Un peu plus loin dans la commande echo

1. Les commandes spéciales:

Sequence	description	Example/Preview
\d	The date (`+%a\ %b\ %d` format) [1]	`Mon Jun 13`
\D{*format*}	The date in the desired *format* (in `strftime` format) [2]	
\t	The time, 24-hour (`+%k:%M:%S` format) [3]	`22:42:01`
\T	The time, 12-hour (`+%l:%M:%S` format) [4]	`10:42:01`
\@	The time, 12-hour, with AM/PM (`+%l:%M\ %p` format) [5]	`10:42 PM` or `10:42`
\A	The time, 24-hour (`+%k:%M` format) [6]	`22:42`
\h	The host name	`hostname`
\H	The full host name (with the domain name)	`hostname.domain`
\j	The number of suspended processes in the current shell (<Ctrl>+Z)	`0`
\l	The name of the shell's terminal device	`tty1` or `1`
\s	The name of the shell executable	`bash`
\u	The current user name	`username`
\v	The version of the shell (short)	`4.1`
\V	The version of the shell (with the patch level)	`4.1.5`
\w	The path of the working directory	`~/Documents` or `/usr/bin`
\W	The name of the working directory	`Documents` or `bin`
\!	The current command number in the history	`538`
\#	The command number (from the start of the shell)	`42`
\$	If the current user is root, displays a `#`, else displays a `$`	`$` or `#`

\\[	Start a sequence of non-printing characters	
\\]	End a sequence of non-printing characters	
\\a \\007	The ASCII bell character	
\\n	Start a new line	
\\r	Carriage return	
\\\\	A single backslash	\\
\\e \\033	The ASCII `Escape` character. Used by some [Control Sequences](#) (see bellow)	

2. Le caractère d'échappement

La norme ANSI (American National Standards Institute) définie un certain nombre de caractères qui permettent d'afficher un peu plus que du texte blanc sur fond noir.

Il est possible d'afficher des couleurs et du texte formaté en utilisant les séquences d'échappement.

En Bash le caractère d'échappement peut-être obtenu ainsi :

- \\e
- \\033 (notation octale)
- \\x1B (notation hexadécimale)

Exemples:

```
echo -e "\e[31mHello World\e[0m"
```
`Hello World`

```
echo -e "\033[31mHello\e[0m World"
```

`Hello World`

Note :

- L'option –e permet d'activer le caractère d'échappement,
- \\e[0m permet une remise à zéro de la couleur et du formatage,
- Ceci peut être utilisé dans n'importe quel langage qui écrit des caractères dans un terminal.

3. Le contrôle du curseur

ANSI	terminfo equivalent	Description
[<X> ; <Y> H [<X> ; <Y> f	cup <X> <Y>	Home-positioning to X and Y coordinates
[H	home	Home-positioning to root (0-0)
7	sc	Save current cursor position
8	rc	Restore current cursor position

4. Formatage :

Activation :

Code	Description	Example	Preview
1	Bold/Bright	echo -e "Normal \e[1mBold"	Normal **Bold**
2	Dim	echo -e "Normal \e[2mDim"	Normal Dim
4	Underlined	echo -e "Normal \e[4mUnderlined"	Normal <u>Underlined</u>
5	Blink [1]	echo -e "Normal \e[5mBlink"	Normal
7	Reverse (invert the foreground and background colors)	echo -e "Normal \e[7minverted"	Normal inverted
8	Hidden (usefull for passwords)	echo -e "Normal \e[8mHidden"	Normal

Désactivation :

Code	Description	Example	Preview
0	Reset all attributes	echo -e "\e[0mNormal Text"	Normal Text
21	Reset bold/bright	echo -e "Normal \e[1mBold \e[21mNormal"	Normal **Bold** Normal

22	Reset dim	`echo -e "Normal \e[2mDim \e[22mNormal"`	Normal Dim Normal
24	Reset underlined	`echo -e "Normal \e[4mUnderlined \e[24mNormal"`	Normal Underlined Normal
25	Reset blink	`echo -e "Normal \e[5mBlink \e[25mNormal"`	Normal Normal
27	Reset reverse	`echo -e "Normal \e[7minverted \e[27mNormal"`	Normal inverted Normal
28	Reset hidden	`echo -e "Normal \e[8mHidden \e[28mNormal"`	Normal Normal

5. 8/16 Couleurs

Foreground (text)

Code	Color	Example	Preview
39	Default foreground color	`echo -e "Default \e[39mDefault"`	Default Default
30	Black	`echo -e "Default \e[30mBlack"`	Default Black
31	Red	`echo -e "Default \e[31mRed"`	Default Red
32	Green	`echo -e "Default \e[32mGreen"`	Default Green
33	Yellow	`echo -e "Default \e[33mYellow"`	Default Yellow
34	Blue	`echo -e "Default \e[34mBlue"`	Default Blue
35	Magenta	`echo -e "Default \e[35mMagenta"`	Default Magenta
36	Cyan	`echo -e "Default \e[36mCyan"`	Default Cyan
37	Light gray	`echo -e "Default \e[37mLight gray"`	Default Light gray
90	Dark gray	`echo -e "Default \e[90mDark gray"`	Default Dark gray
91	Light red	`echo -e "Default \e[91mLight red"`	Default Light red
92	Light green	`echo -e "Default \e[92mLight green"`	Default Light green

93	Light yellow	`echo -e "Default \e[93mLight yellow"`	Default Light yellow
94	Light blue	`echo -e "Default \e[94mLight blue"`	Default Light blue
95	Light magenta	`echo -e "Default \e[95mLight magenta"`	Default Light magenta
96	Light cyan	`echo -e "Default \e[96mLight cyan"`	Default Light cyan
97	White	`echo -e "Default \e[97mWhite"`	Default White

Background

Code	Color	Example	Preview
49	Default background color	`echo -e "Default \e[49mDefault"`	Default Default
40	Black	`echo -e "Default \e[40mBlack"`	Default Black
41	Red	`echo -e "Default \e[41mRed"`	Default Red
42	Green	`echo -e "Default \e[42mGreen"`	Default Green
43	Yellow	`echo -e "Default \e[43mYellow"`	Default Yellow
44	Blue	`echo -e "Default \e[44mBlue"`	Default Blue
45	Magenta	`echo -e "Default \e[45mMagenta"`	Default Magenta
46	Cyan	`echo -e "Default \e[46mCyan"`	Default Cyan
47	Light gray	`echo -e "Default \e[47mLight gray"`	Default Light gray
100	Dark gray	`echo -e "Default \e[100mDark gray"`	Default Dark gray
101	Light red	`echo -e "Default \e[101mLight red"`	Default Light red
102	Light green	`echo -e "Default \e[102mLight green"`	Default Light green
103	Light yellow	`echo -e "Default \e[103mLight yellow"`	Default Light yellow

104	Light blue	echo -e "Default \e[104mLight blue"	Default Light blue
105	Light magenta	echo -e "Default \e[105mLight magenta"	Default Light magenta
106	Light cyan	echo -e "Default \e[106mLight cyan"	Default Light cyan
107	White	echo -e "Default \e[107mWhite"	Default

6. 88/256 Couleurs

Foreground (texte)

Il est possible, dans certain terminaux d'utiliser jusqu'à 256 couleurs :

```
\e[38;5;ColorNumberm
```


Exemples:

```
echo -e "\e[38;5;82mHello \e[38;5;198mWorld"
```

```
for i in {16..21} {21..16} ; do echo -en "\e[38;5;${i}m#\e[0m" ; done ; echo
```

Background

```
\e[48;5;ColorNumberm
```


7. Les combinaisons d'attributs

Il est possible de faire des combinaisons mais les attributs doivent être séparés par des « ; » :

```
echo -e "\e[1;4mBold and Underlined"
```



```
echo -e "\e[1;31;42m Yes it is awful \e[0m"
```

8. Scripts de démonstration :

```bash
#!/bin/bash

# This program is free software. It comes without any warranty, to
# the extent permitted by applicable law. You can redistribute it
# and/or modify it under the terms of the Do What The Fuck You Want
# To Public License, Version 2, as published by Sam Hocevar. See
# http://sam.zoy.org/wtfpl/COPYING for more details.

#Background
for clbg in {40..47} {100..107} 49 ; do
        #Foreground
        for clfg in {30..37} {90..97} 39 ; do
                #Formatting
                for attr in 0 1 2 4 5 7 ; do
                        #Print the result
                        echo -en "\e[${attr};${clbg};${clfg}m ^[${attr};${clbg};${clfg}m
\e[0m"
                done
                echo #Newline
        done
done

exit 0
```

```bash
#!/bin/bash

# This program is free software. It comes without any warranty, to
# the extent permitted by applicable law. You can redistribute it
# and/or modify it under the terms of the Do What The Fuck You Want
# To Public License, Version 2, as published by Sam Hocevar. See
# http://sam.zoy.org/wtfpl/COPYING for more details.

for fgbg in 38 48 ; do #Foreground/Background
        for color in {0..256} ; do #Colors
                #Display the color
                echo -en "\e[${fgbg};5;${color}m ${color}\t\e[0m"
                #Display 10 colors per lines
                if [ $((($color + 1) % 10)) == 0 ] ; then
                        echo #New line
                fi
        done
        echo #New line
done

exit 0
```