



GITHUB FOLHA DE DÍCAS DE GIT

V1.1.1

Git é um sistema de controle de versão distribuído open source que facilita ações com o GitHub em seu notebook ou desktop. Esta folha de dicas resume instruções comumente usadas via linha de comando do Git para referência rápida.

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

<https://windows.github.com>

GitHub para Mac

<https://mac.github.com>

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

Git para todas plataformas

<http://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 "[endereço-de-email]"
```

Configura o email que você quer ligado às suas transações de commit

```
$ git config --global color.ui auto
```

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 específico

```
$ 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 committed

```
$ git diff
```

Mostra diferenças no arquivo que não foram realizadas

```
$ git add [arquivo]
```

Faz o snapshot de um arquivo na preparação para versionamento

```
$ git diff --staged
```

Mostra a diferença entre arquivos selecionados e as suas últimas versões

```
$ git reset [arquivo]
```

Deseleciona o arquivo, 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 checkout [nome-do-branch]
```

Muda para o branch específico e atualiza o diretório de trabalho

```
$ git merge [branch]
```

Combina o histórico do branch específico com o branch atual

```
$ git branch -d [nome-do-branch]
```

Exclui o branch específico



GITHUB FOLHA DE DÍCAS DE GIT

REFATORE NOMES DOS ARQUIVOS

Mude e remova os arquivos versionados

\$ git rm [arquivo]

Remove o arquivo do diretório de trabalho e o seleciona para 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 seleciona para o commit

SUPRIMA O RASTREAMENTO

Exclua 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 --other --ignored --exclude-standard

Lista todos os arquivos ignorados neste projeto

SALVE FRAGMENTOS

Arquive e restaure mudanças incompletas

\$ git stash

Armazena temporariamente todos os arquivos rastreados 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 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 [primeiro-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 marcador de repositório e troque o histórico de versão

\$ git fetch [marcador]

Baixe todo o histórico de um marcador de repositório

\$ git merge [marcador]/[branch]

Combina o marcador do branch no branch local

\$ git push [alias] [branch]

Envia todos os commits do branch local para o GitHub

\$ git pull

Baixa o histórico e incorpora as mudanças

GitHub Training

Aprenda mais sobre o uso do GitHub e do Git. Envie um email para a Equipe de Treinamentos ou visite nosso site para ver a agenda de eventos ou a disponibilidade de cursos particulares.

✉ training@github.com

☞ training.github.com