escuela ADMIN

cd

change directory

El comando cd se usa para cambiar del directorio en el que estamos posicionados a otro.
Supongamos que estamos en el directorio ~ que simboliza a /home/usuario. Necesitamos ir a /home/usuario/Descargas, para ver si efectivamente tenemos el archivo firefox-42.tar.gz

```
usuario@maquina:~$
usuario@maquina:~$ cd Descargas
usuario@maquina:~ /Descargas$
```

El prompt nos indica que ya estamos en el directorio Descargas. Vemos que después de los dos puntos hay un ñuflo y a su lado, un simbolo $ porque el prompt nos avisa que nuestro usuario no es root. En ese caso el prompt mostrará un signo #
Si deseamos volver al directorio ~ lo que tipeamos es:

```
usuario@maquina:~ /Descargas$ cd ..
usuario@maquina:~$
```

Esto es, el comando cd seguido de un espacio y dos puntos.

escuela ADMIN

mkdir

make directory

El comando mkdir se usa para crear directorios. La sintaxis mas común es

```
usuario@maquina:~$ mkdir fotos_2015
```

lo cual crea la carpeta "fotos_2015" en /home/usuario.

Se puede hacer una serie de carpetas anidadas, anteponiendo el parámetro -p antes del nombre:

```
usuario@maquina:~$ mkdir -p fotos/fotos_2015/fotos_enero
```

Si usamos el parámetro -v nos da una confirmación de haber creado la carpeta exitosamente:

```
usuario@maquina:~$ mkdir -v fotos_2015
usuario@maquina:~$ mkdir: created directory 'fotos_2015'
```

escuela ADMIN

present working directory

pwd

Puede suceder que nos encontremos con la situación que el prompt de bash (o de algún otro interprete de comandos) esté formulado de tal manera que no nos muestra ni nuestro nombre de usuario ni el directorio donde estamos ubicados. Para esa razón existe el comando pwd, el cual nos mostrará la ruta de nuestra ubicación. Supongamos que tenemos un prompt asi:
`[$]:

Nuestra alternativa es tipear

```
[$]: pwd
[$]: /home/usuario/lista/original
```

escuela ADMIN

list files and directories

ls

Cuando estamos dentro de un directorio y precisamos saber su contenido, mas puntualmente cuales son los archivos y subdirectorios que contiene, contamos con el comando ls.
En su sintaxis regular, nos va a mostrar las carpetas y archivos en columnas, dependiendo de los nombres que tengan.

```
usuario@maquina:~$ ls
usuario@maquina:~$ fotos_2015 fotos_2014 pelicula.mkv
```

Los nombres de las carpetas aparecerán coloreados, y los nombres de archivo también, aunque con distintos colores, según sus permisos. Si usamos el parámetro -l nos muestra una lista detallada y si usamos -la nos agrega los archivos ocultos.

```
usuario@maquina:~$ ls -la
total 608
drwxr-xr-x 15 usuario usuarios   4096 jun 30 18:01 fotos_2015
drwxr-xr-x 95 usuario usuarios   4096 ene 30 22:18 fotos_2014
-rw-r--r-- 1  usuario usuarios 107858 jun 30 12.34 pelicula.mkv
-rw-r--r-- 1  usuario usuarios    234 ago 11 19.44 .oculto
```

escuela ADMIN

rm

remove files

Borrar archivos es algo rutinario, por eso el comando rm tiene algunos parámetros interesantes. Por ejemplo, si le tipeamos el parámetro -r usa la recursividad para borrar todos los archivos de esa carpeta. Debemos usar esta opción con cuidado, porque también borra los directorios.
Con -f usa el forzado, para borrar incluso si los archivos tienen acceso de solo lectura.
Con -i aparece el modo interactivo, que pregunta si queremos borrar el archivo.
Es conocida la combinación de rm con los parámetros -rf, que borran recursivamente una serie de directorios recursivamente, es algo que hay que usar con responsabilidad porque si lo usamos en la raíz del disco con permisos de superusuario, el resultado va a ser que todo el disco se borrará.

escuela ADMIN

touch

Este comando es usado para crear archivos vacíos. Como "vacíos" entendemos a los archivos que tienen o bytes, pero tienen datos como fecha y hora de creación. La operación de crear con touch un archivo nos ahorra el tener que abrir un programa y guardar sin contenido al archivo, para cerrar el programa.
Con el parámetro -m modificamos la hora y dia de la creación.
Con -t le cambiamos el momento de la creación a una fecha y hora especificada:

```
usuario@maquina:~$ touch -t 201510101525.15 prueba
```

Donde `201510101525.15` está en formato `YYMMDDHHMM.SS`
También podemos hacer muchos archivos vacíos en cadena:

```
usuario@maquina:~$ touch -c prueba archivo carpeta config
```

escuela ADMIN

concatenate cat

Este comando es particular, porque tiene algunos usos que son un poco dispares. Principalmente se lo usa para obtener la impresión en pantalla del contenido de un archivo de texto, ya sea un log, un archivo de configuración o código fuente.
Si lo usamos asi:

```
usuario@maquina:~$ cat README
README
El archivo README contiene información acerca de este
proyecto. Se incluye en formato tar.gz
```

También tiene la facultad de unir archivos consecutivamente, usando el operador dump.

```
usuario@maquina:~$ cat texto1.txt texto2.txt > textazo.txt
```

Puede escribir a un archivo lo que le tipeamos en pantalla. Para salir al prompt usamos CTRL+D:

```
usuario@maquina:~$ cat >escribimos.txt
```