GIT CHEATSHEET

Git es un sistema de control de versiones distribuido muy usado en el mundo IT. Permite una forma de trabajar colaborativa donde puedes ver el histórico de un proyecto donde participan varios miembros de un equipo. A través de git podemos trabajar en local (GIT BASH) para luego volver a subir los cambios en remoto (GITHUB).) para luego volver a subir los cambios en remoto (GITHUB). Su prompt es $,

---

ANTES DE EMPEZAR A USAR GIT

INSTALACIÓN

https://git-scm.com/downloads

CONFIGURACIÓN

$ git config -global user.name "..."
Establecer nombre para realizar las transacciones de confirmación

$ git config -global user.email "..."
Establecer email para realizar las transacciones de confirmación

$ git config --global --edit
Comprobar la configuración de las transacciones de confirmación

COMANDOS TUTORIALES

$ git help
Ver principales uso en la misma consola

$ git help tutorial
Ver un tutorial en html (página web)

$ git help + comando específico
Mostrar tutorial sobre ese comando en html (página web)

PRIMEROS PASOS EN EL TERMINAL

COMANDOS MOVERSE POR EL TERMINAL

$ cd
Moverse a la carpeta origen del terminal de git

$ cd ..
Moverse al sitio anterior al que estamos

$ cd nombrecarpeta
Moverse a la carpeta que le hemos dicho que se encuentra dentro de dicha ruta

COMANDOS OBSERVACIÓN

$ ls
Ver los archivos que hay en la ruta actual

$ pwd
Ver la ruta del repositorio donde estoy ubicado ahora

OTROS COMANDOS ÚTILES

$ clear
Limpiar la pantalla del terminal

$ git .
Abrir visual studio code (si esta instalado)

$ Exit
Cerrar el terminal de git

COMANDOS CARPETAS Y FICHEROS

CREACIÓN

$ mkdir nombrecarpeta
Para crear una nueva carpeta

$ git init nombrecarpeta
Para crear un nuevo repositorio (con fichero .git)

$ git init .
Crear un fichero git

$ touch nombre fichero.extensión
Crear un fichero de dicha extensión

$ notepad fichero.txt
Crear un fichero txt y abrir el editor de texto (bloc de notas)

$ echo "escrito" > archivo.extensión
Añadir texto en el archivo y crear el archivo si este no existía

MODIFICACIÓN

$ git mv "archivoactual.extensión" "nuevonombre.extensión"
Cambiar nombre o extensión de un archivo/carpeta

$ echo -e "texto\n texto2\n..." > archivo.extensión
Sobrescribir texto con saltos de línea

$ echo -e "texto adicional\n texto2..." >> archivo.extensión
Añadir texto en el archivo que había

$ sed -i 's/(condiciones) /' archivo.ext
Modificar el texto de un archivo (sin -i vemos como queda en el propio terminal pero no se modifica)

PRINCIPALES COMANDOS

ACTUALIZAR CAMBIOS

$ git status
Ver si hay cambios o commits pendientes de hacer

$ git add .
Trakear los cambios pendientes de todos los archivos

$ git add "nombrearchivo.extemsión"
Trakear los cambios pendientes de dicho archivo

$ git add "*.extensión"
Trakear los cambios pendientes de archivos con esa extensión

COMMITS

$ git commit -m "comentario"
Añadir un commit después de una acción. Sirve de marcador.

$ git commit --amend -m "Comentario nuevo"
Modificar el comentario del último commit

$ git log
Para ver todos los commits realizados en un repositorio

$ git show
Para ver en detalle las modificaciones que se han hecho

$ git show idcommit
Para ver en detalle las modificaciones de un commit en concreto

BRANCAS

$ git branch "nombre nueva rama"
Crear una nueva rama (clonara la branca en la que estábamos)

$ git checkout Nombrerama
Cambiar de rama a la citada

$ git checkout -b "Nombre rama"
Crear una nueva rama y situarse en ella

$ git branch
Ver todas las ramas creadas

$ git diff nombre_otra_branch
Ver las diferencias de la branca actual con otra

$ git merge nombre_branca_fuente
Aplicar las diferencias de la branca fuente a la actual

COMANDOS DE ELIMINACIÓN

FICHEROS Y BRANCAS

$ git rm -f "archivo.extensión"
Fuerza la eliminación de un fichero

$ git branch -d nombre_rama
Eliminar una branca en local (No puede ser la branca activa!)

$ git push origin --delete nombre_rama
Elimina una branca en remoto (hay que estar conectado)

GIT BASH CON GITHUB

CLONAR

$ git clone dirección_repositorio_github
Descargar un repositorio en la carpeta actual en git bash

$ git clone --branch nombre_rama dirección_repositorio_github
Descargar una branca específica de un repositorio

VINCULAR - DESVINCULAR

$ git remote -v
Comprobar que repositorios están vinculados en local y remoto. La primera palabra que sale es el bookmark

$ git remote add bookmark dirección_repositorio_github
Vincular git local con el repositorio de github en remoto

$ git branch -vv
Muestra información detallada de las ramas locales (git) vinculadas con sus ramas en remoto (github)

$ git remote remove bookmark
Desvincula git local con el repositorio en remoto (github)

SINCRONIZACIÓN DE CAMBIOS (estando vinculados)

$ git fetch
Descargar el historial en remoto al git local

$ git diff bookmark/main..main
Compara diferencias entre las ramas principales de remoto y local

$ git merge bookmark/main
Fusionar los cambios de la rama remoto a la local (estando misma branca)

$ git push bookmark nombre_rama
Subir los cambios realizados en dicha rama al remoto. Crea un pull request en Github

$ git pull
Obtener los cambios de una rama remota correspondiente y fusiona automáticamente en la rama local actual

$ git pull nombre_repositorio branca_remoto
Fusionar la branca actual del git local con la branca en remoto correspondiente

ARCHIVOS CON FUNCIONES

$ source fichero.ipynb
Subir al git un fichero

$ nombre_función_fichero "$1" "$n"
Llamar a la función del fichero para ejecutar-la. Sustituir $ por el nombre real en esta ocasión

$ cat fichero.extensión
Ver lo que tiene el fichero en el terminal

$ less fichero.ext
Abrir el fichero en una vista página por página (q para salir)

PASO A PASO

1. CREAR REPOSITORIO EN GITHUB CON README
2. SUBIMOS FICHEROS ORIGINALES AL GITHUB
3. CREAMOS UNA RAMA A PARTIR DE MAIN DE VERSIÓN PARA CLONAR
4. CLONAR EL REPOSITORIO A PARTIR DE LA RAMA QUE HEMOS CREADO
5. CREAR UNA RAMA EN LOCAL CON MI NICKNAME (PARA SABER QUE HE HECHO YO ESOS CAMBIOS)
6. TRABAJO EN LOCAL... HAGO CAMBIOS, ADD, COMMIT... Y PUSH
7. VOY AL GITHUB Y COMPARO EL PULL REQUEST, SI TODO CORRECTO HAGO EL MERGE!
8. ELIMINO LA RAMA CON MI NICK NAME EN EL GITHUB (Y LOCAL SI QUEREMOS)

WWW.GITHUB.COM