Divers

- Ajouter des couleurs à ses scripts shell

Ajouter des couleurs à ses scripts shell

Repris de https://stackoverflow.com/a/28938235.

Parce que la vie est plus agréable avec des couleurs, j'aime bien agrémenter mes scripts avec des couleurs.

Outre le fait que ce soit plus joli, cela permet aussi de rendre plus lisible la sortie d'un script (un texte en rouge, on se doute que tout ne s'est pas bien passé avant même de lire le texte).

Je note donc ici les codes pour écrire en couleur dans des scripts shell, en tant que pense-bête.

Les codes

Remise à zéro du formatage

```
NC='\033[0m'
```

Couleurs de base

```
BLACK='\033[0;30m'
RED='\033[0;31m'
GREEN='\033[0;32m'
YELLOW='\033[0;33m'
BLUE='\033[0;34m'
PURPLE='\033[0;35m'
CYAN='\033[0;36m'
WHITE='\033[0;37m'
```

En couleur et en gras

En gras, c'est *bold* en anglais, d'où le préfixe `B`.

```
BBLACK='\033[1;30m'
BRED='\033[1;31m'
BGREEN='\033[1;32m'
BYELLOW='\033[1;33m'
BBLUE='\033[1;34m'
BPURPLE='\033[1;35m'
BCYAN='\033[1;36m'
BWHITE='\033[1;37m'
```

Utilisation

```
RED='\033[0;31m'
NC='\033[0m'
echo -e "${RED}Hello world${NC}"
```

On notera ici trois choses :

1. l'encadrement de la variable par des accolades, pour la séparer du texte (sinon le shell essayerait d'interpréter la variable `$REDHello`, qui n'existe pas) ;
2. l'utilisation de l'option `-e` d'echo : selon le shell utilisé, le comportement sera différent (les shells bash, dash, zsh (et les autres aussi, sans doute) utilisent leur propre commande `echo`). L'option `-e` est nécessaire pour activer l'interprétation des backslash pour `/usr/bin/echo`, pas pour le `echo` de zsh. Dans le doute, il faut utiliser cette option.
3. l'utilisation de la variable de remise à zéro à la fin de la phrase. Certains shells vont conserver le changement de formatage de texte après le `echo` (bash, dash), d'autres non (zsh). Encore une fois, dans le doute, on remet le formatage à zéro.

On m'a signalé que `printf` était plus standardisé, donc au comportement plus constant entre les shells. Avec `printf`, ça donne :

```
RED='\033[0;31m'
NC='\033[0m'
printf "${RED}Hello world${NC}\n"
```

À noter, le `\n` final, car `printf` ne fait pas de retour à la ligne automatiquement.