GIT: sistema de controle de versões.

COMANDOS

git status == status

git init == iniciar

git add (nome do arquivo) == começar o monitorar o arquivo

git add . == começar o monitorar todos os arquivos

git commit -am == checkpoint do arquivo == adicionar e mensagem

git log == histórico de modificações

git log --oneline == histórico de modificações resumidas

git log -p == histórico de modificações detalhadas

<https://devhints.io/git-log> == variações do git log

git init --bare == criar servidor para armazenar alterações (PURO)

git remote add “nome do servidor” “endereço” == criar servidor remoto

git remote == mostra os servidores do git

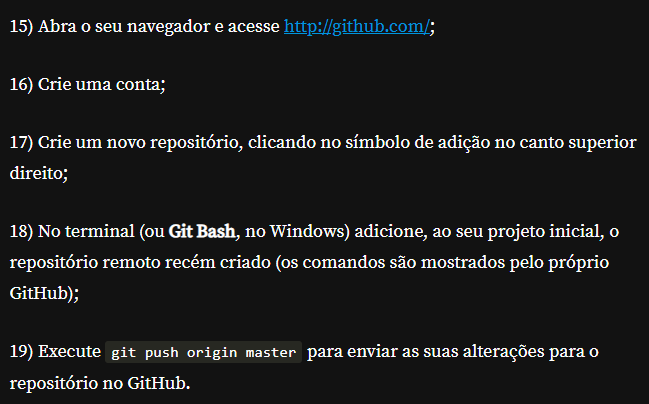
git remote -v == mostra o endereço

git push “servidor” master == empurrar arquivos para o servidor

git remote rename “nome atual” “novo nome” == renomear nome servidor

git pull “servidor” master == puxar arquivos do servidor

repositório remoto:



------------------------------------------------------------------------------------------------

git branch == mostrar branches

git branch “nome do branch” == adicionar novo branch

git checkout “nome do branch” == mudar para outro branch

git checkout -b “nome do branch” == adicionar novo branch e mudar para ele

**A BRANCH master É A BRANCH PADRÃO.**

git merge “nome do branch” == uni os trabalhos criando novo branch

git rebase “nome do branch” == atualizar os trabalhos em uma única linha

git checkout -- “nome do arquivo” == remover alteração de add o arquivo

git reset HEAD “nome do arquivo” == remover alteração antes de fazer um commit

git revert “número do commit” == desfazer um commit

git stash == guardar pra depois

git stash list == lista de salvos para depois

git stash apply “número da stash” == voltar a trabalhar no arquivo

git stash pop == voltar a trabalhar no arquivo e tirar do stash

git checkout “sete primeiros números do commit” == voltar para o commit informado

**criar novo branch para salvar as modificações;**

git diff “número do primeiro commit”..”número do segundo commit” == ver diferenças do arquivo

git diff == (antes de adicionar) == mostra as alterações do arquivo atual

tag no git == ponto não modificável

git tag -a “nomedatag” -m “mensagem da tag” == criar uma tag

git push “nome do servidor” “tag” == mandar modificação para o servidor

DEFINIÇÕES QUE APARECEM NO GIT STATUS

HEAD: estado atual do código

Working tree: Local onde os arquivos estão sendo armazenados e editados

index: Local onde o git armazena o que será commitado

------------------------------------------------------------------------------------------------

**PARA QUE O GIT NÃO MONITORE O ARQUIVO:** crie um arquivo com o nome .gitignore, dentro desse arquivo escreva os arquivos que não deseja que o git monitore e adicione o .gitignore em seu git.

**NUNCA COMMITE UM CÓDIGO QUE NÃO FUNCIONA**

**COMMITAR A CADA ALTERAÇÃO SIGNIFICATIVA**