

# imagemagick

- Objet : imagemagick
- Niveau requis :  
[débutant, avisé](#)
- Commentaires : *La retouche d'images en ligne de commande.*
- Débutant, à savoir : [Utiliser GNU/Linux en ligne de commande, tout commence là !](#) 😊
- Suivi :  
[à-placer](#)
  - Création par [smolski](#) 2010
  - Testé par [smolski](#) 2010
  - Suivi et augmenté: par [lagrenouille](#) mai 2023
- Commentaires sur le forum : [Lien vers le forum concernant ce tuto](#)<sup>1)</sup>

## Introduction

ImageMagick est un puissant outil en ligne de commandes pour la manipulation d'images. Il permet de rendre une quantité incroyable de services les plus variés, et peut servir de moteur à tout type d'interface graphique qui choisirait de s'appuyer sur lui pour devenir un nouveau The Gimp. Mais, même sans interface, ImageMagick n'en reste pas moins un formidable outil à utiliser, qui vous évitera de lancer des logiciels de retouche d'image complets pour réaliser de très simples et basiques opérations.

Grâce à Imagemagick, vous disposerez notamment des fonctionnalités suivantes :

- convertir une image dans un autre format ;
- redimensionner, déformer, effectuer des rotations, atténuer les couleurs, ajouter des effets spéciaux ;
- créer un montage d'images et en faire une animation ;
- créer une nouvelle image, composée de plusieurs autres images ;
- dessiner ou écrire sur une image.

Une fois installé, ImageMagick se pilote très facilement grâce à une série de lignes de commandes. Vous trouverez ci-après, quelques exemples de commandes usuelles.

## Commande magick

En 2019, les paquets Debian pour installer ImageMagick, n'offrent pas la dernière version 7 qui semble contenir une nouvelle commande "magick" ( une sorte de raccourci pour les autres commandes ) que vous pourriez voir sur d'autres tutos, ou sur [la doc officielle](#). Elle n'est pas nécessaire, et le tuto ci dessous fonctionne toujours :)

## Images d'illustration pour ce tuto



On demande des volontaires pour nous placer des illustrations réalisées (avant → après si nécessaire) !

Pour les rédacteurs, voir ce tuto pour insérer ses images : :

- [http://wiki.debian-facile.org/insertion\\_image](http://wiki.debian-facile.org/insertion_image)

## Installation

Dans un terminal root (voir [SU](#)), comme ça :

```
apt-get update && apt-get install imagemagick
```

Trôôôp, non ? 😊



### ATTENTION Grave !

N'exécutez les commandes qui suivent que sur **des images de sauvegarde** et non sur les originaux !



## Convertir

Les formats acceptés en lecture ou en écriture par ImageMagick sont très nombreux. Il paraît donc logique qu'il soit utilisé en ligne de commandes pour convertir un format d'image en un autre. Par exemple, la ligne suivante permet de convertir une image .png en image .gif :

```
mogrify -format png image.gif
```

ou encore

```
convert image.gif image.png
```

À noter que l'opération n'efface pas l'image d'origine. Il est possible de réaliser la même opération par lots :

```
mogrify -format png *.jpg
```

On peut spécifier une résolution particulière de l'image avec l'option `density`. Plus la résolution est élevée, plus la qualité est élevée (et plus le fichier est lourd). Cette option est particulièrement utile lorsque l'on veut convertir un fichier pdf en image.

```
convert -density 300x300 monFichier.pdf monFichier.png
```

Dans ce cas, ImageMagick va convertir chaque image .jpg en .png tout en conservant le nom de base.

## Animation GIF

1. Je pars d'une vidéo nommée film.mkv dont je sectionne les 4 premières secondes avec [ffmpeg](#) :

```
ffmpeg -ss 00:00:00 -t 00:0:07 -i film.mkv -c copy cat.mkv
```

2. Puis convertir les images de cette mini video en fichiers PNG :

```
ffmpeg -i cat.mkv cat%d.png
```

3. Supprimer les images en trop pour conserver une petite séquence d'une vingtaine d'images, puis les réduire afin d'obtenir une taille vignette :

```
mogrify -resize x120 *.png
```

4. Et créer la séquence par exemple en 25 images seconde :

```
convert -delay 25 -loop 0 *.png cat.gif
```

Résultat : 

Source de ce tuto :

- [http://blog.dorian-depriester.fr/linux/creer-et-editer-des-gif-animes-avec-imagemagick#Convertir\\_un\\_ensemble\\_de\\_fichiers\\_image\\_en\\_GIF\\_anime](http://blog.dorian-depriester.fr/linux/creer-et-editer-des-gif-animes-avec-imagemagick#Convertir_un_ensemble_de_fichiers_image_en_GIF_anime)

## Redimensionnement d'images

Un autre usage courant consiste à redimensionner une image, grâce à l'option `-resize`. Vous pouvez alors

- spécifier un redimensionnement en pourcentage (`-resize 125%` ou `-resize 75%`),
- spécifier une nouvelle largeur en pixels (`-resize 640`), une nouvelle hauteur (`-resize x480`) ou les deux à la fois (`-resize 640x480`).

Par exemple, pour une réduction uniforme à 80% de l'image originale, vous utiliseriez :

```
convert -resize '80%' image_entrée.png image_sortie.png
```

et pour obtenir une image redimensionnée uniformément dont la nouvelle largeur serait 480 pixels :

```
convert -resize x480 image_entrée.png image_sortie.png
```

Bien sûr, vous pouvez à nouveau utiliser la commande `mogrify` pour redimensionner un lot d'images. Cette commande va redimensionner et écraser les images `.png` par la nouvelle version, de plus petite dimension :

```
mogrify -resize x480 *.png
```

Une dernière astuce très pratique consiste à ne redimensionner que les images qui sont plus grandes que la taille souhaitée en utilisant l'option `>`.  
Par exemple, la commande qui suit :

```
mogrify -resize x480> *.png
```

Ne redimensionnera l'image que si sa hauteur est plus grande que 480 pixels, ce qui fait qu'une image faisant, par exemple, 200×200 pixels sera ignorée par la commande.

## Création de miniatures

Supposons que vous avez créé un répertoire de miniatures dans le même répertoire que les images à traiter.

Vous pouvez créer une miniature pour chaque image et la placer dans le répertoire approprié, grâce à une commande unique composée des options `-path` (pour stocker le résultat de la commande dans un autre endroit ; ne marche qu'avec les versions récentes d'ImageMagick) et `-thumbnail` (pour générer les miniatures) :

```
mogrify -path miniatures -thumbnail 125x125 *.png
```

Vous noterez que l'option `-thumbnail` fonctionne de la même façon que l'option `-resize`. 😊

## Découper une image

Voir sur le forum : <https://debian-facile.org/viewtopic.php?pid=240982#p240982>

## Bordures

ImageMagick permet également d'ajouter des bordures colorées, biseautées et d'ajouter une ombre portée à vos photos.

- Pour créer un cadre biseauté, on utilise l'option `-raise`, qui permet d'éclaircir et d'assombrir les bords d'une image pour créer un effet 3D :

```
convert mon_image.jpg -raise 50x50 image_cadre.jpg
```

- On utilisera `-raise` ou `+raise` selon que l'on souhaite un effet relief ou creux. On notera que

l'option -raise utilise ici les pixels de l'image pour créer le cadre. Les dimensions de l'image restent donc identiques.

- Pour ajouter un cadre 3D, on utilise l'option -frame :

```
convert mon_image.jpg -mattecolor peru -frame 30x30+20+8 \  
image_framed2.jpg
```

- L'option -mattecolor introduit la couleur du cadre (la couleur peru est utilisée ici) ; l'option -frame permet de spécifier l'épaisseur du cadre (horizontale puis verticale) et l'épaisseur du biseau (externe puis interne).
- Pour ajouter une ombre portée, on utilise l'option -shadow :

```
convert mon_image.jpg \( +clone -background lightsteelblue \  
-shadow 60x3+30+30 \) +swap -background none \  
-mosaic image_shadow2.png
```

Remarquez ici l'emploi du caractère \ qui agit comme protection des parenthèses pour que celles-ci, considérées comme caractères spéciaux du shell, ne soient pas interprétées par ce dernier.

L'utilisation des parenthèses dans une commande Imagemagick permet de créer une image intermédiaire et d'introduire le résultat dans la suite de la ligne de commandes.

Le principe ici est de générer un clone de l'image, de le convertir en une image floue et colorée de façon semi-transparente, puis de le superposer à l'image originale via l'option -mosaic, selon le décalage spécifié (ici 30 px vers la droite et 30 px vers le bas).

- +clone permet de cloner la dernière image de la séquence (ici une seule image, l'image originale).
- -background définit la couleur de l'ombre (ici bleu acier) ; la deuxième mention -background, hors parenthèses, définit la couleur de l'arrière-plan du canevas sur lequel est construite la mosaïque (ici l'ensemble image + ombre), pas d'arrière-plan ici (valeur none).
- -shadow permet de définir une ombre portée ; elle admet 4 paramètres (dans cet ordre) : pourcentage d'opacité x sigma du flou x décalage vers la droite x décalage vers le bas
- +swap permet d'échanger la position (dans l'empilement) des deux dernières images d'une séquence (ici l'originale et son clone ombré).
- -mosaic crée une mosaïque à partir d'une séquence d'images.

Notez enfin que la création d'une ombre portée étend les dimensions de l'image originale.

- Pour encadrer la photo d'une bordure colorée, on peut utiliser l'option -draw (on considère ici une photo de 2048×1536 pixels) :

```
convert mon_image.jpg -fill none -stroke grey -strokewidth 59 \  
-draw 'rectangle 0,0 2047,1535' image_bordure.jpg
```

On trace en fait un rectangle autour de l'image :

- L'option -draw permet de dessiner une forme géométrique (ici un rectangle de coordonnées x0, y0 x1, y1 ; d'autres formes sont possibles).
- L'option -fill définit la couleur de remplissage du rectangle (ici pas de remplissage).
- L'option -stroke définit la couleur de contour du rectangle.
- L'épaisseur du contour est contrôlée par l'option -stroke width, qui fonctionne comme suit :

valeur de `strokewidth` = épaisseur de bordure  $\times 2 - 1$ . Pour une épaisseur de 30 pixels, nous avons donc spécifié `-strokewidth 59` ( $30 \times 2 - 1$ ).

Tout comme avec l'option `-raise`, l'image conserve ici ses dimensions initiales. En revanche, si l'on utilise l'option `-border`, les dimensions de la bordure viennent s'ajouter à celles de l'image initiale.

## draw

L'option `-draw` inscrit une bordure à l'*intérieur* de l'image, ce qui fait que cette image conserve toujours les mêmes dimensions.

## border

L'option `-border` ajoute une bordure *autour* de l'image.

```
convert mon_image.jpg -bordercolor teal -border 50 image_border.jpg
```

On peut ajouter des bordures à la bordure précédente, par exemple :

```
convert mon_image.jpg -bordercolor white -border 10 -bordercolor black \
  -border 20 -bordercolor gold -border 8 -bordercolor black -border 50 \
  -image_border.jpg
```

Ce qui se lit :

```
-bordercolor white -border 10 Ajoute une première bordure blanche de 10
pixel
-bordercolor black -border 20 -> Ajoute ensuite une bordure noir de 20 pixel
-bordercolor gold -border 8 -> Ajoute ensuite une bordure or de 8 pixel (le
liseret) au précédent
-bordercolor black -border 50 -> Ajoute enfin une bordure noir de 50 pixel
au précédent
```

Ce qui donne :



Et nous pouvons y ajouter une petit bord façon cadre en bois ainsi :

```
convert mon_image.jpg -mattecolor peru -frame 15x15+5+5 image_border.jpg
```

Ce qui donne :



## Légender une image

Il est possible d'incruster une légende directement dans l'image de votre choix. Toutefois, un mauvais choix de couleur de police par rapport au fond de l'image peut rendre la légende totalement illisible. Une astuce consiste alors à créer un bandeau de couleur qui va être intégré dans la photographie à légender.

La commande est toutefois complexe :

```
width=`identify -format %w [fichier_source]`convert -background  
'[couleur_fond]' -fill [couleur_texte] -gravity [alignement_horizontal] -  
size ${width}x[hauteur] caption:"[Légende]" +size [photo_source.jpg] +swap -  
gravity [alignement_vertical] -composite [fichier_cible]
```

Les paramètres à spécifier sont les suivants :

- [fichier\_source] : définit le nom et l'extension de l'image à légender, par exemple dscn0178.jpg.
- [couleur\_fond] : définit la couleur du bandeau qui sera superposé à l'image originale, par exemple '#0008' (entre guillemets) pour un simple gris.
- [couleur\_texte] : définit la couleur du texte de la légende, par exemple white pour un texte blanc.
- [alignement\_horizontal] : indique la position du texte sur la ligne de légende, par exemple center pour un texte centré sur l'image.
- [hauteur] : définit la hauteur d'une ligne, en pixels.
- [alignement\_vertical] : indique la position du texte sur l'image, par exemple au sud (south) de celle-ci.
- [fichier\_cible] : définit le nom et l'extension de l'image de sortie, permettant de ne pas écraser l'image d'origine.

On déduit dans un premier temps la largeur (width) du fichier source, afin de réutiliser cette information plus tard, et ensuite on utilise la commande convert d'ImageMagick pour générer le bandeau de couleur qui portera la légende. Le texte en lui-même peut bien sûr être personnalisé ; en effet, les paramètres -font et -pointsize optionnels permettent respectivement de déterminer la police et la taille de la légende, et sa couleur est, comme nous l'avons vu, définie par l'option -fill. Le texte est ensuite simplement positionné selon l'une des directions cardinales admissibles : NorthWest, North, NorthEast, West, Center, East, SouthWest, South, SouthEast. Dans vos essais, vous prendrez garde à ce que [hauteur] reste compatible avec le paramètre -pointsize éventuellement spécifié

## Lister les Polices ttf

Dans un [terminal user](#), utiliser la commande [locate](#) ainsi :

```
locate -r '\.ttf$'
```

[retour de la commande](#)

```
/home/mon_user/.fonts/F00.ttf  
/home/mon_user/Téléchargements/font/foo/F00.ttf
```

```
/usr/share/cups/fonts/FreeMono.ttf
/usr/share/cups/fonts/FreeMonoBold.ttf
/usr/share/cups/fonts/FreeMonoBoldOblique.ttf
/usr/share/cups/fonts/FreeMonoOblique.ttf
/usr/share/fonts/truetype/freefont/FreeMono.ttf
/usr/share/fonts/truetype/freefont/FreeMonoBold.ttf
etc...
```

Maintenant, pour utiliser une des polices **.ttf** disponibles, par exemple la police : *LiberationSerif-Italic.ttf* vous ajouter l'option `-font` suivi du chemin de cette police trouvé par la commande précédente ainsi :

```
width=`identify -format %w [fichier_source]`; convert -background
'[couleur_fond]' -font /usr/share/fonts/truetype/ttf-
liberation/LiberationSerif-Italic.ttf etc...
```

## Placement du texte

Pour placer le titrage en bas et au centre d'une photo, par exemple :

```
width=`identify -format %w source.jpg`; convert -geometry +0+0 \
-background 'none' -fill black -font /home/mon_user/polices/ARIALNI.ttf \
-gravity center -size ${width}x200 caption:"Test de titrage 2012" +size
source.jpg \
+swap -gravity south -composite source_resultat.jpg
```

Toutefois, avec cette disposition, le titrage est pile au bord de l'image, pour le distancer un peu de ce bord, nous le relevons de 20 pixel du bas en utilisant l'option `-geometry` ainsi :

```
width=`identify -format %w source.jpg`; convert -geometry +0+0 -background
'none' \
-fill black -font /home/mon_user/polices/ARIALNI.ttf -gravity center \
-size ${width}x200 caption:"Test de titrage 2012" +size source.jpg +swap \
-gravity south -geometry +0+20 -composite source_resultat.jpg
```

Pour un texte en *Arial* de couleur blanche (white), à l'est (east), en bas (south), de 50 pixel de haut et à 20 pixel des bords droits et bas, la ligne est :

```
width=`identify -format %w source.jpg`; convert -geometry +0+0 \
-background 'none' -fill white -font /home/mon_user/polices/ARIALNI.ttf \
-gravity east -size ${width}x50 caption:"Photo Joël Smolski" +size
source.jpg \
+swap -gravity south -geometry -20+20 -composite source_resultat.jpg
```

À noter l'utilisation en **négatif** des 20 pixel du bord dans l'option `geometry` : `-geometry -20+20`



**Nota :**

Pour le choix des couleurs, il est possible d'employer la notation Hexa(décimale), par exemple :

```
black = #000000
```

Il suffit de mettre cette notation entre quote ainsi :

```
'#000000'
```

Utiliser l'application **Agave** (dans les dépôts debian) pour identifier avec les couleurs. Hop ! 😊

**TP\_Titrage**

Nous allons légender la photo mis en exemple précédemment :

Pour le titre :

```
width=`identify -format %w image_border.jpg`; convert -geometry +0+0 \  
-background 'none' -fill gold -font /home/user/polices/BRUSHSCI.ttf -  
gravity center \  
-size ${width}x50 caption:"Le Mont du Chat" +size image_border.jpg +swap \  
-gravity south -geometry -0+30 -composite cadre_titre.jpg
```

**TP\_Signature**

Pour la signature :

```
width=`identify -format %w cadre_titre.jpg`; convert -geometry +0+0 \  
-background 'none' -fill gold -font /home/user/polices/arialbd.ttf -  
gravity east \  
-size ${width}x8 caption:"Photo Joel Smolski" +size cadre_titre.jpg +swap \  
\   
-gravity south -geometry -25+20 -composite cadre_titres.jpg
```

Et voilà, c'est fini Nini ! 😊

**Source :**

- [http://www.imagemagick.org/Usage/annotating/#text\\_position](http://www.imagemagick.org/Usage/annotating/#text_position)

## Sous-exposition

Corriger une photographie sous-exposée.

Si vous êtes aussi bon photographe que moi, de nombreuses photographies sous-exposées doivent occuper votre album photo. Heureusement, la commande `convert` nous apporte une correction simple des images, grâce à la fonction `-sigmoidal-contrast`, qui permet d'augmenter le contraste sans pour autant saturer les zones de lumière et les ombres. Elle admet deux paramètres, séparés par une virgule : le premier indique l'intensité de la correction de contraste (essayez de faibles valeurs, de l'ordre de 1 à 4, dans un premier temps, avant de vous hasarder à des valeurs plus élevées qui gâcheront probablement vos clichés) ; le deuxième (en %) permet de tirer la luminosité vers le haut (0%, blanc) ou vers le bas (100%, noir), par translation des tons moyens de la photographie vers le pourcentage spécifié. Par exemple :

```
convert [fichier_source] -sigmoidal-contrast 5,0% [fichier_cible]
```

Cela va augmenter sensiblement le contraste tout en tirant la luminosité vers le haut.

Quelques autres fonctions courantes

Il s'agit d'opérateurs appliqués à la commande `convert`.

## Paint

Cet opérateur permet de transformer une photographie en aplats aux contours diffus qui ne seront pas sans rappeler des coups de pinceau sur une toile de maître, d'où son nom. L'opérateur `-paint` admet un rayon en guise de valeur. Plus il est élevé, plus la peinture paraîtra grossière ; essayez des valeurs comprises entre 3 et 10 pour saisir l'influence du rayon... Exemple de commande :

```
convert [fichier_source] -paint 3 [fichier_cible]
```

## Colorspace

Cet opérateur permet de changer l'espace de couleur d'une image. Une application typique consiste à s'en servir pour transformer une image couleur en image noir et blanc. Il suffit en effet de spécifier à l'opérateur `-colorspace` la valeur `Gray`.

Par exemple :

## Negate

L'opérateur `-negate` va tout simplement réaliser l'inversion vidéo de la photographie. Par exemple :

Et cette dernière commande peut être appliquée à un seul canal de couleur, via l'option `-channel` comme ceci :

## Flip et Flop

Ces deux opérateurs sont très simples à mettre en oeuvre, car ils ne nécessitent aucune autre valeur pour fonctionner. Le premier, -flip, permet de retourner l'image de bas en haut, tandis que le second, -flop, permet de retourner l'image de gauche à droite, comme si elle se reflétait dans un miroir. Par exemple :

```
convert [fichier_source] -flip [fichier_cible]
```

et

```
convert [fichier_source] -flop [fichier_cible]
```

## Couleurs

### identify

Quelles couleurs utiliser dans les commandes ImageMagick ?

Pour en obtenir la liste complète (noms + codes RGB), entrez simplement la commande<sup>2)</sup> :

```
identify -list color | less
```

### transparent

Pour rendre une couleur transparente, par exemple le blanc :

```
convert -transparent "#ffffff" image_origine.png image_transformee.png
```

De **bendia** sur le forum, sympa ! 😊

## Tableau

Source :

- <http://www.imagemagick.org/script/color.php> Vous y trouverez l'avantage d'en visionner la couleur en plus ! 😊

## Capture d'écran

Pour effectuer une capture totale ou partielle de l'écran, on utilisera la commande import, comme suit : où N représente le nombre de secondes qui vont s'écouler avant la mise en route de la fonction capture.

```
import -pause N fichier_cible
```

Par exemple, vous souhaitez capturer une partie de votre écran. Le temps de bien organiser l'ensemble des fenêtres de l'écran et de n'afficher que ce qui vous intéresse, vous estimez le temps de pause à 10 secondes. Et vous souhaitez enregistrer la capture sous le nom de capture.png :

```
import -pause 10 capture.png
```

10 secondes après avoir entré cette commande, votre pointeur de souris se transforme en petite croix : vous n'avez qu'à sélectionner la portion d'écran que vous souhaitez capturer (par cliquer/glisser). L'image sera enregistrée à l'endroit où vous saisissez la ligne de commandes.

## Automatisation



### ATTENTION Grave !

N'exécutez ces commandes que sur **des images de sauvegarde** et non sur les originaux !

Recompresser une image :

```
mogrify -quality 60 mon_image.jpg
```

Ou encore, pour envoyer par mail :

```
mogrify -quality 60 -size 800 mon_image.jpg
```

Évidemment, pour toute une arborescence (fichier, répertoires et sous répertoires) utilisez [find](#) :

```
find -iname *.jpg -exec mogrify -quality 60 -size 800 {} \+
```

## TP convert

- [TP signature renommage titrage bordures avec convert](#)

## Liens et remerciements

Le site de l'utilisation détaillée d'**imagemagick** :

- <http://www.imagemagick.org/script/index.php>
- <http://www.imagemagick.org/Usage/>

Et là un lien en français des options dans le détail. Attention, c'est rempli ! 😊

- [http://softlibre.gloobe.org/imagemagick/doc\\_start](http://softlibre.gloobe.org/imagemagick/doc_start)

Autre lien vers le forum :

- <https://debian-facile.org/viewtopic.php?pid=374286#p374286>

## Logiciel analogue

- **GRAPHICSMAGICK** Une variante, performante sur les très grosses images.

Pour les allergiques à la ligne de commande, **Converseen** est un outil graphique de convert.

```
apt install converseen
```

Converseen est un processeur d'images par lots multiplateforme gratuit qui vous permet de convertir, redimensionner, faire pivoter et retourner un nombre infini d'images en un clic de souris, qui, grâce à ImageMagick , prend en charge plus de 100 formats d'image, il utilise la bibliothèque de manipulation d'images gratuite ImageMagick.

1)

N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !

2)

Voir : [less](#) et [pipe](#)

From:

<http://debian-facile.org/> - **Documentation - Wiki**

Permanent link:

<http://debian-facile.org/doc:media:imagemagick>

Last update: **15/05/2023 18:06**

