

Cheat Sheet GITHUB GIT



Git es un sistema de control de revisiones distribuido, rápido y escalable con un conjunto de comandos inusualmente rico que proporciona operaciones de alto nivel y acceso completo a los componentes internos.

CONFIGURAR HERRAMIENTAS

Establecer un nombre de usuario en Git que se adjunta a los commits

```
$ git config --global user.name "[name]"
```

Establecer un nombre de usuario en Git que se adjunta a los commits

```
$ git config --global user.email "[email address]"
```

CREAR REPOSITORIOS

Convierte un directorio existente en un nuevo repositorio de Git dentro de la carpeta en la que está ejecutando este comando.

```
$ git init [project-name]
```

Vincula un repositorio remoto con un repositorio local. La URL apunta a un repositorio GitHub

```
$ git remote add origin [url]
```

Clonar (descargar) un repositorio que ya existe en GitHub, incluidos todos los archivos, ramas y confirmaciones

```
$ git clone [url]
```

HACER CAMBIOS

Enumera todos los archivos nuevos o modificados que se deben confirmar

```
$ git status
```

Toma una instantánea del archivo para preparar la versión. Pasa al Stage Area.

```
$ git add [file]
```

Confirma el estado de un archivo en ese determinado momento

```
$ git commit -m "[descriptive message]"
```

SINCRONIZAR CAMBIOS

Carga todos los commits de la rama local al GitHub

```
$ git push origin [branch]
```

Actualiza los archivos contra la branch actual

```
$ git pull
```

obtenie los cambios, pero no aplicarlos a la branch actual

```
$ git fetch
```

Cheat Sheet GITHUB GIT



Git es un sistema de control de revisiones distribuido, rápido y escalable con un conjunto de comandos inusualmente rico que proporciona operaciones de alto nivel y acceso completo a los componentes internos.

RAMAS

Crear una nueva rama

```
$ git branch [branch-name]
```

Lisar todas las ramas

```
$ git branch
```

Cambia a la rama especificada y actualiza el directorio activo

```
$ git checkout [branch-name]
```

Borra la rama especificada

```
$ git branch -d [branch-name]
```

Une el historial de la rama especificada con la actual

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

REFACTORIZAR ARCHIVOS

Borra el archivo del directorio activo y pone en el área de espera el archivo borrado

```
$ git rm [file]
```

Remueve el directorio y los archivos que contiene

```
$ git rm -r directorio
```

Retira el archivo del control de versiones, pero preserva el archivo a nivel local

```
$ git rm --cached [file]
```

Cambia el nombre del archivo y lo prepara para commit

```
$ git mv [file-original] [file-renamed]
```

VER HISTORIAL

Enumera el historial de la versión para la rama actual

```
$ git log
```

Muestra las diferencias de contenido entre dos ramas

```
$ git diff [first-branch]...[second-branch]
```

Muestra los cambios de contenido y metadatos de la confirmación especificada

```
$ git show [commit]
```

REPOSITORIOS REMOTOS

Lista el repositorio al que estamos vinculados

```
$ git remote -v
```

Renombra un repositorio remoto

```
$ git remote rename origin [nombre_nuevo]
```

Desvincula de un repositorio remoto

```
$ git remote rm [nombre_repo]
```