FOLHA DE DICAS GIT

Instale o Git

GitHub fornece clientes desktop que incluem uma interface gráfica para as ações mais comuns em um repositório e atualiza automaticamente para a linha de comando do Git para cenários avançados.

GitHub para Windows

windows.github.com

GitHub para Mac

mac.github.com

Distribuições do Git para Linux e sistemas POSIX são disponíveis no site oficial do Git SCM.

Git para todas as plataformas

git-scm.com

Configure a ferramenta

Configure informações de usuário para todos os repositórios locais

```
$ git config --global user.name "[nome]"
```
Configura o nome que você quer ligado às suas transações de commit

```
$ git config --global user.email "[endereco-de-email]"
```
Configura o email que você quer ligado às suas transações de commit

Crie repositórios

Inicie um novo repositório ou obtenha de uma URL existente

```
$ git init [nome-do-projeto]
```
Cria um novo repositório local com um nome especificado

```
$ git clone [url]
```
Baixa um projeto e seu histórico de versão inteiro

Faça mudanças

Revise edições e crie uma transação de commit

```
$ git status
```
Lista todos os arquivos novos ou modificados para serem commitados

```
$ git diff
```
Mostra diferenças no arquivo que ainda não foram preparadas

```
$ git add [arquivo]
```
Faz o snapshot de um arquivo na preparação para versionamento

```
$ git diff --staged
```
Mostra a diferença entre arquivos preparados e suas últimas versões

```
$ git reset [arquivo]
```
Retira o arquivo da área de preparação, mas preserva seu conteúdo

```
$ git commit -m "[mensagem descritiva]"
```
Grava o snapshot permanentemente do arquivo no histórico de versão

Mudanças em grupo

Nomeie uma série de commits e combine os esforços completos

```
$ git branch
```
Lista todos os branches locais no repositório atual

```
$ git branch [nome-do-branch]
```
Cria um novo branch

```
$ git switch -c [nome-do-branch]
```
Muda para o branch especificado e atualiza o diretório de trabalho

```
$ git merge [nome-do-branch]
```
Combina o histórico do branch especificado ao branch atual

```
$ git branch -d [nome-do-branch]
```
Exclui o branch especificado

Refatore nomes de arquivos

Mude e remova os arquivos versionados

`$ git rm [arquivo]`
Remove o arquivo do diretório de trabalho e o prepara a remoção

`$ git rm --cached [arquivo]`
Remove o arquivo do controle de versão mas preserva o arquivo localmente

`$ git mv [arquivo-original] [arquivo-renomeado]`
Muda o nome do arquivo e o prepara para o commit

Suprima o monitoramento

Ignore arquivos e diretórios temporários

```
*.log
build/
temp-*
```

Um arquivo de texto chamado `.gitignore` suprime o versionamento acidental de arquivos e diretórios correspondentes aos padrões especificados
`$ git ls-files --others --ignored --exclude-standard`
Lista todos os arquivos ignorados neste projeto

Salve fragmentos

Arquive e restaure mudanças incompletas

`$ git stash`
Armazena temporariamente todos os arquivos monitorados modificados

`$ git stash pop`
Restaura os arquivos recentes em stash

`$ git stash list`
Lista todos os conjuntos de alterações em stash

`$ git stash drop`
Descarta os conjuntos de alterações mais recentes em stash

Revise o histórico

Navegue e inspecione a evolução dos arquivos do projeto

`$ git log`
Lista o histórico de versões para o branch atual

`$ git log --follow [arquivo]`
Lista o histórico de versões para um arquivo, incluindo mudanças de nome

`$ git diff [primerio-branch]...[segundo-branch]`
Mostra a diferença de conteúdo entre dois branches

`$ git show [commit]`
Retorna mudanças de metadata e conteúdo para o commit especificado

Desfaça commits

Apague enganos e crie um histórico substituto

`$ git reset [commit]`
Desfaz todos os commits depois de `[commit]`, preservando mudanças locais
`$ git reset --hard [commit]`
Descarta todo histórico e mudanças para o commit especificado

Sincronize mudanças

Registre um repositório remoto e troque o histórico de versão

`$ git fetch [nome-remoto]`
Baixe todo o histórico de um repositório remoto

`$ git merge [nome-remoto]/[branch]`
Combina o branch remoto ao branch local atual

`$ git push [alias] [branch]`
Envia todos os commits do branch local para o GitHub

`$ git pull`
Baixa o histórico e incorpora as mudanças