echo in Color

This document describes how to use the builtin echo functionality of BASH to produce color output on the terminal. To print in color one must first send the control sequences for the color to the terminal, then output the message and reset the terminal, to be nice. One can also create nifty looking prompts with the teniques described here.

Changing the Colors

The below key describes all the different sequences that can be sent to change the display format. The disclaimer should be made that "this is how the test computer performed".

```
Style           Foreground      Background
1st Digit       2nd Digit       3rd Digit
0 - Reset       30 - Black      40 - Black
1 - FG Bright   31 - Red        41 - Red
2 - Unknown     32 - Green      42 - Green
3 - Unknown     33 - Yellow     43 - Yellow
4 - Underline   34 - Blue       44 - Blue
5 - BG Bright   35 - Magenta    45 - Magenta
6 - Unknown     36 - Cyan       46 - Cyan
7 - Reverse     37 - White      47 - White
```

To instruct `echo` to interpret these codes you must tell it `-en`. The `-e` switch tells echo to interpret your escape sequences and `-n` tells echo not to make a newline at the end.The activation sequence is `\033` this starts the output modification codes. Next is a `[` followed by a digits from the above list to represent the style, foreground and background. The sequence is terminated by the letter m. The sequence to get plain red on black would look like this: `echo -en "\033[0;31;40m"` or once could say `echo -en "\033[0;31m"` to only affect the foreground. Portions of the sequence can be left out but the digits are still interpreted in the same order. One can switch only the background by saying `echo -en "\033[7;43m"`, this would change the background to yellow without affecting the current foreground settings.

After the control sequence has been sent the output that follows will use the specified colors until it is reset. Some programs that are run may reset the terminal.

Once output is complete the terminal should be reset with `echo -e "\033[0m"`.

Echo in Color Test Script

The code below is a sample of echo in color, it will run through all the sequences and ouput some nice looking tables. This can be use to test what different colors will look like as well as show the hidious combinations.

```
#/bin/sh
# Show all the colors of the rainbow, should be run under bash
for STYLE in 0 1 2 3 4 5 6 7; do
  for FG in 30 31 32 33 34 35 36 37; do
    for BG in 40 41 42 43 44 45 46 47; do
      CTRL="\033[${STYLE};${FG};${BG}m"
      echo -en "${CTRL}"
      echo -n "${CTRL}"
      echo -en "\033[0m"
    done
    echo
  done
  echo
done
# Reset
echo -e "\033[0m"
```

printf() in color

This document describes how to produce output from printf in color. There only three steps to using printf on color.

1. Tell the terminal that it should print in color
2. Write text of desired length
3. Reset the console to the normal color.

Using printf in color is actually way easier that one would think. The first part is to tell the terminal that it should print in color; tell it what attributes, foreground and background colors to use. This is usually done with escape sequences similar to the following: `^[[1;33;40m`. This escape sequence would set the foreground to bright red and the background to black. It should also be mentioned that the first two characters are actually one. '^[' is created by CTRL+V, ESC. How do you tell the console this information? With printf. More on the colours and their number ordering later.

Using printf to make this colors is really easy, check this out:

```
printf("\033[1;36m");     // set colours
printf("Hello World");    // print
printf("\033[0m");        // reset colours
```

The above code would output the phrase "Hello World" in bright cyan. Notice the `\033`, this is the octal code for `^[` the start of our escape sequence, `\x1b` could also be used if one chooses to use hexidecimal over octal. After writing our phrase the colours are reset using `\033[0m`.

Truly that is all there is it to it. The only thing now is to learn the color code escape sequences. Some examples follow and a key is provided below that.

```
printf("\x1b[1;31;40m");      // Bright red on black
printf("\x1b[3;33;45m");      // Blinking yellow on magenta
printf("\x1b[1;30;47m");      // Bright black (grey) on dim white

Style           Foreground      Background
1st Digit       2nd Digit       3rd Digit
0 - Reset       30 - Black      40 - Black
1 - FG Bright   31 - Red        41 - Red
2 - Unknown     32 - Green      42 - Green
3 - Unknown     33 - Yellow     43 - Yellow
4 - Underline   34 - Blue       44 - Blue
5 - BG Bright   35 - Magenta    45 - Magenta
6 - Unknown     36 - Cyan       46 - Cyan
7 - Reverse     37 - White      47 - White
```

Last Modified: Thu, 24 Mar 2005 23:57:37 GMT