

5

Trabajando con comandos.

Up to this point, we have seen a series of mysterious commands, each with its own options and arguments. In this chapter, we will attempt to remove some of that mystery and even create our own commands. The commands introduced in this chapter are as follows:

Hasta este punto, hemos visto una serie de comandos misteriosos, cada uno con sus propias opciones y argumentos. En este capítulo, intentaremos eliminar parte de ese misterio e incluso crear nuestros propios comandos. Los comandos presentados en este capítulo son los siguientes:

`type` -- Indicate how a command name is interpreted

`type` -- indica cómo se interpreta el nombre de un comando

`which` -- Display which executable program will be executed

`which` -- muestra qué programa ejecutable se ejecutará

`help` -- Get help for shell builtins

`help` -- obtén ayuda para las funciones integradas de shell

`man` -- Display a command's manual page

`man` -- muestra la página del manual de un comando

`apropos` -- Display a list of appropriate commands

`apropos` -- muestra una lista de comandos apropiados

`info` -- Display a command's info entry

`info` -- muestra la entrada de información de un comando

`whatis` -- Display one-line manual page descriptions

`quéés` -- muestra descripciones de páginas de manual de una línea

`alias` -- Create an alias for a command

`alias` -- crea un alias para un comando

What Exactly Are Commands?

¿Qué son exactamente los comandos?

A command can be one of four different things.

Un comando puede ser una de cuatro cosas diferentes.

- **An executable program** like all those files we saw in `/usr/bin`. Within this category, programs can be compiled binaries such as programs written in C and C++, or programs written in scripting languages

such as the shell, Perl, Python, Ruby, and so on.

Un programa ejecutable como todos esos archivos que vimos en `/usr/bin`. Dentro de esta categoría, los programas pueden ser binarios compilados, como programas escritos en C y C ++, o programas escritos en lenguajes de scripting como shell, Perl, Python, Ruby, etc.

- **A command built into the shell itself.** bash supports a number of commands internally called shell builtins. The `cd` command, for example, is a shell builtin.

Un comando integrado en el propio shell. bash admite una serie de comandos llamados internos del shell. El comando `cd`, por ejemplo, es un shell incorporado.

- **A shell function.** Shell functions are miniature shell scripts incorporated into the environment. We will cover configuring the environment and writing shell functions in later chapters, but for now, just be aware that they exist.

Una función de shell. Las funciones de shell son scripts de shell en miniatura incorporados al entorno. Cubriremos la configuración del entorno y la escritura de funciones de shell en capítulos posteriores, pero por ahora, tenga en cuenta que existen.

- **An alias.** Aliases are commands that we can define ourselves, built from other commands.

Un alias. Los alias son comandos que podemos definir nosotros mismos, contruidos a partir de otros comandos.

Identifying Commands

Identificación de comandos

It is often useful to know exactly which of the four kinds of commands is being used, and Linux provides a couple of ways to find out.

A menudo es útil saber exactamente cuál de los cuatro tipos de comandos se está utilizando, y Linux proporciona un par de formas de averiguarlo.

type -- Display a Command's Type

type -- muestra el tipo de un comando

The `type` command is a shell builtin that displays the kind of command the shell will execute, given a particular command name. It works like this:

El comando `type` es un intérprete de comandos que muestra el tipo de comando que ejecutará el shell, dado un nombre de comando en particular. Funciona así:

```
type command
```

where `command` is the name of the command you want to examine. Here are some examples:

donde `comando` es el nombre del comando que desea examinar. Aquí hay unos ejemplos:

```
[me@linuxbox ~]$ type type
type is a shell builtin
[me@linuxbox ~]$ type ls
ls is aliased to `ls --color=tty`
```

```
[me@linuxbox ~]$ type cp
cp is /bin/cp
```

Here we see the results for three different commands. Notice the one for `ls` (taken from a Fedora system) and how the `ls` command is actually an alias for the `ls` command with the `--color=tty` option added. Now we know why the output from `ls` is displayed in color!

Aquí vemos los resultados de tres comandos diferentes. Observe el de `ls` (tomado de un sistema Fedora) y cómo el comando `ls` es en realidad un alias para el comando `ls` con la opción `--color = tty` agregada. ¡Ahora sabemos por qué la salida de `ls` se muestra en color!

which -- Display an Executable's Location que: muestra la ubicación de un ejecutable

Sometimes there is more than one version of an executable program installed on a system. While this is not common on desktop systems, it's not unusual on larger servers. To determine the exact location of a given executable, the `which` command is used.

A veces, hay más de una versión de un programa ejecutable instalada en un sistema. Si bien esto no es común en los sistemas de escritorio, no es inusual en servidores más grandes. Para determinar la ubicación exacta de un ejecutable dado, el comando que se usa

```
[me@linuxbox ~]$ which ls
/bin/ls
```

`which` works only for executable programs, not builtins or aliases that are substitutes for actual executable programs. When we try to use `which` on a shell builtin—for example, `cd`—we either get no response or get an error message.

que funciona solo para programas ejecutables, no incorporados o alias que son sustitutos de programas ejecutables reales. Cuando tratamos de usar qué intérprete de órdenes de un shell, por ejemplo, `cd`, no obtenemos respuesta o recibimos un mensaje de error.

```
[me@linuxbox ~]$ which cd
/usr/bin/which: no cd in
(/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games)
```

This result is a fancy way of saying "command not found."

Este resultado es una forma elegante de decir "comando no encontrado".

Getting a Command's Documentation Obtener la documentación de un comando

With this knowledge of what a command is, we can now search for the documentation available for each kind of command.

Con este conocimiento de lo que es un comando, ahora podemos buscar la documentación disponible para cada tipo de comando.

help - Get Help for Shell Builtins

help - Obtenga ayuda para los componentes integrados de Shell

bash has a built-in help facility available for each of the shell builtins. To use it, type help followed by the name of the shell builtin. Here's an example:

bash tiene una función de ayuda integrada disponible para cada una de las funciones integradas de shell. Para usarlo, escriba help seguido del nombre del shell incorporado. Aquí tienes un ejemplo:

```
hernani@EliteDesk:~$ help cd
```

```
cd: cd [-L|[-P [-e]]] [dir]
```

Modifica el directorio de trabajo del shell.

Modifica el directorio actual a DIR. DIR por defecto es el valor de la variable de shell HOME.

La variable CDPATH define la ruta de búsqueda para el directorio que contiene DIR. Los nombres alternativos de directorio en CDPATH se separan con dos puntos (:). Un nombre de directorio nulo es igual que el directorio actual. Si DIR comienza con una barra inclinada (/), entonces no se usa CDPATH.

Si no se encuentra el directorio, y la opción del shell `cdable_vars` está activa, entonces se trata la palabra como un nombre de variable. Si esa variable tiene un valor, se utiliza su valor para DIR.

Opciones:

-L fuerza a seguir los enlaces simbólicos: resuelve los enlaces simbólicos en DIR después de procesar las instancias de `..`

-P usa la estructura física de directorios sin seguir los enlaces simbólicos: resuelve los enlaces simbólicos en DIR antes de procesar las instancias de `..`

-e si se da la opción -P y el directorio actual de trabajo no se puede determinar con éxito, termina con un estado diferente de cero.

La acción por defecto es seguir los enlaces simbólicos, como si se especificara `-L`.

`..` se procesa quitando la componente del nombre de la ruta inmediatamente anterior hasta una barra inclinada o el comienzo de DIR.

Estado de Salida:

Devuelve 0 si se cambia el directorio, y si \$PWD está definido como correcto cuando se emplee -P; de otra forma es diferente a cero.

A note on notation: When square brackets appear in the description of a command's syntax, they indicate optional items. A vertical bar character indicates mutually exclusive items. In the case of the `cd` command above:

Una nota sobre la notación: Cuando aparecen corchetes en la descripción de la sintaxis de un comando, indican elementos opcionales. Un carácter de barra vertical indica elementos mutuamente excluyentes. En el caso del comando `cd` anterior:

```
cd [-L|[-P[-e]]] [dir]
```

This notation says that the command `cd` may be followed optionally by either a `-L` or a `-P` and further, if the `-P` option is specified the `-e` option may also be included followed by the optional argument `dir`.

Esta notación dice que el comando `cd` puede ir seguido opcionalmente por una `-L` o una `-P` y además, si se especifica la opción `-P`, la opción `-e` también puede incluirse seguida del argumento opcional `dir`.

While the output of help for the `cd` commands is concise and accurate, it is by no means tutorial and as we can see, it also seems to mention a lot of things we haven't talked about yet! Don't worry. We'll get there.

Si bien el resultado de la ayuda para los comandos `cd` es conciso y preciso, de ninguna manera es un tutorial y, como podemos ver, ¡también parece mencionar muchas cosas de las que no hemos hablado todavía! No se preocupe. Vamos a llegar.

--help -- Display Usage Information

--help - Mostrar información de uso

Many executable programs support a `--help` option that displays a description of the command's supported syntax and options. For example:

Muchos programas ejecutables admiten una opción `--help` que muestra una descripción de la sintaxis y las opciones admitidas por el comando. Por ejemplo:

```
[me@linuxbox ~]$ mkdir --help
```

Modo de empleo: `mkdir [OPCIÓN]... DIRECTORIO...`

Crea los `DIRECTORIO(s)`, si no existen ya.

Los argumentos obligatorios para las opciones largas son también obligatorios

para las opciones cortas.

`-m, --mode=MODE` establece los permisos (como en `chmod`), en lugar de `a=rwx - umask`

`-p, --parents` no hay error si existen, crea los directorios padres en caso necesario

`-v, --verbose` muestra un mensaje por cada directorio creado

`-Z` establece el contexto de seguridad SELinux de cada directorio creado al tipo predeterminado

`--context[=CTX]` como `-Z`, o si se especifica `CTX` entonces establece

el

contexto de seguridad SELinux o SMACK a `CTX`

`--help` muestra esta ayuda y finaliza

`--version` informa de la versión y finaliza

```
ayuda en línea sobre GNU coreutils:  
<https://www.gnu.org/software/coreutils/>  
Report mkdir translation bugs to <https://translationproject.org/team/>  
Full documentation at: <https://www.gnu.org/software/coreutils/mkdir>  
or available locally via: info '(coreutils) mkdir invocation'
```

Some programs don't support the `--help` option, but try it anyway. Often it results in an error message that will reveal the same usage information.

Algunos programas no son compatibles con la opción `--help`, pero inténtelo de todos modos. A menudo da como resultado un mensaje de error que revelará la misma información de uso.

intenta:

```
[hernani@EliteDesk:~]$ dir --help
```

man -- Display a Program's Manual Page

man: muestra la página del manual del programa

Most executable programs intended for command line use provide a formal piece of documentation called a manual or man page. A special paging program called `man` is used to view them. It is used like this:

La mayoría de los programas ejecutables destinados a la línea de comandos proporcionan una pieza formal de documentación llamada manual o página de manual. Se utiliza un programa de localización especial llamado `man` para verlos. Se usa así:

```
man program
```

where `program` is the name of the command to view. Man pages vary somewhat in format but generally contain the following:

donde `programa` es el nombre del comando a ver. Las páginas de manual varían un poco en formato, pero generalmente contienen lo siguiente:

- A title (the page's name)
Un título (el nombre de la página)
- A synopsis of the commands syntax
Una sinopsis de la sintaxis del comando
- A description of the command's purpose
Una descripción del propósito del comando
- A listing and description of each of the commands options
Una lista y descripción de cada una de las opciones del comando.

Man pages, however, do not usually include examples and are intended as a reference, not a tutorial. As an example, let's try viewing the man page for the `ls` command.

Las páginas de manual, sin embargo, no suelen incluir ejemplos y están pensadas como referencia, no como tutorial. Como ejemplo, intentemos ver la página de manual del comando `ls`.

```
[me@linuxbox ~]$ man ls
```

On most Linux systems, `man` uses `less` to display the manual page, so all of the familiar `less` commands work while displaying the page.

En la mayoría de los sistemas Linux, `man` usa `less` para mostrar la página del manual, por lo que todos los comandos menos familiares funcionan mientras se muestra la página.

The "manual" that `man` displays is broken into sections and covers not only user commands but also system administration commands, programming interfaces, file formats, and more. Table 5-1 describes the layout of the manual.

El "manual" que muestra el hombre está dividido en secciones y cubre no solo los comandos de usuario, sino también los comandos de administración del sistema, interfaces de programación, formatos de archivo y más. La Tabla 5-1 describe el diseño del manual.

Table 5-1: Man Page Organization

Tabla 5-1: Organización de la página de manual

Section	Contents
1	User commands Comandos de usuario
2	Programming interfaces for kernel system calls Interfaces de programación para llamadas al sistema del kernel
3	Programming interfaces to the C library Interfaces de programación para la biblioteca C
4	Special files such as device nodes and drivers Archivos especiales como controladores y nodos de dispositivos
5	File formats Formatos de archivo
6	Games and amusements such as screen savers Juegos y diversiones como protectores de pantalla
7	Miscellaneous Diverso
8	System administration commands Comandos de administración del sistema

Sometimes we need to refer to a specific section of the manual to find what we are looking for. This is particularly true if we are looking for a file format that is also the name of a command. Without specifying a section number, we will always get the first instance of a match, probably in section 1.

A veces necesitamos consultar una sección específica del manual para encontrar lo que estamos buscando. Esto es particularmente cierto si buscamos un formato de archivo que también sea el nombre de un comando. Sin especificar un número de sección, siempre obtendremos la primera instancia de una coincidencia, probablemente en la sección 1

To specify a section number, we use man like this:

Para especificar un número de sección, usamos man así:

```
man section search_term
```

Here's an example:

Aquí tienes un ejemplo:

```
[me@linuxbox ~]$ man 5 passwd
```

This will display the man page describing the file format of the /etc/passwd file.

Esto mostrará la página de manual que describe el formato de archivo del archivo / etc / passwd.

apropos -- Display Appropriate Commands

apropos -- muestra los comandos adecuados

It is also possible to search the list of man pages for possible matches based on a search term. It's crude but sometimes helpful. Here is an example of a search for man pages using the search term partition:

También es posible buscar en la lista de páginas de manual posibles coincidencias basadas en un término de búsqueda. Es tosco pero a veces útil. A continuación se muestra un ejemplo de una búsqueda de páginas de manual utilizando la partición de términos de búsqueda:

```
[hernani@EliteDesk:~]$ apropos partition
addpart (8)          - tell the kernel about the existence of a partition
cfdisk (8)           - display or manipulate a disk partition table
cgdisk (8)           - Curses-based GUID partition table (GPT) manipulator
delpart (8)          - tell the kernel to forget about a partition
fdisk (8)            - manipulate disk partition table
fixparts (8)         - MBR partition table repair utility
gdisk (8)            - Interactive GUID partition table (GPT) manipulator
mpartition (1)       - partition an MSDOS hard disk
parted (8)           - a partition manipulation program
partprobe (8)        - inform the OS of partition table changes
partx (8)            - tell the kernel about the presence and numbering of
on...
resizepart (8)       - tell the kernel about the new size of a partition
sfdisk (8)           - display or manipulate a disk partition table
sgdisk (8)           - Command-line GUID partition table (GPT) manipulator
fo...
systemd-gpt-auto-generator (8) - Generator for automatically discovering
and ...
```

The first field in each line of output is the name of the man page, and the second field shows the section.

Note that the man command with the -k option performs the same function as apropos.

El primer campo de cada línea de salida es el nombre de la página de manual y el segundo campo muestra la sección. Tenga en cuenta que el comando man con la opción -k realiza la misma función que apropos.

whatis -- Display One-line Manual Page Descriptions

whatis -- Descripciones de página del manual de una línea de visualización

The `whatis` program displays the name and a one-line description of a man page matching a specified keyword.

El programa `whatis` muestra el nombre y una descripción de una línea de una página de manual que coincide con una palabra clave específica.

```
[me@EliteDesk:~$ whatis ls  
ls (1) - list directory contents
```

The Most Brutal Man Page of Them All

La página de hombre más brutal de todos

As we have seen, the manual pages supplied with Linux and other Unix-like systems are intended as reference documentation and not as tutorials. Many man pages are hard to read, but I think the grand prize for difficulty has got to go to the man page for `bash`. As I was doing research for this book, I gave the `bash` man page careful review to ensure that I was covering most of its topics.

Como hemos visto, las páginas del manual suministradas con Linux y otros sistemas similares a Unix están pensadas como documentación de referencia y no como tutoriales. Muchas páginas de manual son difíciles de leer, pero creo que el gran premio por dificultad tiene que ir a la página de manual de `bash`. Mientras investigaba para este libro, revisé cuidadosamente la página de manual de `bash` para asegurarme de que estaba cubriendo la mayoría de sus temas.

When printed, it's more than 80 pages long and extremely dense, and its structure makes absolutely no sense to a new user.

Cuando se imprime, tiene más de 80 páginas y es extremadamente densa, y su estructura no tiene ningún sentido para un nuevo usuario.

On the other hand, it is very accurate and concise, as well as being extremely complete. So check it out if you dare and look forward to the day when you can read it and it all makes sense.

Por otro lado, es muy preciso y conciso, además de ser extremadamente completo. Así que échale un vistazo si te atreves y espera el día en que puedas leerlo y todo tenga sentido.

info -- Display a Program's Info Entry

info: muestra la entrada de información de un programa

The GNU Project provides an alternative to man pages for their programs, called `info`. Info manuals are displayed with a reader program named, appropriately enough, `info`. Info pages are hyperlinked much like web pages.

El Proyecto GNU proporciona una alternativa a las páginas de manual para sus programas, llamada `info`. Los

manuales de información se muestran con un programa lector llamado, apropiadamente, info. Las páginas de información tienen hipervínculos como las páginas web

Here is a sample:

He aquí una muestra:

```
[hernani@EliteDesk:~]$ info ls
[hernani@EliteDesk:~]$ info ls > info-ls.txt
[hernani@EliteDesk:~]$ ls
archivo.txt      Escritorio      Música          snap
bin              espeak-data    openvpn-install.sh  streama
bola-navidad.pdf ffmpeg-flv-mp4.pdf  opt             sudo-usermod.pdf
Descargas        flash_seting.ini  Plantillas
tensorflow_datasets
Documentos       GNS3            practice-csr     tor-browser_en-US
dwhelper         Imágenes        pseint           tor.keyring
easy-rsa         info-ls.txt     Público          Vídeos
EliteDesk.ovpn   log-espeakedit  seting.ini       winehq.key
[hernani@EliteDesk:~]$ cat info-ls.txt
```

The info program reads info files, which are tree structured into individual nodes, each containing a single topic. Info files contain hyperlinks that can move you from node to node. A hyperlink can be identified by its leading asterisk and is activated by placing the cursor upon it and pressing enter.

El programa de información lee archivos de información, que están estructurados en árbol en nodos individuales, cada uno de los cuales contiene un solo tema. Los archivos de información contienen hipervínculos que pueden moverlo de un nodo a otro. Un hipervínculo se puede identificar por su asterisco inicial y se activa colocando el cursor sobre él y presionando enter.

To invoke info , type info followed optionally by the name of a program. Table 5-2 describes the commands used to control the reader while displaying an info page.

Para invocar información, escriba info seguido opcionalmente por el nombre de un programa. La Tabla 5-2 describe los comandos utilizados para controlar el lector mientras se muestra una página de información.

Table 5-2: info Commands

Tabla 5-2: Comandos de información

Command	Action
?	Display command help Mostrar ayuda de comandos
page up or backspace	Display previous page Mostrar página anterior
page down or spacebar	Display next page Mostrar la página siguiente

Command	Action
n	Next -- display the next node Siguiente -- muestra el siguiente nodo
p	Previous -- display the previous node Anterior -- muestra el nodo anterior
U	Up -- display the parent node of the currently displayed node, usually a menu Arriba -- muestra el nodo principal del nodo que se muestra actualmente, generalmente un menú
enter	Follow the hyperlink at the cursor location Siga el hipervínculo en la ubicación del cursor
Q	Quit dejar, abandonar

Most of the command line programs we have discussed so far are part of the GNU Project's **coreutils** package, so typing the following:

La mayoría de los programas de línea de comandos que hemos discutido hasta ahora son parte del paquete **coreutils** del Proyecto GNU, por lo que debe escribir lo siguiente:

```
[me@linuxbox ~]$ info coreutils
```

will display a menu page with hyperlinks to each program contained in the **coreutils** package.

mostrará una página de menú con hipervínculos a cada programa contenido en el **paquete coreutils**

README and Other Program Documentation Files

README y otros archivos de documentación de programas

Many software packages installed on your system have documentation files residing in the `/usr/share/doc` directory. Most of these are stored in ordinary text format and can be viewed with the `less` command. Some of the files are in HTML format and can be viewed with a web browser. We may encounter some files ending with a `.gz` extension. This indicates that they have been compressed with the `gzip` compression program. The `gzip` package includes a special version of `less` called `zless` that will display the contents of `gzip`-compressed text files.

Muchos paquetes de software instalados en su sistema tienen archivos de documentación que residen en el directorio `/usr/share/doc`. La mayoría de estos se almacenan en formato de texto normal y se pueden ver con el comando `less`. Algunos de los archivos están en formato HTML y se pueden ver con un navegador web. Es posible que encontremos algunos archivos que terminen con una extensión `.gz`. Esto indica que se han comprimido con el programa de compresión `gzip`. El paquete `gzip` incluye una versión especial de `less` llamada `zless` que mostrará el contenido de los archivos de texto comprimidos con `gzip`.

Creating Our Own Commands with alias

Creando nuestros propios comandos con alias

Now for our first experience with programming! We will create a command of our own using the alias command. But before we start, we need to reveal a small command line trick. It's possible to put more than one command on a line by separating each command with a semicolon. It works like this:

¡Ahora para nuestra primera experiencia con la programación! Crearemos un comando propio usando el comando alias. Pero antes de comenzar, necesitamos revelar un pequeño truco de línea de comandos. Es posible poner más de un comando en una línea separando cada comando con un punto y coma. Funciona así:

```
command1; command2; command3...
```

Here's the example we will use:

Este es el ejemplo que usaremos:

```
[me@linuxbox ~]$ cd /usr; ls; cd -  
bin games include lib      local      sbin share src  
/home/me  
[me@linuxbox ~]$
```

As we can see, we have combined three commands on one line. First we change the current working directory to /usr, then list the directory, and finally return to the original directory (by using cd -) so we end up where we started. Now let's turn this sequence into a new command using alias.

Como podemos ver, hemos combinado tres comandos en una línea. Primero cambiamos el directorio de trabajo actual a /usr, luego listamos el directorio y finalmente regresamos al directorio original (usando cd -) para terminar donde comenzamos. Ahora convierta esta secuencia en un nuevo comando usando un alias.

The first thing we have to do is dream up a name for our new command.

Lo primero que tenemos que hacer es inventar un nombre para nuestro nuevo comando.

Let's try test. Before we do that, it would be a good idea to find out whether the name test is already being used. To find out, we can use the type command again.

Intentemos probar. Antes de hacer eso, sería una buena idea averiguar si ya se está utilizando la prueba de nombre. Para averiguarlo, podemos usar el comando type nuevamente.

```
[me@linuxbox ~]$ type test  
test is a shell builtin
```

Oops! The name test is already taken. Let's try foo.

¡Ups! La prueba de nombre ya se ha realizado. Intentemos foo.

```
[me@linuxbox ~]$ type foo  
bash: type: foo: not found
```

Otras pruebas:

Otras pruebas:

```

hernani@EliteDesk:~$ type foo
bash: type: foo: no encontrado
hernani@EliteDesk:~$ type test
test es una orden interna del intérprete de ordenes
hernani@EliteDesk:~$ type gif.
gif. es un alias de `function _(){ cd
/media/hernani/WDatos/Share/afhernani.com/gifview/ ; eval "python3
flowlayout.py &" ; cd - ; }; _'
hernani@EliteDesk:~$ type prok.
bash: type: prok.: no encontrado
hernani@EliteDesk:~$ type pink.
pink. es un alias de `function _(){ cd
/media/hernani/WDatos/Share/afhernani.com/kivy/afhernani.com/kvcomic/ ;
eval "python3 splitfloat.py &" ; cd - ; }; _'
hernani@EliteDesk:~$ type Tor
Tor es un alias de `function _(){ eval "sh -c "/home/hernani/tor-
browser_en-US/Browser/start-tor-browser" &" ; }; _'

```

Great! foo is not taken. So, let's create our alias.

¡Excelente! foo no se toma. Entonces, creemos nuestro alias.

```
[me@linuxbox ~]$ alias foo='cd /usr; ls; cd -'
```

Notice the structure of this command, shown here:

Observe la estructura de este comando, que se muestra aquí:

```
alias name='string'
```

After the command alias , we give our alias a name followed immediately (no whitespace allowed) by an equal sign, followed immediately by a quoted string containing the meaning to be assigned to the name. After we define our alias, we can use it anywhere the shell would expect a command.

Después del alias del comando, le damos a nuestro alias un nombre seguido inmediatamente (no se permiten espacios en blanco) por un signo igual, seguido inmediatamente por una cadena entre comillas que contiene el significado que se le asignará al nombre. Después de definir nuestro alias, podemos usarlo en cualquier lugar donde el shell espere un comando.

Let's try it.

Vamos a intentarlo.

```

[me@EliteDesk:~]$ alias foo='cd /usr; ls; cd -'
hernani@EliteDesk:~$ foo
bin      include  lib32    libexec  local    share
games    lib      lib64    libx32   sbin     src
/home/hernani

```

We can also use the type command again to see our alias.

También podemos usar el comando type nuevamente para ver nuestro alias.

14 / 16

```
alias ls='ls --color=auto'
alias pdf.='function _(){ eval "evince  &" ;}; _'
alias pink.='function _(){ cd
/media/hernani/WDatos/Share/afhernani.com/kivy/afhernani.com/kvcomic/ ;
eval "python3 splitfloat.py &" ; cd - ; }; _'
alias play.='function _(){ cd /media/hernani/WDatos/dragdrop/playparse/ ;
eval "python3 playparse.py &" ; cd - ; }; _'
alias trm='gnome-terminal --window &'
```

While we purposefully avoided naming our alias with an existing command name, it is not uncommon to do so. This is often done to apply a commonly desired option to each invocation of a common command. For instance, we saw earlier how the `ls` command is often aliased to add color support.

Si bien evitamos a propósito nombrar nuestro alias con un nombre de comando existente, no es raro hacerlo. Esto se hace a menudo para aplicar una opción comúnmente deseada a cada invocación de un comando común. Por ejemplo, vimos anteriormente cómo el comando `ls` a menudo tiene un alias para agregar soporte de color.

```
[me@EliteDesk:~]$ type ls
ls es un alias de `ls --color=auto`
```

To see all the aliases defined in the environment, use the `alias` command without arguments. Here are some of the aliases defined by default on a Fedora system. Try to figure out what they all do.

Para ver todos los alias definidos en el entorno, use el comando `alias` sin argumentos. Estos son algunos de los alias definidos por defecto en un sistema Fedora. Trate de averiguar qué hacen todos.

```
alias l='ls -CF'
alias l.='ls -d .* --color=auto'
alias la='ls -A'
alias ll='ls -l --color=auto'
alias ls='ls --color=auto'
```

There is one tiny problem with defining aliases on the command line.

Existe un pequeño problema con la definición de alias en la línea de comandos.

They vanish when your shell session ends. In Chapter 11, we will see how to add our own aliases to the files that establish the environment each time we log on, but for now, enjoy the fact that we have taken our first, albeit tiny, step into the world of shell programming!

Desaparecen cuando termina su sesión de shell. En el Capítulo 11, veremos cómo agregar nuestros propios alias a los archivos que establecen el entorno cada vez que iniciamos sesión, pero por ahora, disfrute el hecho de que hemos dado nuestro primer, aunque pequeño, paso en el mundo de la programación de shell.
!

Pasando parámetros a un alias.

Creamos un alias simple pasando parametros como sigue:

```
hernani@EliteDesk:~$ alias foo='echo $?'  
hernani@EliteDesk:~$ foo codo  
0 codo
```

Como podemos apreciar la \$?, recorre todos los parametros desde 0 que corresponde al foo y a codo. limpiando el alias y re-escribiendolo:

```
hernani@EliteDesk:~$ unalias foo  
hernani@EliteDesk:~$ type foo  
bash: type: foo: no encontrado  
hernani@EliteDesk:~$ alias foo='echo $1'  
hernani@EliteDesk:~$ foo hernani  
hernani
```

Summing Up

Resumen

Now that we have learned how to find the documentation for commands, look up the documentation for all the commands we have encountered so far. Study what additional options are available and try them!

Ahora que hemos aprendido cómo encontrar la documentación de los comandos, busque la documentación de todos los comandos que hemos encontrado hasta ahora. ¡Estudie qué opciones adicionales están disponibles y pruébelas!