান্ড দিব `#cat file_name`। এইতো ব্যাপার। আর যদি আমরা একেবারে বর্তমান অবস্থায় ফিরে আসতে চাই তবে আমাদের লিখতে হবে `#git checkout master`।

৮। পুরনো কোন নির্দিষ্ট ফাইলকে যদি আমরা ফেরত পেতে চাই। মানে পুরো প্রজেক্টের শুধুমাত্র একটি ফাইল পেতে চাই, তাহলে আমাদের অন্য কমান্ড লিখতে হবে।

```
#git checkout has_value file_name
```

এখন আমরা চাইলেই ঐ ফালকে নিতে পারি খুব সহজেই। `#cat file_name` এটি লিখলেই আমরা ঐ ফাইলের তৎকালীন অবস্থা দেখতে পারব। আর যদি পুনরায় আগের অবস্থায় ফেরত যেতে চাই তাহলে আমাদেরকে ভিন্ন কমান্ড লিখতে হবে।

```
#git checkout master -f
```

৯। যদি আমরা কখন ভুল কমিট করে বসি তাহলে আমাদের রিসেট করতে হবে। পুশ-পুলে আমাদের এই ব্যাপারে খুব কাজ লাগবে। কোন প্রজেক্ট যদি আমরা একবার পুশ করে ফেলি তাহলে ভুলেও সেটা আর রিসেট করা যাবে না। এখন বলি রিসেট দুই প্রকার:

(I) **soft:**

এটির মাধ্যমে আমরা ঐ কমিট ডিলিট করতে পারব এবং ঐ কমিট চলাকালীন ঐ ফাইলে যেসব পরিবর্তন আনা হয়েছে সেসবের কোন পরিবর্তন আসবে না। সেক্ষেত্রে গিট লগ বের করতে হবে। আর আমরা ঠিক যে কমিট পর্যন্ত রাখতে চাই ঠিক সেই কমিট এর হ্যাশ ভ্যালু জানতে হবে। যেমন:

```
f98340456jnhk456346 (last commit)
```

```
f65g43534bk3454355 (3rd commit)
```

```
f435jl3453j4l545435ll (2nd commit)
```

```
f58765ljlghj35453453 (1st commit)
```

ধরে নেই যে আমরা **3rd commit** আর **last commit** রাখতে চাই না কিন্তু এই সময়ে কাজগুলো হয়েছে সেসব ফাইলে থাকবে সেক্ষেত্রে আমাদের কমান্ড হবে `#git reset --soft f435jl3453j4l545435ll`। কিন্তু এরপরে আমাদের আবার কমিট করতে হবে সাধারণ নিয়মে কারণ ঐ ফাইলের গিট লগে তার কমিট হিস্টোরি আছে কিন্তু এই মুহূর্তে তার কমিট কি হওয়া উচিত সেটা সে জানে না। সত্যি বলি এটি হল এক ধরনের জান বাচান ফিচার গিট হাবের।

(II) **hard:**

নামের সাথে কাজের মিল। এটি একই সাথে কমিট এবং ঐ সময়ে হওয়া কাজগুলোকে ডিলিট করে দেয়। আমি আসলে ঠিক যতটুকু পর্যন্ত ডিলিট করতে চাই গিট লগের মাধ্যমে দেখে নিতে পারি। তারপরে আবার ঐ হ্যাশ ভ্যালু এবং সেটা কপি করে রেখে দেয়া। অতঃপর কমান্ড:

```
#git reset --hard hash_value
```

১০। গিটের মেমোরি ঠিক কতখানি সেটা নিয়ে আমার সন্দেহ হয়। এটিতে `reflog` নামের একটি কমান্ড আছে। এহেন কোন জিনিস নাই

যে সেটা মনে রাখতে না, সেটা হোক কমিট করা, ফাইল এডিট করা, কমিট রিসেট করা সব সে মনে রাখে। আমরা ভুল করেও কোন কিছু রিসেট করতে পারি হতে পারে সেটা সফট কিংবা হার্ড। সেক্ষেত্রে কি করতে করব। তখন আমাদের reflog থেকে দেখে নিতে হবে আসলে কি কি কাজ কখন করেছি, আর সেসব থেকে আমাদের ঠিক ক্রে নিতে হবে আসলে আমাদের কি পুরোন কাজে আমরা ফেরত যেতে চাইছি। কমান্ড লাইনঃ #git reflog.

এইখানে আমাদের কাজের ইতিহাস তুলে ধরা হবে। এরপর আমরা যে স্টেজে ফিরে যেতে চাই সেই স্টেজের HEAD টা মনে রাখতে হবে। অতঃপর কমান্ড দিতে হবেঃ #git reset HEAD(@)8 । এই কমান্ড দিলে আমাদের ফাইলগুলো ঐ কমিট করার সময় যেসকল ছিল ঠিক সেভাবে থাকবে। এতে একটি সমস্যা আছে। সম্পূর্ণ প্রজেক্ট আগের ঐ কমিটের অবস্থায় ফিরে যায়। মানে শুধুমাত্র যে ঐ ফাইল চেঞ্জ হয় তা নয়, পুরো প্রজেক্ট চেঞ্জ হয়। HEAD(@)8 এই কমান্ডের পর কমান্ড হবেঃ

#git reset hard অবশেষে আমাদের এই কমান্ড শেষ হল।

১১। অনেক সময় দেখা যায়, আমি পুশ করলাম আর গিটহাবে গিয়ে দেখি আমার ফাইল নাই, কিন্তু কনসোল বলছে আমার গিট করা

হয়ে গেছে। তাহলে, আমরা পুশ করলাম কই? বিশাল বিপদ!! যদি দেখতে চাই, কোথায় আমরা আপলোড করেছি সেক্ষেত্রে কমান্ড দিতে হবেঃ #git remote show origin তাহলে আমাদের দেখাবে আমরা কোন লিংক/রিপোসিটোরিতে পুশ করেছি। সহজ না!!