

51 comandos do Git que você deve conhecer (e, portanto, os mais úteis para serem lembrados)..

Git tem muitos comandos diferentes que você pode usar. E descobri que esses cinquenta são os que uso com mais frequência (e, portanto, os mais úteis para serem lembrados).

Sempre começo perguntando : Como configurar? ; Como verificar? ; Como armazenar ? ; Como fazer ? ;Como ver ? Como deletar ? ;

1-Como verificar sua configuração do Git:

O comando abaixo retorna uma lista de informações sobre a configuração do git, incluindo nome de usuário e e-mail:

git config -l

2-Como configurar seu nome de usuário do Git:

Com o comando abaixo você pode configurar seu nome de usuário:

git config --global user.name "ROGÉRIO TAVARES "

3-Como configurar seu e-mail de usuário do Git:

Este comando permite configurar o endereço de e-mail do usuário que você usará em seus commits.

git config --global user.email "rogerio@almeidatavares.com"

4-Como armazenar em cache suas credenciais de login no Git:

Você pode armazenar as credenciais de login no cache para não precisar digitá-las todas as vezes. Basta usar este comando:

git config --global credential.helper cache

5-Como inicializar um repositório Git:

Tudo começa a partir daqui. A primeira etapa é inicializar um novo repositório Git localmente na raiz do projeto. Você pode fazer isso com o comando abaixo:

git init

6-Como adicionar um arquivo à área de preparação no Git:

O comando abaixo adicionará um arquivo à área de preparação. Basta substituir filename_here pelo nome do arquivo que você deseja adicionar à área de preparação.

git add filename_here

7-Como adicionar todos os arquivos na área de preparação no Git

Se você deseja adicionar todos os arquivos do seu projeto à área de teste, pode usar um curinga . e todos os arquivos serão adicionados para você.

git add .

8-Como adicionar apenas determinados arquivos à área de preparação no Git

Com o asterisco no comando abaixo, você pode adicionar todos os arquivos começando com 'fil' na área de preparação.

git add fil*

9-Como verificar o status de um repositório no Git:

Este comando mostrará o status do repositório atual, incluindo arquivos testados, não testados e não rastreados.

git status

10-Como fazer commit das alterações no editor no Git:

Este comando abrirá um editor de texto no terminal onde você pode escrever uma mensagem de confirmação completa.

Uma mensagem de confirmação é composta por um breve resumo das alterações, uma linha vazia e uma descrição completa das alterações após ela.

git commit

11-Como confirmar alterações com uma mensagem no Git:

Você pode adicionar uma mensagem de confirmação sem abrir o editor. Este comando permite especificar apenas um breve resumo para sua mensagem de confirmação.

git commit -m "your commit message here"

12-Como enviar alterações para um repositório remoto no Git:

Quando todo o seu trabalho estiver pronto para ser salvo em um repositório remoto, você pode enviar todas as alterações usando o comando abaixo:

git push

13-Como extrair alterações de um repositório remoto no Git:

Se outros membros da equipe estiverem trabalhando em seu repositório, você pode recuperar as alterações mais recentes feitas no repositório remoto com o comando abaixo:

git pull

14-Como confirmar alterações (e pular a área de preparação) no Git:

Você pode adicionar e confirmar arquivos rastreados com um único comando usando as opções -a e -m.

git commit -a -m"your commit message here"

15-Como ver seu histórico de commits no Git:

Este comando mostra o histórico de commits do repositório atual:

git log

16-Como ver seu histórico de commits incluindo alterações no Git:

Este comando mostra o histórico do commit incluindo todos os arquivos e suas alterações:

git log -p

17-Como ver um commit específico no Git:

Este comando mostra um commit específico.

Substitua commit-id pelo id do commit que você encontra no log de commit após a palavra commit.

git show commit-id

18-Como ver as estatísticas de log no Git:

Este comando fará com que o log do Git mostre algumas estatísticas sobre as alterações em cada confirmação, incluindo linha(s) alterada(s) e nomes de arquivo.

git log --stat

19-Como ver as alterações feitas antes de confirmá-las usando "diff" no Git:

Você pode passar um arquivo como parâmetro para ver apenas as alterações em um arquivo específico.

git diff mostra apenas alterações não testadas por padrão.

Podemos chamar o diff com o --staged sinalizador para ver as alterações preparadas.

git diff

git diff all_checks.py

git diff --staged

20-Como ver as alterações usando "git add -p":

Este comando abre um prompt e pergunta se você deseja preparar as alterações ou não, e inclui outras opções.

git add -p

21-Como remover arquivos rastreados da árvore de trabalho atual no Git:

Este comando espera uma mensagem de confirmação para explicar por que o arquivo foi excluído.

git rm filename

22-Como renomear arquivos no Git:

Este comando organiza as alterações e, em seguida, espera uma mensagem de confirmação.

git mv oldfile newfile

23-Como ignorar arquivos no Git:

Crie um .gitignorearquivo e confirme-o.

Como reverter alterações não testadas no Git:

git checkout filename

24-Como reverter alterações em estágios no Git:

Você pode usar o sinalizador de opção -p para especificar as alterações que deseja redefinir.

git reset HEAD filename

git reset HEAD -p

25-Como alterar o commit mais recente no Git:

git commit --amend permite que você modifique e adicione alterações ao commit mais recente.

git commit --amend

((!!Observação!!: corrigir um commit local com a alteração é ótimo e você pode enviá-lo para um repositório compartilhado depois de consertá-lo. Mas você deve evitar emendar commits que já foram tornados públicos.))

26-Como reverter o último commit no Git:

git revert criará um novo commit que é o oposto de tudo no commit dado.

Podemos reverter o commit mais recente usando o alias head assim:

git revert HEAD

27-Como reverter um commit antigo no Git:

Você pode reverter um commit antigo usando seu id de commit. Isso abre o editor para que você possa adicionar uma mensagem de confirmação.

git revert comit_id_here

28-Como criar um novo branch no Git:

Por padrão, você tem uma ramificação, a ramificação principal. Com este comando, você pode criar uma nova ramificação. O Git não mudará para ele automaticamente – você precisará fazer isso manualmente com o próximo comando.

git branch branch_name

29-Como mudar para um branch recém-criado no Git:

Quando você quiser usar um ramo diferente ou recém-criado, você pode usar este comando:

git checkout branch_name

30-Como listar branches no Git:

Você pode visualizar todas as ramificações criadas usando o git branch comando. Ele mostrará uma lista de todas as ramificações e marcará a ramificação atual com um asterisco e a destacará em verde.

git branch

31-Como criar um branch no Git e mudar para ele imediatamente:

Em um único comando, você pode criar e mudar para uma nova ramificação imediatamente.

git checkout -b branch_name

32-Como deletar um branch no Git:

Quando você terminar de trabalhar com uma ramificação e mesclá-la, poderá excluí-la usando o comando abaixo:

git branch -d branch_name

33-Como mesclar duas ramificações no Git:

Para mesclar o histórico da ramificação em que você está atualmente com o branch_name, você precisará usar o comando abaixo:

git merge branch_name

34-Como mostrar o log de confirmação como um gráfico no Git:

Podemos usar --graph para obter o log de confirmação para mostrar como um gráfico. Além disso, --oneline limitará as mensagens de confirmação a uma única linha.

git log --graph --oneline

35-Como mostrar o log de confirmação como um gráfico de todas as ramificações no Git:

Faz o mesmo que o comando acima, mas para todas as ramificações.

git log --graph --oneline --all

36-Como abortar uma mesclagem conflitante no Git:

Se você quiser descartar uma mesclagem e começar de novo, execute o seguinte comando:

git merge --abort

37-Como adicionar um repositório remoto no Git

Este comando adiciona um repositório remoto ao seu repositório local (basta substituir **https://repo_here** pelo URL do repositório remoto).

git add remote https://repo_here

38-Como ver URLs remotos no Git:

Você pode ver todos os repositórios remotos do seu repositório local com este comando:

git remote -v

39-Como obter mais informações sobre um repositório remoto no Git:

Basta substituir origin pelo nome do controle remoto obtido executando o comando git remote -v.

git remote show origin

40-Como verificar branches remotos que o Git está rastreando:

Este comando mostra o nome de todos os branches remotos que o Git está rastreando para o repositório atual:

git branch -r

41-Como buscar alterações de repositório remoto no Git:

Este comando fará o download das alterações de um repositório remoto, mas não executará uma mesclagem em sua ramificação local (já que o git pull faz isso).

git fetch

42-Como verificar o log de commits atual de um repositório remoto no Git

Commit após commit, o Git cria um log. Você pode descobrir o log do repositório remoto usando este comando:

git log origin/main

43-Como mesclar um repositório remoto com seu repositório local no Git:

Se o repositório remoto tiver alterações que você deseja mesclar com o local, este comando fará isso por você:

git merge origin/main

44-Como obter o conteúdo de ramificações remotas no Git sem mesclar automaticamente:

Isso permite que você atualize o controle remoto sem mesclar nenhum conteúdo nas ramificações locais. Você pode chamar git merge ou git checkout para fazer a mesclagem.

git remote update

45-Como enviar uma nova ramificação para um repositório remoto no Git:

Se você deseja enviar uma ramificação para um repositório remoto, pode usar o comando abaixo. Apenas lembre-se de adicionar -u para criar a ramificação upstream:

git push -u origin branch_name

46-Como remover um branch remoto no Git:

Se você não precisar mais de uma ramificação remota, poderá removê-la usando o comando abaixo:

git push --delete origin branch_name_here

47-Como usar o Git rebase:

Você pode transferir o trabalho concluído de uma ramificação para outra usando git rebase.

git rebase branch_name_here

Git Rebase pode ficar muito confuso se você não fizer isso corretamente. Antes de usar este comando, sugiro que você leia novamente a documentação oficial

<https://git-scm.com/book/it/v2/Git-Branching-Rebasing>

48-Como executar o rebase interativamente no Git:

Você pode executar git rebase interativamente usando o sinalizador -i.

Ele abrirá o editor e apresentará um conjunto de comandos que você pode usar.

git rebase -i master

p, pick = use commit

r, reword = use commit, but edit the commit message

e, edit = use commit, but stop for amending

s, squash = use commit, but meld into previous commit

f, fixup = like "squash", but discard this commit's log message

x, exec = run command (the rest of the line) using shell
d, drop = remove commit

49-Como forçar uma solicitação push no Git:

Este comando forçará uma solicitação push. Isso geralmente é bom para ramificações de solicitação de pull, porque ninguém mais deveria tê-las clonado.

Mas isso não é algo que você deseja fazer com repositórios públicos.

git push -f

50-Como parmeira nao tem mundial

git fax -

51- como agradecer rogerio tavares

git obrigado por ler! A propósito, eu sou o Rogério Tavares.

Você não precisa se lembrar de todos – é por isso que escrevi esta folha de dicas.

Marque esta página para referência futura ou imprima-a, se desejar.