

Git Cheat Sheet

Básico

<code>git init</code> <code><carpeta></code>	Crea un repositorio vacio en <code><carpeta></code> . Sin argumentos permite inicializar la carpeta actual como un repositorio.
<code>git clone</code> <code><repositorio></code>	Clona un <code><repositorio></code> en nuestra maquina local. El <code><repositorio></code> puede ser local o remoto mediante <code>HTTP</code> or <code>SSH</code> .
<code>git config</code> <code>user.name <autor></code>	Define al <code><autor></code> para todas las confirmaciones en el repositorio actual.
<code>git add</code> <code><carpeta archivo></code>	Agrega los cambios realizados en dicha <code><carpeta/archivo></code> para la siguiente confirmación. Acepta expresiones regulares.
<code>git commit -m</code> " <code><mensaje></code> "	Confirma los cambios en esa instancia, pero sin abrir o usar un editor de texto, usa el <code><mensaje></code> como mensaje de confirmación.
<code>git status</code>	Muestra la lista de archivos de los cambios montados, no montados y sin seguimiento.
<code>git log</code>	Muestra el historal de confirmaciones (formato por defecto). Para personalizar ver la sección de opciones adicionales.
<code>git diff</code>	Muestra los cambios no marcados entre el índice y la carpeta de trabajo.

Deshaciendo cambios

<code>git revert</code> <code><confirmacion></code>	Crea una confirmación que deshace todos los cambios realizado en la <code><confirmacion></code> y lo aplica a la rama actual.
<code>git reset</code> <code><archivo></code>	Elimina el <code><archivo></code> del área de montaje, dejando el directorio de trabajo sin cambios por montar, sin sobrescribir ningun cambio.
<code>git clean -n</code>	Muestra los archivos que van a ser eliminados de la carpeta de trabajo. Para eliminar dichos archivos se usa <code>-f</code> en vez de <code>-n</code> .

git config

<code>git config --global</code> <code>user.name <autor></code>	Define el nombre del <code><autor></code> de todas las confirmaciones para el usuario actual.
<code>git config --global</code> <code>user.email <correo></code>	Define el <code><correo></code> del autor de todas las confirmaciones para el usuario actual.
<code>git config --global</code> <code>alias.<alias></code> <code><comando></code>	Crea un atajo de <code><comando></code> en Git que se usara mediante el <code><alias></code> . Ejm. <code>alias.glog log --graph</code> equivalente. <code>glog -> git log --graph</code> .
<code>git config --system</code> <code>core.editor <editor></code>	Define el <code><editor></code> usado por los comandos para todos los usuarios. Por ejemplo <code><editor></code> puede ser <code>vim</code> .
<code>git config --global --edit</code>	Abre el archivo de configuración global para que pueda ser editado manualmente. Usa el editor por defecto de la maquina. Ejm. <code>nano</code> .

git log

<code>git log --<limite></code>	Limita el número de logs usando el <code><limite></code> . Ejemplo: <code>git log -5</code> muestra los últimos 5 logs.
<code>git log --online</code>	Condensa cada una de las confirmaciones a una sola línea.
<code>git log -p</code>	Muestra todos los cambios de cada confirmación.
<code>git log --stat</code>	Incluye que archivos se modificaron y el número relativo de líneas que fueron añadidas o borradas de cada uno de ellos.
<code>git log --author="<autor>"</code>	Busca las confirmaciones por un <code><autor></code> en particular.
<code>git log --grep="<patron>"</code>	Busca confirmaciones cuyo mensaje de confirmación coincida con el <code><patron></code> asignado.
<code>git log <desde>..<code><hasta></code></code>	Muestra las confirmaciones entre <code><desde></code> y <code><hasta></code> . Se puede usar hash de confirmación, rama, <code>HEAD</code> o cualquier punto de revisión.
<code>git log --<archivo></code>	Muestra solo las confirmaciones del <code><archivo></code> especificado.
<code>git log --graph --decorate</code>	El <code>--graph</code> muestra un gráfico en texto al lado izquierdo de los mensajes de confirmación y <code>--decorate</code> el nombre de la rama o tag.

Reescribiendo el historial

<code>git commit --ammend</code>	Reemplaza la última confirmación y/o cambios montados. Usarlo sin cambios montados edita el mensaje del la ultima confirmación.
<code>git rebase</code> <code><base></code>	Elimina el <code><archivo></code> del área de montaje, dejando el directorio de trabajo sin cambios sin sobrescribir ningun cambio.
<code>git reflog</code>	Muestra un log de los cambios en el <code>HEAD</code> del repositorio local. Agrega <code>--relative-date</code> para mostrar la fecha o <code>--all</code> para todos.
<code>git branch</code> <code><rama></code>	Crea una nueva <code><rama></code> . Sin argumentos muestra la lista de ramas del repositorio.
<code>git checkout -b</code> <code><rama></code>	Crea una nueva <code><rama></code> en base a la rama actual. Si quitamos <code>-b</code> permite posicionarse en la <code><rama></code> .
<code>git merge</code> <code><rama></code>	Mezcla la <code><rama></code> con la rama actual.

Repositorios remotos

<code>git remote add</code> <code><nombre> <enlace></code>	Crea una nueva conexión a un repositorio remoto. El <code><nombre></code> es el alias del <code><enlace></code> al cual se comunica.
<code>git fetch</code> <code><remoto> <rama></code>	Obtiene la <code><rama></code> del repositorio <code><remoto></code> . Sino se especifica la rama trae toda todas las ramas.
<code>git pull</code> <code><remoto></code>	Obtiene una copia del repositorio <code><remoto></code> y la mezcla al instante con el repositorio local.
<code>git push</code> <code><remoto> <rama></code>	Envia la <code><rama></code> al repositorio <code><remoto></code> actualizandola si existe y sino creandola.

git diff

<code>git diff HEAD</code>	Muestra diferencias de la ult. confirmación y la carpeta trabajo.
<code>git diff --cached</code>	Muestra diferencias de la ult. confirmación y los cambios marcados.

git reset

<code>git reset</code>	Reinicia el área de marcado para coincidir con la última confirmación, pero deja la carpeta de trabajo sin cambios.
<code>git reset --hard</code>	Reinicia el área de marcado y carpeta de trabajo para coincidir con la ultima confirmación, sobrescribiendo todos los cambios.
<code>git reset <confirmacion></code>	Mueve la rama actual hacia la <code><confirmacion></code> reiniciando el área de marcado, pero deja la carpeta de trabajo.
<code>git reset --hard <confirmacion></code>	Lo mismo que <code>git reset --hard</code> pero todos los cambios no confirmados y todas las confirmaciones despues de la <code><confirmacion></code> .

git rebase

<code>git rebase -i</code> <code><base></code>	Un rebase interactivo que abre un editor para ingresar los comandos para cada confirmación que sera transferida a la nueva <code><base></code> .
---	--

git pull

<code>git pull --rebase</code> <code><remoto></code>	Obtiene una copia de <code><remoto></code> de la rama actual pero en vez de realizar una mezcla realiza un rebase de las ramas.
---	---

git push

<code>git push --force</code> <code><remoto></code>	Fuerza a Git a enviar la rama al <code><remoto></code> ignorando si hay cambios e incompatibilides. Solo usar cuando estas seguro de lo que haces.
<code>git push --all</code> <code><remoto></code>	Envia todas las ramas del repositorio local al repositorio <code><remoto></code> .
<code>git push --tags</code> <code><remoto></code>	Las etiquetas no son enviadas automaticamente por lo que es necesario usar <code>--tags</code> para enviarlas al repositorio <code><remoto></code> .