

Git Commands

- ပထမဦးစွာ git ကိုစသုံးမည်ဆိုပါက git တွက် global username & global emailကိုအရင်သိအောင်လုပ်ပေးရပါသည်။

For global username,

```
git config --global user.name <name>
```

For global username,

```
git config --global user.email <email>
```

1. `git init` ⇒ git local repository တခုတည်ဆောက်တဲ့အခါအသုံးပြုပါသည်။
2. `git add` ⇒ staging area ထဲကိုပို့ချင်တဲ့အခါအသုံးပြုပါသည်။ file name or (.) ဖြင့်ပေးနိုင်ပါသည်။
3. `git commit -m ".."` ⇒ ကိုယ်ပို့လိုက်တဲ့ staging file တွက် message တခု အမျိုးအမည်ပေးရပါသည်။*(-m) သည် message ကိုဆိုလိုပါသည်။
4. `git status` ⇒ modify or staging ခြေနေကိုကြည့်ချင်တဲ့အခါ git status ကိုအသုံးပြု၍ကြည့်နိုင်ပါသည်။
5. `git log --oneline` ⇒ git commit history ကြည့်ချင်တဲ့အခါ git log ဖြင့်ကြည့်နိုင်ပါသည်။
6. `git diff` ⇒ working directory & index file ကြားမှာ unstage ဖြစ်တဲ့ file တွေကြည့်ချင်တဲ့အခါအသုံးပြုပါသည်။

7. `git checkout` \Rightarrow `commit` တခုနှင့်တခုကိုကြည့်ချင်တဲ့အခါ အသုံးပြုပါသည်။(** ကြည့်ရှုသာရပါသည် ကျန်တာဘာမှလုပ်၍မရပါ)
8. `git revert` \Rightarrow `commit` တခုကို `safe remove` လုပ်ချင်တဲ့အခါအသုံးပြုနိုင်ပါသည်။ (** ထူးခြားချက်အနေဖြင့် `remove` လုပ်လိုက်သော `commit` တခုကို ထပ်ပြီး `commit` တခုအနေဖြင့်သိမ်းပေးပါသည်။အကယ်၍ ပြန်ခေါ်သုံးချင်တဲ့အခါအသုံးပြုနိုင်ပါသည်။)
9. `git reset` \Rightarrow `reset command` သည် `commit` တခုကို `remove` လုပ်ချင်တဲ့အခါအသုံးပြုပါသည်။သို့သော် ကိုယ်လုပ်လိုက်တဲ့ `commit` တခုကိုသာမကနောက်မှာရှိသော `commit` ကိုပါဖျက်ပေးသွားမှာဖြစ်ပါသည်။(** ထူးခြားချက်ကတော့ ပြန်ခေါ်၍မရတော့ပါ `git revert` ဖြင့်ပြောင်းပြန်ဖြစ်ပါသည်။)
10. `git branch` \Rightarrow `git branch` ကိုကြည့်ချင်တဲ့အခါအသုံးပြုပါသည်။
11. `git branch <name>` \Rightarrow `git branch` တည်ဆောက်ချင်တဲ့အခါအသုံးပြုပါသည်။
12. `git checkout -b branchname` \Rightarrow `git branch` တည်ဆောက်ချင်တဲ့အခါအသုံးပြုနိုင်ပြီး ဆောက်လိုက်သော `branch` ကိုပါ `switch` လုပ်ပေးပါသည်။
13. `git checkout -D branchname` \Rightarrow `git branch` ကို ဖျက်ချင်တဲ့အခါအသုံးပြုပါသည်။
14. `git merge` \Rightarrow `git branches` တွေကိုပေါင်းချင်တဲ့အခါအသုံးပြုပါသည်။
15. `git stage` \Rightarrow လက်ရှိရေးနေတဲ့ `source code files` တွေကိုတနေရာပို့ထားချင်တဲ့အခါအသုံးပြုပါသည်။(** သတိပြုစရာတချက်က `add` မလုပ်ရသေးတဲ့ `file` ကိုသာလုပ်၍ရပါသည်။)
16. `git stage pop` \Rightarrow `git stage` ဖြင့်တနေရာပို့ထားသော `code` ကိုပြန်ခေါ်ချင်ပါက အသုံးပြုနိုင်ပါသည်။
17. `git pull` \Rightarrow `online` ပေါ်က `files` တွေကိုစစ်ချင်ပါက `git pull` ဖြင့်ဆွဲချရပါသည်။(ဥပမာ- တူညီသော `file` ကိုမတူညီတဲ့ `branch` တခုစီကဝင်ပြင်ခဲ့မည်ဆိုပါက `conflict` ဖြစ်သွားနိုင်ပါသည်။ ဒါတော့ တခြား

branch ကရေးလိုက်သော file ကို pull အရင်ဆွဲချပြီးမှသာ push လုပ်၍ရနိုင်ပါသည်။)

18. `git push` ⇒ cloud repository ပေါ်သို့တင်တဲ့အခါအသုံးပြုပါသည်။

19. `git clone` ⇒ cloud repository တခုကို ခေါ်ယူအသုံးပြုချင်တဲ့ခါ `git clone` ဖြင့် ဆွဲပြီး feature တွေအသစ်ထည့်ကာအသုံးပြုနိုင်ပါသည်။

20. `git fetch` ⇒ online ပေါ်က branch တခုကို local မှာ ထားချင်တဲ့အခါအသုံးပြုပါသည်။