13 Customizing the Prompt
13 Personalización del aviso

In this chapter, we will look at a seemingly trivial detail — our shell prompt. This examination will reveal some of the inner workings of the shell and the terminal emulator program.
En este capítulo, veremos un detalle aparentemente trivial: nuestro indicador de shell. Este examen revelará algunos de los trabajos internos del shell y el programa emulador de terminal.

Like so many things in Linux, the shell prompt is highly configurable, and while we have pretty much taken it for granted, the prompt is a really useful device once we learn how to control it.
Como tantas cosas en Linux, el indicador de shell es altamente configurable, y aunque lo hemos dado por sentado, el indicador es un dispositivo realmente útil una vez que aprendemos a controlarlo.

Anatomy of a Prompt
Anatomía de un indicador

Our default prompt looks something like this:
Nuestro indicador predeterminado se parece a esto:

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

Notice that it contains our username, our hostname, and our current working directory, but how did it get that way? Very simply, it turns out. The prompt is defined by an environment variable named PS1 (short for “prompt string 1”). We can view the contents of PS1 with the echo command.
Tenga en cuenta que contiene nuestro nombre de usuario, nuestro nombre de host y nuestro directorio de trabajo actual, pero ¿cómo llegó a ser así? Resulta muy simple. La solicitud se define mediante una variable de entorno denominada PS1 (abreviatura de "cadena de solicitud 1"). Podemos ver el contenido de PS1 con el comando echo.

Don’t worry if your results are not the same as this example. Every Linux distribution defines the prompt string a little differently, some quite exotically. From the results, we can see that PS1 contains a few of the characters we see in our prompt such as the brackets, the at-sign, and the dollar sign, but the rest are a mystery. The astute among us will recognize these as backslashescaped special characters like those we saw in Chapter 7. Table 13-1 provides a partial list of the characters that the bash treats specially in the prompt string.
No se preocupe si sus resultados no son los mismos que en este ejemplo. Cada distribución de Linux define la cadena de solicitud de forma un poco diferente, algunas de forma bastante exótica. A partir de los resultados, podemos ver que PS1 contiene algunos de los caracteres que vemos en nuestro mensaje, como los corchetes, el signo arroba y el signo de dólar, pero el resto es un misterio. Los astutos los reconocerán como caracteres especiales con barra invertida, como los que vimos en el Capítulo 7. La tabla 13-1 proporciona una lista parcial de los caracteres que el bash trata especialmente en la cadena de comandos.

Table 13-1: Escape Codes Used in Shell Prompts
Tabla 13-1: Códigos de escape utilizados en las indicaciones de Shell

Sequence secuencia	Value displayed valor presentado
\a	ASCII bell. This makes the computer beep when it is encountered. Campana ASCII. Esto hace que la computadora emita un pitido cuando se encuentra.
\d	Current date in day, month, date format. For example, “Mon May 26.” Fecha actual en formato de día, mes y fecha. Por ejemplo, "Lunes 26 de mayo".
\h	Hostname of the local machine minus the trailing domain name. nombre de host de la máquina local menos el nombre de dominio final.
\H	Full hostname. Nombre entero del host
\j	Number of jobs running in the current shell session. Número de trabajos que se ejecutan en la sesión de shell actual.
\l	Name of the current terminal device. Nombre del dispositivo terminal actual.
\n	A newline character. Caracter de nueva linea
\r	A carriage return. Retorno de carro
\s	Name of the shell program. Nombre de la shell del programa
\t	Current time in 24-hour hours:minutes:seconds format. Hora actual en horas de 24 horas: minutos: formato de segundos
\T	Current time in 12-hour format. Hora actual en formato de 12 horas
@	Current time in 12-hour Hora actual en 12 horas
\A	Current time in 24-hour hours:minutes format. Hora actual en horas de 24 horas: formato de minutos.
\u	Username of the current user. Nombre de usuario del usuario actual.
\v	Version number of the shell.
\V	Version and release numbers of the shell. Versión y números de lanzamiento del shell.
\w	Name of the current working directory. Nombre del directorio de trabajo actual.

Sequence secuencia	Value displayed valor presentado
\w	Last part of the current working directory name. Última parte del nombre del directorio de trabajo actual
\!	History number of the current command. Número de historial del comando actual.
\#	Number of commands entered during this shell session. Número de comandos ingresados durante esta sesión de shell
\$	This displays a $ character unless we have superuser privileges. In that case, it displays a # instead. Esto muestra un carácter $ a menos que tengamos privilegios de superusuario. En ese caso, muestra un # en su lugar.
\[	Signals the start of a series of one or more non-printing characters. This is used to embed non-printing control characters that manipulate the terminal emulator in some way, such as moving the cursor or changing text colors. Señala el inicio de una serie de uno o más caracteres que no se imprimen. Esto se usa para incrustar caracteres de control que no son de impresión y que manipulan el emulador de terminal de alguna manera, como mover el cursor o cambiar los colores del texto.
\]	Signals the end of a non-printing character sequence. Señala el final de una secuencia de caracteres que no se imprimen.

Trying Some Alternative Prompt Designs
Probar algunos diseños de mensajes alternativos

With this list of special characters, we can change the prompt to see the effect. First, we'll back up the existing prompt string so we can restore it later. To do this, we will copy the existing string into another shell variable that we create ourselves.
Con esta lista de caracteres especiales, podemos cambiar el mensaje para ver el efecto. Primero, haremos una copia de seguridad de la cadena de solicitud existente para poder restaurarla más tarde. Para hacer esto, copiaremos la cadena existente en otra variable de shell que creamos nosotros mismos.

```
[me@linuxbox ~]$ ps1_old="$PS1"
```

We create a new variable called ps1_old and assign the value of PS1 to it. We can verify that the string has been copied by using the echo command.
Creamos una nueva variable llamada ps1_old y le asignamos el valor de PS1. Podemos verificar que la cadena ha sido copiada usando el comando echo.

```
[me@linuxbox ~]$ echo $ps1_old
[\u@\h \W]\$
```

We can restore the original prompt at any time during our terminal session by simply reversing the process.
Podemos restaurar el mensaje original en cualquier momento durante nuestra sesión de terminal simplemente invirtiendo el proceso.

```
[me@linuxbox ~]$ PS1="$ps1_old"
```

Now that we are ready to proceed, let's see what happens if we have an empty prompt string.
Ahora que estamos listos para continuar, veamos qué sucede si tenemos una cadena de solicitud vacía.

```
[me@linuxbox ~]$ PS1=
```

If we assign nothing to the prompt string, we get nothing. No prompt string at all! The prompt is still there but displays nothing, just as we asked it to do. Because this is kind of disconcerting to look at, we'll replace it with a minimal prompt.
Si no asignamos nada a la cadena de solicitud, no obtenemos nada. ¡Ninguna cadena de aviso en absoluto! El mensaje sigue ahí, pero no muestra nada, tal como le pedimos. Debido a que esto es algo desconcertante de ver, lo reemplazaremos con un aviso mínimo.

```
PS1="\$ "
```

That's better. At least now we can see what we are doing. Notice the trailing space within the double quotes. This provides the space between the dollar sign and the cursor when the prompt is displayed.
Eso es mejor. Al menos ahora podemos ver lo que estamos haciendo. Observe el espacio final dentro de las comillas dobles. Esto proporciona el espacio entre el signo de dólar y el cursor cuando se muestra el mensaje.

Let's add a bell to our prompt.
Agreguemos una campana a nuestro mensaje.

```
$ PS1="\[\a\]\$ "
```

Now we should hear a beep each time the prompt is displayed, though some systems disable this "feature." This could get annoying, but it might be useful if we needed notification when an especially long-running command has been executed. Note that we included the \[and \] sequences. Since the ASCII bell (\a) does not "print," that is, it does not move the cursor, we need to tell bash so it can correctly determine the length of the prompt.
Ahora deberíamos escuchar un pitido cada vez que se muestra el mensaje, aunque algunos sistemas desactivan esta "función". Esto podría resultar molesto, pero podría ser útil si necesitáramos una notificación cuando se haya ejecutado un comando de ejecución especialmente prolongada. Tenga en

cuenta que incluimos las secuencias \ [y \]. Dado que la campana ASCII (\ a) no "imprime", es decir, no mueve el cursor, necesitamos decirle a bash para que pueda determinar correctamente la longitud del indicador.

Next, let's try to make an informative prompt with some hostname and time-of-day information.
A continuación, intentemos crear un mensaje informativo con información sobre el nombre de host y la hora del día.

```
$ PS1="\A \h \$ "
17:33 linuxbox $
```

Adding time-of-day to our prompt will be useful if we need to keep track of when we perform certain tasks. Finally, we'll make a new prompt that is similar to our original.
Agregar la hora del día a nuestro mensaje será útil si necesitamos realizar un seguimiento de cuándo realizamos ciertas tareas. Finalmente, crearemos un nuevo mensaje similar al original.

```
17:37 linuxbox $ PS1="<\u@\h \W>\$ "
<me@linuxbox ~>$
```

Try the other sequences listed in Table 13-1 and see whether you can come up with a brilliant new prompt.
Pruebe las otras secuencias enumeradas en la Tabla 13-1 y vea si puede encontrar un mensaje nuevo y brillante.

Adding Color
Añadiendo color

Most terminal emulator programs respond to certain non-printing character sequences to control such things as character attributes (such as color, bold text, and the dreaded blinking text) and cursor position. We'll cover cursor position in a little bit, but first we'll look at color.
La mayoría de los programas emuladores de terminal responden a ciertas secuencias de caracteres que no se imprimen para controlar cosas como los atributos de los caracteres (como el color, el texto en negrita y el temido texto parpadeante) y la posición del cursor. Cubriremos la posición del cursor en un momento, pero primero veremos el color.

Terminal Confusion
Confusión terminal

> Back in ancient times, when terminals were hooked to remote computers, there were many competing brands of terminals, and they all worked differently. They had different keyboards, and they all had different ways of interpreting control information. Unix and Unix-like systems have two rather complex subsystems to deal with the babel of terminal control (called termcap and terminfo). If you look in the deepest recesses of your terminal emulator settings, you might find a setting for the type of terminal emulation.
> En la antigüedad, cuando las terminales estaban conectadas a computadoras remotas, había muchas marcas de terminales en competencia y todas funcionaban de manera diferente. Tenían diferentes

> teclados y todos tenían diferentes formas de interpretar la información de control. Los sistemas Unix y similares a Unix tienen dos subsistemas bastante complejos para lidiar con el babel del control de terminal (llamado termcap y terminfo). Si busca en los rincones más profundos de la configuración de su emulador de terminal, es posible que encuentre una configuración para el tipo de emulación de terminal.

> In an effort to make terminals speak some sort of common language, the American National Standards Institute (ANSI) developed a standard set of character sequences to control video terminals. Old-time DOS users will remember the ANSI.SYS file that was used to enable interpretation of these codes.
> En un esfuerzo por hacer que los terminales hablen algún tipo de lenguaje común, el Instituto Nacional Estadounidense de Estándares (ANSI) desarrolló un conjunto estándar de secuencias de caracteres para controlar los terminales de video. Los usuarios de DOS antiguos recordarán el archivo ANSI.SYS que se utilizó para permitir la interpretación de estos códigos.

Character color is controlled by sending the terminal emulator an ANSI escape code embedded in the stream of characters to be displayed. The control code does not “print out” on the display; rather, it is interpreted by the terminal as an instruction. As we saw in Table 13-1, the \[and \] sequences are used to encapsulate non-printing characters. An ANSI escape code begins with an octal 033 (the code generated by the esc key), followed by an optional character attribute, followed by an instruction. For example, the code to set the text color to normal (attribute = 0), black text is as follows:
El color de los caracteres se controla enviando al emulador de terminal un código de escape ANSI incrustado en el flujo de caracteres que se mostrarán. El código de control no se “imprime” en la pantalla; más bien, el terminal lo interpreta como una instrucción. Como vimos en la Tabla 13-1, las secuencias \ [y \] se utilizan para encapsular caracteres que no se imprimen. Un código de escape ANSI comienza con un octal 033 (el código generado por la tecla esc), seguido de un atributo de carácter opcional, seguido de una instrucción. Por ejemplo, el código para establecer el color del texto en normal (atributo = 0), el texto en negro es el siguiente:

```
\033[0;30m
```

Table 13-2 lists the available text colors. Notice that the colors are divided into two groups, differentiated by the application of the bold character attribute (1), which creates the appearance of “light” colors.
La Tabla 13-2 enumera los colores de texto disponibles. Observe que los colores se dividen en dos grupos, diferenciados por la aplicación del atributo de carácter en negrita (1), que crea la apariencia de colores “claros”.

Table 13-2: Escape Sequences Used to Set Text Colors
Tabla 13-2: Secuencias de escape utilizadas para establecer colores de texto

Sequence	Text color	Sequence	Text color
\033[0;30m	Black	\033[1;30m	Dark gray
\033[0;31m	Red	\033[1;31m	Light red
\033[0;32m	Green	\033[1;32m	Light green

Sequence	Text color	Sequence	Text color
\033[0;33m	Brown	\033[1;33m	Yellow
\033[0;34m	Blue	\033[1;34m	Light blue
\033[0;35m	Purple	\033[1;35m	Light purple
\033[0;36m	Cyan	\033[1;36m	Light cyan
\033[0;37m	Light gray	\033[1;37m	White

Let's try to make a red prompt (seen here as gray). We'll insert the escape code at the beginning.
Intentemos hacer un mensaje rojo (visto aquí como gris). Insertaremos el código de escape al principio.

```
<me@linuxbox ~>$ PS1="\[\033[0;31m\]<\u@\h \W>\$ "
<me@linuxbox ~>$
```

That works, but notice that all the text that we type after the prompt will also display in red. To fix this, we will add another escape code to the end of the prompt that tells the terminal emulator to return to the previous color.
Eso funciona, pero observe que todo el texto que escribimos después del mensaje también se mostrará en rojo. Para solucionar esto, agregaremos otro código de escape al final del mensaje que le dice al emulador de terminal que regrese al color anterior.

```
<me@linuxbox ~>$ PS1="\[\033[0;31m\]<\u@\h \W>\$\[\033[0m\] "
<me@linuxbox ~>$
```

That's better!
¡Eso es mejor!

It's also possible to set the text background color using the codes listed in Table 13-3. The background colors do not support the bold attribute.
También es posible establecer el color de fondo del texto utilizando los códigos enumerados en la Tabla 13-3. Los colores de fondo no admiten el atributo negrita.

Table 13-3: Escape Sequences Used to Set Background Color
Tabla 13-3: Secuencias de escape utilizadas para establecer el color de fondo

Sequence	Background color	Sequence	Background color
\033[0;40m	Black	\033[0;44m	Blue
\033[0;41m	Red	\033[0;45m	Purple
\033[0;42m	Green	\033[0;46m	Cyan
\033[0;43m	Brown	\033[0;47m	Light gray

We can create a prompt with a red background by applying a simple change to the first escape code.
Podemos crear un mensaje con un fondo rojo aplicando un simple cambio al primer código de escape.

```
<me@linuxbox ~>$ PS1="\[\033[0;41m\]<\u@\h \W>\$\[\033[0m\] "
<me@linuxbox ~>$
```

Try the color codes and see what you can create!
¡Prueba los códigos de colores y descubre lo que puedes crear!

> Note
> Nota
> Besides the normal (0) and bold (1) character attributes, text may be given underscore (4), blinking (5), and inverse (7) attributes. In the interests of good taste, many terminal emulators refuse to honor the blinking attribute, however.
> Además de los atributos de carácter normal (0) y negrita (1), el texto puede tener atributos de subrayado (4), parpadeo (5) e inverso (7). Sin embargo, en aras del buen gusto, muchos emuladores de terminal se niegan a respetar el atributo de parpadeo.

Moving the Cursor
Mover el cursor

Escape codes can be used to position the cursor. This is commonly used to provide a clock or some other kind of information at a different location on the screen, such as in an upper corner each time the prompt is drawn. Table 13-4 lists the escape codes that position the cursor.
Se pueden usar códigos de escape para colocar el cursor. Esto se usa comúnmente para proporcionar un reloj o algún otro tipo de información en una ubicación diferente en la pantalla, como en una esquina superior cada vez que se dibuja el mensaje. La Tabla 13-4 enumera los códigos de escape que colocan el cursor.

Table 13-4: Cursor Movement Escape Sequences
Tabla 13-4: Secuencias de escape del movimiento del cursor

Escape code **Codigo de escape**	**Action** **Acción**
\033[l;cH	Move the cursor to line l and column c Mueva el cursor a la línea ly la columna c
\033[nA	Move the cursor up n lines Mueve el cursor hacia arriba n líneas
\033[nB	Move the cursor down n lines Mueve el cursor hacia abajo n líneas

Escape code Codigo de escape	Action Acción
\033[nC	Move the cursor forward n characters Mueve el cursor hacia adelante n caracteres
\033[nD	Move the cursor backward n characters Mover el cursor hacia atrás n caracteres
\033[2J	Clear the screen and move the cursor to the upper-left corner (line 0, column 0) Limpie la pantalla y mueva el cursor a la esquina superior izquierda (línea 0, columna 0)
\033[K	Clear from the cursor position to the end of the current line Limpiar desde la posición del cursor hasta el final de la línea actual
\033[s	Store the current cursor position Almacenar la posición actual del cursor
\033[u	Recall the stored cursor position Recuperar la posición del cursor almacenada

Using the codes in Table 13-4, we'll construct a prompt that draws a red bar at the top of the screen containing a clock (rendered in yellow text) each time the prompt is displayed. The code for the prompt is this formidable looking string:
Usando los códigos de la Tabla 13-4, construiremos un mensaje que dibuja una barra roja en la parte superior de la pantalla que contiene un reloj (representado en texto amarillo) cada vez que se muestra el mensaje. El código del indicador es esta cadena de aspecto formidable:

```
PS1="\[\033[s\033[0;0H\033[0;41m\033[K\033[1;33m\t\033[0m\033[u\]<\u@\h
\W>\$ "
```

Table 13-5 outlines what each part of the string does.
La Tabla 13-5 describe lo que hace cada parte de la cadena.

> Table 13-5: Breakdown of Complex Prompt String
> Tabla 13-5: Desglose de la cadena de solicitud compleja

Sequence	Action
\[	Begin a non-printing character sequence. The purpose of this is to allow bash to properly calculate the size of the visible prompt. Without an accurate calculation, command line editing features cannot position the cursor correctly. Comience una secuencia de caracteres que no se impriman. El propósito de esto es permitir que bash calcule correctamente el tamaño del indicador visible. Sin un cálculo preciso, las funciones de edición de la línea de comandos no pueden colocar el cursor correctamente.

Sequence	Action
\033[s	Store the cursor position. This is needed to return to the prompt location after the bar and clock have been drawn at the top of the screen. Be aware that some terminal emulators do not recognize this code. Almacene la posición del cursor. Esto es necesario para volver a la ubicación del mensaje después de que se hayan dibujado la barra y el reloj en la parte superior de la pantalla. Tenga en cuenta que algunos emuladores de terminal no reconocen este código.
\033[0;0H	Move the cursor to the upper-left corner, which is line 0, column 0. Mueva el cursor a la esquina superior izquierda, que es la línea 0, columna 0.
\033[0;41m	Set the background color to red. Establecer el color de fondo en rojo
\033[K	Clear from the current cursor location (the top-left corner) to the end of the line. Because the background color is now red, the line is cleared to that color, creating our bar. Note that clearing to the end of the line does not change the cursor position, which remains in the upper-left corner. Limpiar desde la ubicación actual del cursor (la esquina superior izquierda) hasta el final de la línea. Debido a que el color de fondo ahora es rojo, la línea se borra a ese color, creando nuestra barra. Tenga en cuenta que borrar hasta el final de la línea no cambia la posición del cursor, que permanece en la esquina superior izquierda.
\033[1;33m	Set the text color to yellow. Establezca el color del texto en amarillo.
\t	Display the current time. While this is a “printing” element, we still include it in the non-printing portion of the prompt since we don’t want bash to include the clock when calculating the true size of the displayed prompt. Muestra la hora actual. Si bien este es un elemento de "impresión", aún lo incluimos en la parte no imprimible del mensaje, ya que no queremos que bash incluya el reloj al calcular el tamaño real del mensaje mostrado.
\033[0m	Turn off color. This affects both the text and the background. Apague el color. Esto afecta tanto al texto como al fondo.
\033[u	Restore the cursor position saved earlier. Restaura la posición del cursor guardada anteriormente.
\]	End the non-printing characters sequence. Finaliza la secuencia de caracteres que no se imprimen.
<\u@\h \W>$	Prompt string. Prompt string.

Saving the Prompt
Guardando el mensaje

Obviously, we don’t want to be typing that monster all the time, so we’ll want to store our prompt someplace. We can make the prompt permanent by adding it to our .bashrc file. To do so, add these two

lines to the file:
Obviamente, no queremos estar escribiendo ese monstruo todo el tiempo, por lo que querremos almacenar nuestro mensaje en algún lugar. Podemos hacer que el mensaje sea permanente agregándolo a nuestro archivo .bashrc. Para hacerlo, agregue estas dos líneas al archivo:

```
PS1="\[\033[s\033[0;0H\033[0;41m\033[K\033[1;33m\t\033[0m\033[u\]<\u@\h
\W>\$ "
export PS1
```

Summing Up
Resumen

Believe it or not, there is much more that can be done with prompts involving shell functions and scripts that we haven't covered here, but this is a good start. Not everyone will care enough to change the prompt since the default prompt is usually satisfactory. But for those of us who like to tinker, the shell provides the opportunity for many hours of casual fun.
Lo crea o no, hay mucho más que se puede hacer con indicaciones que involucran funciones de shell y scripts que no hemos cubierto aquí, pero este es un buen comienzo. No todo el mundo se preocupará lo suficiente como para cambiar el mensaje, ya que el mensaje predeterminado suele ser satisfactorio. Pero para aquellos de nosotros que nos gusta jugar, el caparazón brinda la oportunidad de muchas horas de diversión informal.