

# GIT Cheat-sheet

`$ git status`

Enumera todos los archivos nuevos o modificados que se deben confirmar

`$ git diff`

Muestra las diferencias de archivos que no se han enviado aún al área de espera

`$ git add [file]`

Toma una instantánea del archivo para preparar la versión

`$ git diff --staged`

Muestra las diferencias del archivo entre el área de espera y la última versión del

`$ git reset [file]`

Mueve el archivo del área de espera, pero preserva su contenido

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

Registra las instantáneas del archivo permanentemente en el historial de versión

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

Establece el nombre que desea esté anexado a sus transacciones de commit

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

Establece el e-mail que desea esté anexado a sus transacciones de commit

`$ git branch`

Enumera todas las ramas en el repositorio actual

`$ git branch [branch-name]`

Crea una nueva rama

`$ git checkout [branch-name]`

Cambia a la rama especificada y actualiza el directorio activo

`$ git merge [branch]`

Combina el historial de la rama especificada con la rama actual

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

Borra la rama especificada

`$ git init [project-name]`

Crea un nuevo repositorio local con el nombre especificado

`$ git clone [url]`

Descarga un proyecto y toda su historia de versión

`$ git push [alias] [branch]`

Carga todos los commits de la rama local al GitHub

`$ git pull`

Descarga el historial del marcador e incorpora cambios

`$ git init`

Crear un nuevo repositorio

`$ git log`

Enumera el historial de la versión para la rama actual

`$ git log --follow [file]`

Enumera el historial de versión para el archivo, incluidos los cambios de

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

Muestra las diferencias de contenido entre dos ramas

`$ git show [commit]`

Produce metadatos y cambios de contenido del commit especificado

`$ git rm [file]`

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

`$ git rm --cached [file]`

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

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

Cambia el nombre del archivo y lo prepara para commit

`$ git ls-files --other --ignored --exclude-standard`

Enumera todos los archivos ignorados en este proyecto

`$ git reset [commit]`

Deshace todos los commits después de [commit], preservando los cambios localmente

`$ git reset --hard [commit]`

Desecha todo el historial y regresa al commit especificado

`$ git stash`

Almacena temporalmente todos los archivos tracked modificados

`$ git stash pop`

Restaura los archivos guardados más recientemente

`$ git stash list`

Enumera todos los sets de cambios en guardado rápido

`$ git stash drop`

Elimina el set de cambios en guardado rápido más reciente

`$ git fetch [bookmark]`

Descarga todo el historial del marcador del repositorio

`$ git merge [bookmark]/[branch]`

Combina la rama del marcador con la rama local actual