

## COMANDOS MÁS UTILIZADOS HASTA EL MOMENTO

COMANDOS	EXPLICACION DE SINTAXIS
NAVEGACIÓN E INTERACCIONES CON FICHEROS (DE TIPO INFORMATIVO)	
pwd	Mostrar el nombre de la carpeta en el que uno se encuentra situado.
cd	Entrar en la carpeta.
ls	Ver lo que hay dentro de la carpeta.
NAVEGACIÓN E INTERACCIONES CON FICHEROS (CREAR O ELIMINAR ARCHIVOS)	
mkdir "Nombre de la carpeta"	Crear una carpeta.
rmdir	Borrar una carpeta vacía.
gedit	Crear y editar archivos de texto.
rm nombre	Borrar archivo.
NAVEGACIÓN E INTERACCIONES CON FICHEROS (COPIAR ARCHIVOS Y DIRECTORIOS)	
cp	Copiar un archivo o carpeta en el directorio especificado.
cp -r	Copiar carpetas.
lmv	Mover un archivo o carpeta a un archivo o carpeta.
OTROS COMANDOS	
ls -t	Ordena los archivos y/o carpetas según su última modificación.
ls -a	Muestra si hay directorios ocultos.
ls -r	Muestra el contenido de todas las carpetas.
rm -r nombre	Borrar directorio y todos los archivos que contiene.
mv "Carpeta nueva"	Renombrar carpeta.
cd ..	Salir de la carpeta.
ctrl+l o clear	Limpiar la consola

COMANDOS	EXPLICACION DE SINTAXIS
COMANDOS PARA CREAR UN REPOSITORIO LOCAL	
git init	Crea el repositorio local.
git config user.name "nombre de usuario"	Agrega nuestra identidad.
git config user.email "nombre de usuario"	Agrega nuestro email.
COMANDOS PARA SUBIR ARCHIVOS	
git add .	Agrega todos los archivos.
git commit -m "mensaje"	Comitea los cambios hechos.
git push origin main	Envía cambios al repositorio remoto.
git status	Seguimiento del estado de los archivos.
COMANDOS PARA BAJAR ARCHIVOS	
git clone	Clonar un repositorio remoto existente.
git remove -v	Indica si estamos conectados con el repositorio remoto.
git pull origin main	Descargar los cambios que existen en el repositorio remoto.
OTROS COMANDOS	
git branch	Muestra la cantidad de ramas que tengo.
git branch "nombre de la rama"	Crear una rama.
git branch -M "nombre de la nueva rama"	Renombrar la rama.
git checkout "nombre de la rama"	Moverse entre las ramas.
git remote add origin	Enlaza el repositorio local con un repositorio remoto.
git log	Muestra los commit que he realizado.
git log --online	Resumir commits.

# Git Cheat Sheet

by Jan Krüger <jk@jk.gs>, <http://jan-krueger.net/git/>  
Based on work by Zack Rusin

## Basics

Use `git help [command]` if you're stuck.

master	default devel branch
origin	default upstream branch
HEAD	current branch
HEAD^	parent of HEAD
HEAD~4	great-great grandparent of HEAD
foo..bar	from branch foo to branch bar

## Create

### From existing files

```
git init
git add .
```

### From existing repository

```
git clone ~/old ~/new
git clone git://...
git clone ssh://...
```

## View

```
git status
git diff [oldid newid]
git log [-p] [file|dir]
git blame file
git show id (meta data + diff)
git show id:file
git branch (shows list, * = current)
git tag -l (shows list)
```

## Revert

In Git, revert usually describes a new commit that undoes previous commits.

```
git reset --hard (NO UNDO)
(reset to last commit)
git revert branch
git commit -a --amend
(replaces prev. commit)
git checkout id file
```

## create

```
init
clone
```

## browse

```
status
log
blame
show
diff
```

## change

mark changes  
to be respected  
by commit:

```
add
```

## revert

```
reset
checkout
revert
```

## update

```
pull
fetch
merge
am
```

## branch

```
checkout
branch
```

## commit

```
commit
```

## push

```
push
format-patch
```

(left to right) **Command Flow**

## Publish

In Git, commit only respects changes that have been marked explicitly with add.

```
git commit [-a]
(-a: add changed files
automatically)
git format-patch origin
(create set of diffs)
git push remote
(push to origin or remote)
git tag foo
(mark current version)
```

## Useful Tools

```
git archive
Create release tarball
git bisect
Binary search for defects
git cherry-pick
Take single commit from elsewhere
git fsck
Check tree
git gc
Compress metadata (performance)
git rebase
Forward-port local changes to
remote branch
git remote add URL
Register a new remote repository
for this tree
git stash
Temporarily set aside changes
git tag
(there's more to it)
gitk
Tk GUI for Git
```

## Update

```
git fetch (from def. upstream)
git fetch remote
git pull (= fetch & merge)
git am -3 patch.mbox
git apply patch.diff
```

## Branch

```
git checkout branch
(switch working dir to branch)
git merge branch
(merge into current)
git branch branch
(branch current)
git checkout -b new other
(branch new from other and
switch to it)
```

## Conflicts

Use add to mark files as resolved.

```
git diff [--base]
git diff --ours
git diff --theirs
git log --merge
gitk --merge
```

## Tracking Files

```
git add files
git mv old new
git rm files
git rm --cached files
(stop tracking but keep files in working dir)
```

## Structure Overview

