

Jacobo's CheatSheet

Bash, Git & GitHub

Terminal Bash

| Mostrar | |
|------------------------------|--|
| \$ ls | Muestra lista de archivos de la ubicación |
| \$ ls -a | Lista archivos incluyendo ocultos |
| \$ ls -l | Muestra info de una carpeta |
| \$ ls -R | Muestra archivos y subcarpetas a partir de ubicación |
| \$ pwd | Muestra la ubicación |
| more <nombre del archivo> | Muestra contenido de archivo |

| Crear | |
|---------------------------------|--|
| \$ mkdir <nombre de carpeta> | Crea nueva carpeta en la ubicación |
| \$ touch <nombre de archivo> | Crea nuevo(s) archivo(s) en la ubicación |
| nano | Crea el archivo y tambien ingresa a este archivo (^ significa tecla Ctrl) Write out => guardar (se puede modificar la extencion) |

| Eliminar | |
|------------------------------|---|
| \$ rm <nombre de archivo> | Elimina archivo mencionado |
| \$ rmdir <nombre carpeta> | Elimina carpeta vacia mencionada |
| \$ rm -r <nombre carpeta> | Elimina carpeta mencionada con su contenido |
| \$ rm -rf | La F elimina de forma forzada |

| Mover Archivos o carpetas | |
|---|--|
| mv <archivo/carpeta> <carpeta destino> | mover un archivo o carpeta al destino adelante o atrás (..) |
| mv * <carpeta destino> | Mueve todos los archivos a la carpeta-destino mencionada |
| mv *<extension> <carpeta destino> | mueve todos los archivos de una misma extension (.txt .png .js etc.) a la carpeta-destino mencionada |

| Renombrar | |
|--|---|
| \$ mv <archivo/carpeta> <nombre nuevo> | Cambia a un archivo o carpeta el nombre por uno que no exista en la ubicación |

| Copiar | |
|--|--|
| \$ cp <archivo/carpeta> <carpeta destino> | Copia un archivo o carpeta |
| opcion: -r | Se añade al comando anterior para indicar que copie contenido de subcarpetas |

| Navegacion | |
|-----------------------|--|
| \$ cd | Directorio raiz |
| \$ cd <carpeta> | Se ubica adelante en carpeta mencionada |
| \$ cd .. | se devuelve a la ubicación anterior |
| \$ cd ../<carpeta> | Se devuelve a ubicación anterior e ingresa a la carpeta mencionada |

| Otros comandos | |
|---|--|
| \$ clear | Limpia pantalla de terminal |
| --help | Al escribirlo frente a comando muestra ayuda de este |
| \$ find . -iname <archivo/carpeta> | Busca el archivo en mension desde la ubicación actual pasando por subcarpetas |
| \$ grep -i --color "palabra" <archivo> | i (que no respete mayuscula o minuscula) color (cambia el color en terminal) "palabra" (determinar que palabra se cambiara color) Archivo (nombre del archivo a tratar) |

| Atajos teclado | |
|----------------|--|
| Ctrl c | Finaliza proceso que esta ejecutando terminal |
| Ctrl L | Limpia pantalla de terminal |
| Ctrl +/- | amplia/reduce Visualizacion pantalla de terminal |

| Caracteres especiales | |
|-----------------------|---|
| " " , ' (comillas) | Se usan para encerrar palabras con espacios, ej: "Archivos Digital House" |
| . (punto) | Hace referencia al directorio donde estamos ubicados en ese momento |

Jacobo's CheatSheet

Bash, Git & GitHub

GIT

| Iniciar GIT | |
|----------------------|--|
| \$ git init | Crea repositorio local |
| \$ git status | Muestra el estado de los archivos en la ubicación actual |

| Seguimiento o Tracked | |
|---|--|
| \$ git add <archivo> | Sube el archivo mencionado a seguimiento |
| \$ git add . | Sube a seguimiento todo lo que contenga la carpeta donde se esta ubicado |

| Commit o Rama principal | |
|---|---|
| \$ git commit -m "mensaje" | Hace commit de los archivos que esten en seguimiento, se debe generar con un mensaje alusivo a los cambios |
| \$ git commit --amend | Editor de texto de VIM i (Se escribe antes de introducir texto a aclarar) para menu de salida presionar esc :wq! (Salir de VIM) |
| git log | Muestra estado detallado del commit |

Jacobo's CheatSheet

Bash, Git & GitHub

GitHub

| Conexión | |
|--|--|
| \$ git config user.name "UsuarioGitHub" | Agrega nuestra identidad |
| \$ git config user.name | Corroborar nombre de usuario GitHub consignado |
| \$ git config user.email "EmailGitHub" | Agrega nuestro e-mail |
| \$ git config user.email | Corroborar Correo electrónico GitHub consignado |
| \$ git config --global user.name "UsuarioGitHub" | Deja configurado el usuario de GitHub de forma predeterminada en el pc |
| \$ git config --global user.email "EmailGitHub" | Deja configurado el e-mail de GitHub de forma predeterminada en el pc |
| git remote -v | Verifica vínculo con repo |

| Acción entre Repos | |
|------------------------|--|
| git pull origin master | Descarga cambios que existen en el repo remoto |
| git push origin master | Envía cambios ya commiteados a repo remoto |

| Vincular Repos | |
|-----------------------------------|---|
| git remote add origin LinkDelRepo | Conecta el repositorio local con el repositorio remoto propio |
| git clone LinkDelRepo | Clona y vincula repositorio en pc sea propio o colaborativo |
| rm -rf .git | Quita enlace de repo github |

| Ramas o Branchs | |
|----------------------------|---|
| git branch nombreRama | Crea una rama con el nombre asignado |
| git branch | Muestra rama creada |
| git checkout nombreRama | Dirige a la rama deseada |
| git branch -d nombreRama | Borrar branch específica |
| git push origin nombreRama | Envía cambios ya commiteados a repo remoto en rama determinada |
| git merge nombreRama | Fusiona rama custom con rama principal (se debe aplicar desde la rama principal nombrando a la rama custom) |
| git branch -M newName | Renombra rama |