

Accueil > Cours > Reprenez le contrôle à l'aide de Linux ! > TP : générateur de galerie d'images

Reprenez le contrôle à l'aide de Linux !

🕒 30 heures 📶 Facile

Mis à jour le 29/06/2021



Le contenu de ce cours n'est plus à jour

Nous avons archivé ce cours et n'actualiserons plus son contenu.

Accédez au contenu le plus récent en découvrant ce cours :



SYSTÈMES & RÉSEAUX

Initiez-vous à Linux

📶 Easy 🕒 8 heures

Dans ce cours débutant, découvrez Linux : un système d'exploitation gratuit et fascinant qui vous donnera un contrôle sans précédent sur votre ordinateur ! Créé par des passionnés d'informatique, Linux est un vecteur important de la philosophie du libre et l'alternative parfaite à Windows ou macOS.

[VOIR LE NOUVEAU COURS](#)

TP : générateur de galerie d'images

L'intérêt du bash ne commence à se faire sentir que lorsque l'on code de vrais scripts, alors... il est grand temps de pratiquer !

Dans ce TP, vous allez devoir réutiliser un peu tout ce que vous avez appris jusqu'ici sur bash et sur Linux en général. N'oubliez pas que dans les scripts bash vous pouvez réutiliser toutes les commandes de la console que vous connaissez : `ls`, `grep`, `cut`, `sort`, les flux... allez-y, tous les coups sont permis. Vous risquez même d'avoir à lire le manuel pour trouver quelques paramètres !

Votre objectif est de créer une page web présentant une galerie d'images en fonction des fichiers présents dans un dossier. Plus facile à dire qu'à faire, car vous allez voir qu'il y a là un vrai défi. Bonne chance à tous.

Objectifs




Nous souhaitons réaliser dans ce TP un **générateur de galerie d'images** en bash.

Le script s'appellera `galerie.sh`. Pour sa première version, il faudra le placer dans un dossier contenant des images ; il générera des miniatures à partir de ces dernières et un fichier HTML présentant toutes les images du dossier.

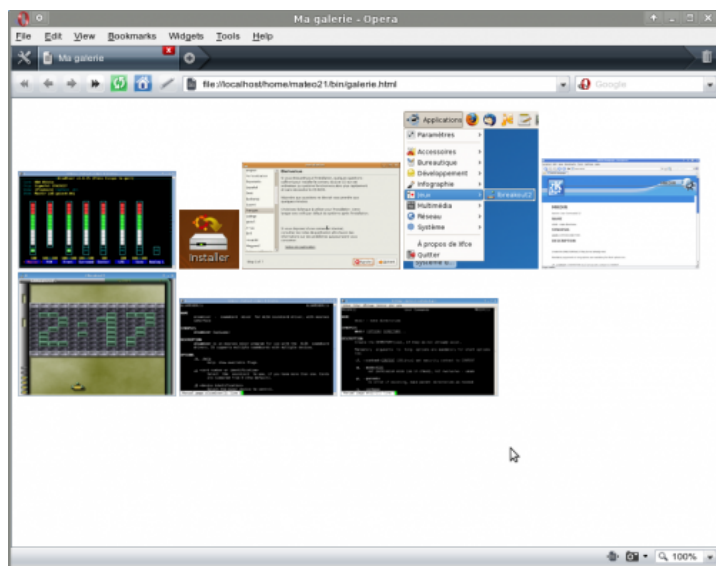
Concrètement, le script devra donc :

- créer une miniature de chaque image du dossier ;
- générer un fichier HTML et y insérer ces miniatures ;
- faire un lien vers les images en taille originale.

 Pour réaliser ce script, il est recommandé de connaître un petit peu le HTML. Vous avez **un cours à votre disposition sur le Site du Zéro**. C'est assez facile : ceux qui ne connaissent pas ne devraient pas y consacrer beaucoup de temps, d'autant plus qu'il suffit de lire la première partie seulement de ce cours pour faire ce TP.

Le rendu final

La page web que vous devez arriver à générer devrait ressembler à la figure suivante.



Bien entendu, c'est un exemple **minimal**. Il est possible de faire quelque chose de beaucoup plus joli : commencez déjà par faire en sorte que cela fonctionne, vous enjoliverez après. 😊

Le code HTML de base

Pour vous aider (enfin surtout pour ceux d'entre vous qui ne sont pas très à l'aise en HTML 😊), je vous propose de partir du code (minimaliste) suivant :

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" >
  <head>
    <title>Ma galerie</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <style type="text/css">
      a img { border:0; }
    </style>
  </head>
  <body>
    <p>

<a href="alsamixer.png"></a>
<a href="icone_installer.png"> 
</a>
<a href="installation.png"></a>
    </p>
  </body>
</html>
    
```

Vous pourrez réutiliser le début et la fin de ce code source pour chaque fichier HTML de galerie que vous créerez. Par contre, au milieu (les

images `` et les liens `<a>`), il faudra adapter automatiquement le code en fonction des images présentes dans le dossier.

Vous afficherez des miniatures sur la page et ferez un lien vers leur version agrandie.

Comment générer des miniatures d'images ?

Bonne question. On n'a pas appris à faire cela en ligne de commande, tout simplement parce que ce n'est pas ce que j'appelle une commande « de base » de Linux. Néanmoins, vous devriez avoir le programme `convert`, capable d'effectuer de nombreuses opérations sur des images. À vous d'afficher le manuel et de comprendre comment on l'utilise. 😊

Bon : je vous aide quand même un peu parce que cette commande a énormément de paramètres. « Miniature » en anglais se dit « *thumbnail* ».

Vous apprécierez probablement [l'aide en ligne](#), plus étoffée et plus lisible peut-être que le man.

Les paramètres

Notre programme devra accepter un paramètre optionnel : le nom du fichier HTML à générer. S'il n'est pas présent, on générera un fichier `galerie.html` par défaut.

Solution



L'heure est venue de passer à la correction !

```
#!/bin/bash

# Vérification des paramètres
# S'ils sont absents, on met une valeur par défaut

if [ -z $1 ]
then
    sortie='galerie.html'
else
    sortie=$1
fi

# Préparation des fichiers et dossiers

echo '' > $sortie

if [ ! -e miniatures ]
then
    mkdir miniatures
fi

# En-tête HTML

echo '<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" >
  <head>
    <title>Ma galerie</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <style type="text/css">
      a img { border:0; }
    </style>
  </head>
  <body><p>' >> $sortie

# Génération des miniatures et de la page

for image in `ls *.png *.jpg *.jpeg *.gif 2>/dev/null`
do
    convert $image -thumbnail '200x200' miniatures/$image
    echo '<a href="'$image'"> </a> '>> $sortie
```

```
done

# Pied de page HTML

echo '</p>
</body>
</html>' >> $sortie
```

Quelques commentaires sur le script :

- Il vérifie d'abord si un paramètre est présent. Si oui, il l'utilise comme nom de fichier de sortie, sinon il utilise `galerie.html` .
- On doit créer un fichier vide pour `galerie.html` . Normalement, on peut faire ça avec `touch` , mais si le fichier existe déjà, on veut le vider. On choisit donc de faire un `echo` vide dans ce fichier pour le vider.
- On crée le dossier qui accueillera les images miniatures s'il n'existe pas.
- On écrit l'en-tête HTML dans le fichier.
- On fait une boucle sur tous les fichiers de type image (*.png, *.jpg, etc.) qui existent dans le dossier. Pour chacun d'eux, on crée une miniature d'une taille maximale de 200 x 200 pixels dans le sous-dossier `miniatures` . Le petit symbole « > » permet, comme indiqué dans la documentation du programme, de ne pas générer de miniature inutilement si l'image est de base plus petite que la taille des miniatures.
- On écrit dans la page web la balise qui affichera l'image et on fait un lien vers la version agrandie.
- Enfin, on termine la page HTML en fermant les balises.

Je tiens à rappeler qu'il n'y a pas une seule façon de réaliser ce script mais plusieurs. Je vous ai présenté la mienne et, bien qu'elle fonctionne, je vous préviens que l'on peut largement l'améliorer. Je vous propose d'ailleurs des pistes pour améliorer ce script.

Améliorations

Comme je le disais plus tôt, le script que je vous ai proposé de faire est minimal. Le but était d'avoir à réaliser un script accessible à tous et qui produise un résultat intéressant.

Si vous voulez l'améliorer, les pistes ne manquent pas. En voici quelques-unes.

- Améliorer le design de la galerie avec un peu de CSS.
- Permettre de choisir le dossier contenant les images dont on veut générer une galerie. Actuellement, il faut que `galerie.sh` soit dans le bon dossier pour que cela fonctionne !
- Utiliser un paramètre pour définir la taille des miniatures à générer.
- Afficher le nom de l'image sous chaque image.
- Afficher d'autres informations, comme les dimensions de l'image originale, sous chaque miniature. Il faudra faire appel à l'outil `convert` pour obtenir ces informations.
- Afficher la date de dernière modification sous chaque image.

Pour certaines de ces améliorations, il faudra vous renseigner dans le manuel voire poser des questions sur les forums.

Ne vous arrêtez pas en si bon chemin ! Cherchez, cherchez et cherchez encore ! Vous allez vous habituer à faire des recherches et vous deviendrez ainsi beaucoup plus autonomes. 😊

Le cours se termine ici. Bien entendu, il aurait été possible de le compléter (presque à l'infini !) mais je ne dispose pas d'autant de temps. 😊

Néanmoins, avec l'ensemble de ce cours vous avez désormais je l'espère une introduction à Linux enfin accessible aux débutants. Il reste bien des choses à découvrir, je vous invite à regarder les [tutoriels de la section Linux](#) du site pour en apprendre plus si vous le souhaitez.

Bonne continuation ! 😊

J'AI TERMINÉ CE CHAPITRE ET JE PASSE AU SUIVANT



LES FONCTIONS

QUIZ : QUIZ 5



Le professeur



Mathieu Nebra

Entrepreneur à plein temps, auteur à plein temps et co-fondateur d'OpenClassrooms :o)

Découvrez aussi ce cours en...



Livre



PDF

OPENCLASSROOMS

Qui sommes-nous ?

Financements

Expérience de formation

Forum

Blog [↗](#)

Presse [↗](#)

OPPORTUNITÉS

Nous rejoindre [↗](#)

Devenir mentor [↗](#)

Devenir coach carrière [↗](#)

AIDE



FAQ

POUR LES ENTREPRISES

Former et recruter

EN PLUS

Boutique [↗](#)

Mentions légales

Conditions générales d'utilisation

Politique de protection des données personnelles

Cookies

Accessibilité

 Français ▼

