

MISE EN FORME DU TEXTE DANS LE TERMINAL
29 mars 2019 — Jawher Jarrey, Hiba Ouni, Pierre Rousselin et Xavier Monnin

Certains terminaux ou émulateurs de terminaux laissent une certaine liberté sur la mise en forme des symboles (couleurs, style, clignotement, etc). Au départ chaque fabricant avait ses propres codes de mise en forme, mais ceci a été standardisé (*ANSI Escape Codes*). Cela dit, la prise en compte ou non de tous les codes dépend encore un peu de l'émulateur de terminal utilisé.

Ces couleurs sont souvent utilisées :

- pour faire une jolie invite de commande (variable d'environnement `PS1`, voir exercice 2) ;
- (au moins dans les systèmes GNU) pour un affichage pratique avec `ls --colors` (voir la variable d'environnement `LS_COLORS`) ;
- (au moins dans les systèmes GNU) pour voir la ou les chaînes qui correspondent avec `grep`, on peut utiliser `grep --color=auto` (et cela peut être configuré avec la variable d'environnement `GREP_COLOR`)

Un code d'échappement (couleur ou autre) est formé des caractères suivants :

1. le caractère d'échappement de code ASCII 27, donc en octal `\033`, en hexadécimal `\x1B` ou avec une séquence d'échappement spéciale (dans bash, donc moins portable) `\e`
2. le caractère `[`
3. le ou les styles, sous la forme d'entiers en écriture décimale, s'il y en a plusieurs, séparés par des `;`
4. la lettre `m` pour signaler la fin du style.

Remarque importante : pour utiliser ces codes, il faut que la commande reconnaisse ces séquences, on utilisera donc `echo -e`, ou mieux, `printf`.

Avant de donner les codes, voici un premier exemple :

```
printf "Ceci est vraiment \033[1;31;4m TRÈS IMPORTANT \033[0m et ceci l'est beaucoup moins\n"
Ceci est vraiment TRÈS IMPORTANT et ceci l'est beaucoup moins
```

Ici, le premier code comporte les entiers :

- 1 pour mettre en gras ;
- 31 pour mettre en rouge ;
- 4 pour souligner.

Le code suivant contient seulement l'entier 0 pour revenir à la normale. Évidemment, il vaut vraiment mieux utiliser des variables, pour rendre le code plus lisible et pouvoir réutiliser ses styles ! La même chose en mieux :

```
impt=' \033[1;31;4m '
normal=' \033[0m '
printf "Ceci est vraiment ${impt} très important ${normal} et
      ceci l'est beaucoup moins"
```

Passons aux fameux codes :

0	texte normal, sans style particulier ni couleur
1	gras, brillant
2	de faible intensité
4	souligné
5	clignotant (ne fonctionne pas sur tous les terminaux)
7	échanger les couleurs du texte et de l'arrière-plan
8	caché
39	couleur par défaut du texte
49	couleur par défaut de l'arrière-plan
30–37 et 90–97	des couleurs simples pour le texte
40–47 et 100–107	des couleurs simples pour l'arrière-plan
38;5;0–255	256 couleurs possibles pour le texte
48;5;0–255	256 couleurs possibles pour l'arrière-plan
38;2;0–255;0–255;0–255	couleurs pour le texte en RGB (<i>Red Green Blue</i>)
48;2;0–255;0–255;0–255	couleurs pour l'arrière-plan en RGB

Exercice 1 :

1. Afficher les couleurs simples pour le texte (codes de 30 à 37 puis de 90 à 97).
2. Afficher les couleurs simples pour le fond (codes de 40 à 47 et de 100 à 107).
3. Créer un style `colere` : gras, rouge, sur fond vert, puis le tester.
4. Créer un style `timide` : de faible intensité, bleu clair, puis le tester.
5. Créer un script qui affiche sous forme de tableaux les couleurs de texte (codes de 38;5;0 à 38;5;255) et d'arrière plan (codes de 48;5;0 à 48;5;255) du mode « 256 couleurs ».

--- * ---

L'invite de commande (*prompt*) principale du shell est contrôlée par la variable d'environnement `PS1`.

Exercice 2 :

1. Sauvegarder l'ancienne valeur de `PS1` dans une variable `oPS1`.
2. Mettre la chaîne vide dans `PS1`. Que se passe-t-il ?
3. Mettre uniquement le symbole `$` dans `PS1`.
4. Mettre votre nom d'utilisateur en rouge souligné dans `PS1`.
5. Afficher l'ancien contenu de `PS1` et aller voir la page https://www.gnu.org/software/bash/manual/html_node/Controlling-the-Prompt.html pour comprendre `PS1` dans GNU `bash`
6. S'amuser à personnaliser son invite de commande. Pour rendre la modification permanente : `export PS1='...'` dans `~/.bashrc`.

--- * ---