

MINI-PROJET
Galerie photos en HTML
Novembre-décembre 2015

Vous allez réaliser un mini projet en shell de Bourne « **sh** ». Ce projet sera à rendre avant le **dimanche 13 décembre 2015 à 23h55** sur moodle.

Vous devrez fournir les scripts et un mini rapport (5 pages maximum) décrivant ces scripts ainsi que des exemples d'utilisation. Vous rendrez ces documents sous forme d'une archive (tar) contenant les fichiers sources des scripts et le rapport au format pdf.

De plus, vous devez vous assurer que vos scripts suivent bien la syntaxe demandée (nom du script, paramètres...) et sont compatibles avec le script de vérification « **verif.sh** » disponible sur moodle.

Sujet

Dans ce projet, vous allez réaliser un ensemble de scripts permettant de générer une galerie photos à l'aide de fichiers HTML.

L'Hypertext Markup Language, généralement abrégé HTML, est le format de données conçu pour représenter les pages Web. C'est un langage de balisage permettant d'écrire des documents hypertextes (document contenant des liens vers d'autres documents). Les documents (fichiers) HTML peuvent être affichés par un navigateur Web (firefox, chrome, safari...).

Nous vous proposons d'écrire des scripts permettant de gérer une galerie photos. Une galerie photos sera représentée par un fichier HTML qui pourra être affiché directement par un navigateur Web. Les fichiers HTML (de type galerie photos) devront être dans un répertoire contenant les sous-répertoires `images` et `styles`. Ces deux sous-répertoires ainsi que leur contenu sont disponibles sur moodle.

Vous allez donc écrire plusieurs scripts de gestion de galerie photos, certains pourront être utilisés pour en réaliser d'autres plus complexes.

Précision : chaque script devra vérifier la validité des paramètres qui lui sont passés, l'accessibilité des fichiers et répertoires qu'il manipule. À chaque appel d'une commande dans un script, il faudra gérer les erreurs éventuelles et afficher des messages clairs pour l'utilisateur.

Vos scripts devront *fonctionner sur la machine **azteca*** accessible de l'extérieur (cf. sujet TDM 1 pour plus de détails). De plus, vous *devez obligatoirement tester vos scripts à l'aide du script de vérification* « **verif.sh** » disponible sur moodle.

Partie 1 : Galerie photos sans fichiers images miniatures

Une miniature est une version d'une image dont la taille est réduite par rapport à l'original. L'utilité principale des miniatures est similaire à celle d'un index ou d'une table des matières pour du texte : permettre la construction de galeries d'images, afin d'en visualiser un plus grand nombre à la fois, et ainsi de disposer plus facilement d'une vue d'ensemble, éventuellement afin d'en choisir une.

Dans cette partie, les images seront redimensionnées directement dans le code HTML.

Vous devez écrire les scripts suivants :

1. `debut_html.sh [titre]`

Ce script affiche sur `stdout` le code HTML ci-dessous. L'argument `titre` correspond au titre de la page HTML. Sans paramètre, le titre par défaut est : « Galerie Photos ». Le code HTML à afficher est :

```
<!doctype html>
<html>
<head>
```

```

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>TITRE</title>
<link rel="stylesheet" href="styles/styles.css" media="all">
</head>
<body>
<div class="container">

```

En remplaçant TITRE par la valeur de l'argument titre ou « Galerie Photos » si aucun argument n'est passé lors de l'appel de la commande.

Attention : le code affiché sur `stdout` doit être strictement identique au code ci-dessus (à l'exception du nombre d'espaces).

La commande :

```
debut_html.sh "Mes photos"
```

Affichera :

```

<!doctype html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>Mes photos</title>
  <link rel="stylesheet" href="styles/styles.css" media="all">
</head>
<body>
<div class="container">

```

2. fin_html.sh

Ce script affiche sur `stdout` le code HTML ci-dessous.

```

</div>
</body>
</html>

```

Attention : le code affiché sur `stdout` doit être strictement identique au code ci-dessus (à l'exception du nombre d'espaces).

3. liste_photos.sh fichier[...]

Ce script prend en entrée une liste de désignations de fichiers et affiche sur `stdout` un code HTML qui permettra de visualiser cette liste.

Uniquement les désignations des fichiers correspondant à un fichier image de type JPEG (Joint Photographic Experts Group), GIF (Graphics Interchange Format) ou PNG (Portable Network Graphics) seront traitées. Nous considérerons que l'extension des fichiers permet de connaître leur type. Les fichiers JPEG ont pour extension « .jpg » ou « .jpeg » (chaque lettre pourra être une majuscule ou bien une minuscule). Les fichiers GIF auront pour extension « .gif » (chaque lettre pourra être en majuscules ou en minuscules). Les fichiers PNG ont pour extension « .png » (chaque lettre pourra être en majuscules ou en minuscules).

Pour chaque fichier image, le code à afficher est :

```

<div class="galleryItem">
  <a href="CHEMIN/NOMFICHIER" target="_blank">
    </a>
    <h3>NOMFICHIER</h3>
    <p>CHEMIN/NOMFICHIER</p>
</div>

```

avec CHEMIN/NOMFICHIER la désignation du fichier image ayant pour nom NOMFICHIER et pour chemin CHEMIN.

Attention : le code affiché sur `stdout` doit être strictement identique au code ci-dessus (à l'exception du nombre d'espaces). Un message d'erreur devra être affiché sur `stderr` pour chaque fichier (de la liste) ne

correspondant pas à un fichier image valide (Joint Photographic Experts Group, Graphics Interchange Format ou Portable Network Graphics).

Par exemple :

La commande :

```
liste_photos.sh photos/Arbres.JPG photos/Souvenirs.jpeg photos/Makefile.am
```

Affichera sur stdout :

```
<div class="galleryItem">
  <a href="photos/Arbres.JPG" target="_blank">
</a>
  <h3>Arbres.JPG</h3>
  <p>photos/Arbres.JPG</p>
</div>
<div class="galleryItem">
  <a href="photos/Souvenirs.jpeg" target="_blank">
</a>
  <h3>Souvenirs.jpeg</h3>
  <p>photos/Souvenirs.jpeg</p>
</div>
```

Affichera sur stderr :

```
photos/Makefile.am n'est pas un fichier image valide (JPEG, GIF ou PNG) !
```

4. `liste_photos_html.sh` titre fichier_destination_html fichier_source[...]

Ce script prend en entrée le titre de la page Web titre, la désignation du fichier HTML et une liste de désignation de fichiers. Il enregistre dans le fichier fichier_destination_html le code HTML complet d'une page Web proposant une visualisation de la liste de fichiers donnée en argument.

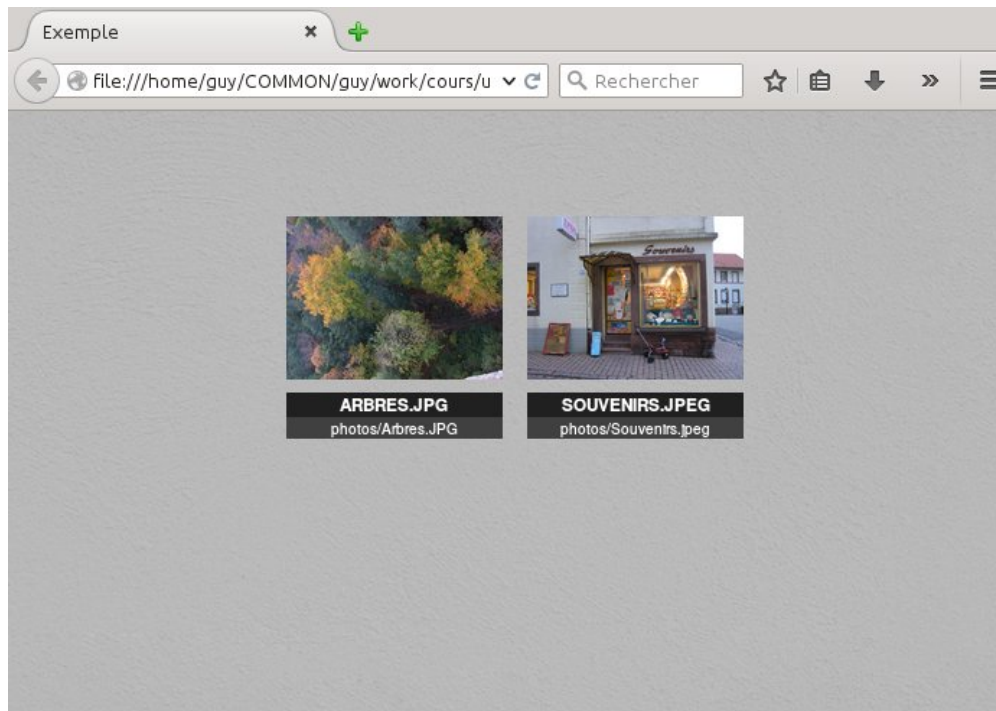
Par exemple :

```
liste_photos_html.sh Exemple f.html photos/Arbres.JPG photos/Souvenirs.jpeg
```

Écrira dans le fichier `f.html` le code suivant :

```
<!doctype html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>Exemple</title>
  <link rel="stylesheet" href="styles/styles.css" media="all">
</head>
<body>
<div class="container">
<div class="galleryItem">
  <a href="photos/Arbres.JPG" target="_blank">
</a>
  <h3>Arbres.JPG</h3>
  <p>photos/Arbres.JPG</p>
</div>
<div class="galleryItem">
  <a href="photos/Souvenirs.jpeg" target="_blank">
</a>
  <h3>Souvenirs.jpeg</h3>
  <p>photos/Souvenirs.jpeg</p>
</div>
</div>
</body>
</html>
```

Le fichier `f.html` pourra être visualisé par un navigateur de la manière suivante :



Attention : Si le fichier `fichier_destination.html` n'existe pas il devra être créé. S'il existe, il devra être effacé avant d'être rempli par le nouveau code HTML. Un message d'erreur devra être affiché sur `stderr` pour chaque fichier (de la liste) ne correspondant pas à un fichier image valide (Joint Photographic Experts Group, Graphics Interchange Format ou Portable Network Graphics).

Partie 2 : Galerie photos avec fichiers images miniatures

Afin d'alléger le chargement des photos par le navigateur, nous vous proposons dans cette partie d'utiliser des fichiers images miniatures (fichiers d'images réduites).

Vous devez écrire les scripts suivants :

5. `miniatures.sh repertoire_source repertoire_destination`

Ce script crée dans le répertoire `repertoire_destination` un fichier image miniature pour chaque fichier image (au format JPEG ou GIF ou PNG) de l'arborescence de racine `repertoire_source`.

Les fichiers images miniatures seront enregistrés directement dans le répertoire `repertoire_destination`, et seront nommés par un numéro¹ suivi de l'extension `.jpg` comme par exemple : `1.jpg`, `2.jpg`, etc.

De plus, le script devra sauvegarder dans un fichier texte `index_miniatures.txt` situé dans le répertoire `repertoire_destination`, le lien entre les fichiers images miniatures et les fichiers images.

Le fichier `index_miniatures.txt` contient sur chaque ligne la désignation d'un fichier image et la désignation du fichier image miniature qui lui correspond et éventuellement la date et l'heure de prise de vue.

Le format d'une ligne est :

- Dans le cas où la date et l'heure de prise de vue sont disponibles dans le fichier image :

`désignation_fichier_miniaature;désignation_fichier_image;JJ:MM:AAAA;hh:mm:ss`
avec `JJ` le jour, `MM` le mois, `AAAA` l'année, `hh` l'heure, `mm` les minutes et `ss` les secondes. `JJ`, `MM`, `AAAA`, `hh`, `mm`, `ss` sont des valeurs numériques.

- Si la date de prise de vue n'est pas présente alors la ligne aura la forme :

`désignation_fichier_miniaature;désignation_fichier_image`

Attention :

1. Un numéro différent par fichier miniature.

Pour générer un fichier image miniature, vous **devez** utiliser la commande² suivante :

```
mogrify -auto-orient -format jpg -write fichier_miniaature -thumbnail 150x150  
fichier_image
```

avec fichier_miniaature la désignation du fichier image miniature qui sera créé, et fichier_image la désignation du fichier image existant.

Pour connaître la date de prise de vue, vous **devez** utiliser la commande³ :

```
exiv2 -Pkv fichier_image
```

qui affiche sur stdout le nom suivi de la valeur des TAGs (étiquettes) disponibles du fichier image fichier_image.

Par exemple :

Supposons que le répertoire photos possède trois images Arbres.JPG, Fort_Foch.JPG et Fosse.JPG. De plus, supposons que le fichier Fort_Foch.JPG ne contient pas de date de prise de vue.

Alors la commande : `miniatures.sh photos minia` créera le fichier `index_miniaatures.txt` dont le contenu sera :

```
minia/1.jpg;photos/Arbres.JPG;03:11:2012;15:17:17  
minia/2.jpg;photos/Fort_Foch.JPG  
minia/3.jpg;photos/Fosse.JPG;03:11:2012;15:26:05
```

6. `miniatures_liste_photos.sh [-d] fichier_miniaatures`

Ce script affiche sur stdout un code HTML qui permettra de visualiser tous les fichiers images contenus dans le fichier index de miniatures fichier_miniaatures à l'aide de fichiers miniatures.

L'option `-d` ajoute en plus la date et l'heure de prise de vue, si celle-ci est disponible. Nous supposons que le format du fichier index de miniatures fichier_miniaatures correspond au format de la question 5.

- Sans l'option `-d`, pour chaque fichier image du fichier fichier_miniaatures (donc pour chaque ligne) le script affichera :

```
<div class="galleryItem">  
  <a href="CHEMIN/NOMFICHIERPHOTO" target="_blank">  
  </a>  
  <h3>NOMFICHIERPHOTO</h3>  
  <p>CHEMIN/NOMFICHIERPHOTO</p>  
</div>
```

CHEMIN/NOMFICHIERPHOTO dénote la désignation du fichier image ayant pour nom NOMFICHIERPHOTO et pour chemin CHEMIN. CHEMIN1/FICHIERMINIATURE correspond à la désignation du fichier image miniature de CHEMIN/NOMFICHIERPHOTO, ayant pour nom FICHIERMINIATURE et pour chemin CHEMIN1.

- Avec l'option `-d`, pour chaque fichier image du fichier fichier_miniaatures (donc pour chaque ligne) le script affichera :

```
<div class="galleryItem">  
  <a href="CHEMIN/NOMFICHIERPHOTO" target="_blank">  
  </a>  
  <h3>NOMFICHIERPHOTO</h3>  
  <p>CHEMIN/NOMFICHIERPHOTO</p>  
  <p>JJ:MM:AAAA<br/>hh:mm:ss</p>  
</div>
```

JJ:MM:AAAA et hh:mm:ss correspond au format date et heure du script `miniatures.sh`.

Attention : le code affiché sur stdout doit être strictement identique au code ci-dessus (à l'exception du nombre d'espaces).

Exemple :

Supposons que le répertoire photos possède trois images Arbres.JPG, Fort_Foch.JPG et Fosse.JPG. La commande

```
miniatures_liste_photos.sh minia/index_miniaatures.txt  
affichera sur stdout :
```

2. La commande `mogrify` fait partie de la suite logicielle ImageMagick (<http://www.imagemagick.org/script/index.php>).

3. <http://www.exiv2.org/>

```

<div class="galleryItem">
  <a href="photos/Arbres.JPG" target="_blank">
</a>
  <h3>Arbres.JPG</h3>
  <p>photos/Arbres.JPG</p>
</div>
<div class="galleryItem">
  <a href="photos/Fort_Foch.JPG" target="_blank">
</a>
  <h3>Fort_Foch.JPG</h3>
  <p>photos/Fort_Foch.JPG</p>
</div>
<div class="galleryItem">
  <a href="photos/Fosse.JPG" target="_blank">
</a>
  <h3>Fosse.JPG</h3>
  <p>photos/Fosse.JPG</p>
</div>

```

La commande

miniatures_liste_photos.sh -d minia/index_miniatures.txt
affichera sur stdout :

```

<div class="galleryItem">
  <a href="photos/Arbres.JPG" target="_blank">
</a>
  <h3>Arbres.JPG</h3>
  <p>photos/Arbres.JPG</p>
<p>03:11:2012<br/>15:17:17</p>
</div>
<div class="galleryItem">
  <a href="photos/Fort_Foch.JPG" target="_blank">
</a>
  <h3>Fort_Foch.JPG</h3>
  <p>photos/Fort_Foch.JPG</p>
</div>
<div class="galleryItem">
  <a href="photos/Fosse.JPG" target="_blank">
</a>
  <h3>Fosse.JPG</h3>
  <p>photos/Fosse.JPG</p>
<p>03:11:2012<br/>15:26:05</p>
</div>

```

7. `miniatures_liste_photos_html.sh` [-d] titre fichier_miniatures fichier_destination_html

Ce script prend en entrée le titre de la page Web titre, la désignation du fichier index des miniatures fichier_miniatures. Il enregistre dans le fichier fichier_destination_html le code HTML complet de la page Web proposant une visualisation des fichiers images présents dans le fichier index à partir des fichiers miniatures.

Par exemple :

```

miniatures_liste_photos_html.sh "Galerie Photos"
                                minia/index_miniatures.txt f1.html

```

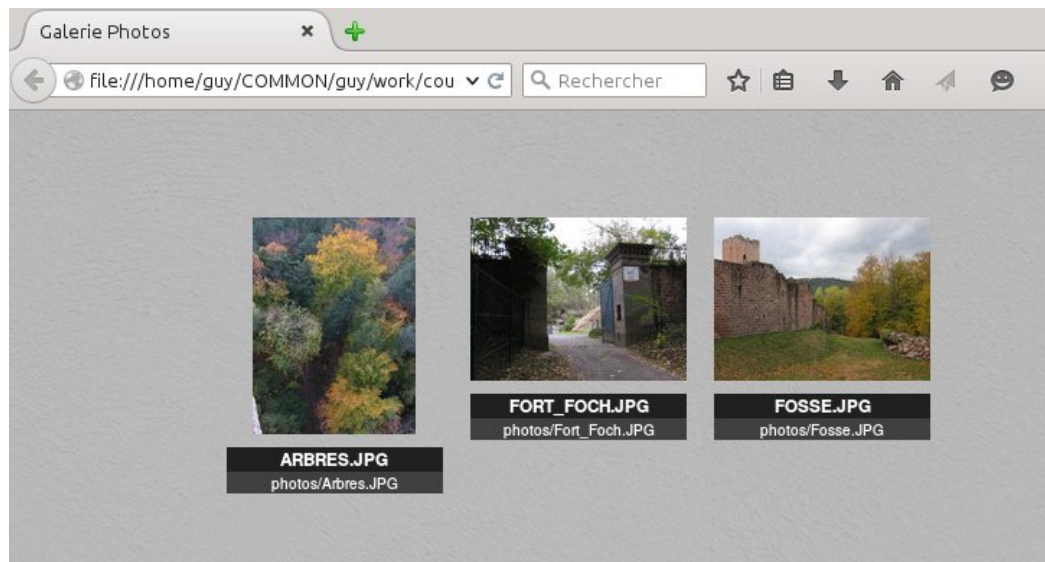
Écrira dans le fichier `f1.html` le code suivant :

```

<!doctype html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>Galerie Photos</title>
  <link rel="stylesheet" href="styles/styles.css" media="all">
</head>
<body>
<div class="container">
<div class="galleryItem">
  <a href="photos/Arbres.JPG" target="_blank">
</a>
  <h3>Arbres.JPG</h3>
  <p>photos/Arbres.JPG</p>
</div>
<div class="galleryItem">
  <a href="photos/Fort_Foch.JPG" target="_blank">
</a>
  <h3>Fort_Foch.JPG</h3>
  <p>photos/Fort_Foch.JPG</p>
</div>
<div class="galleryItem">
  <a href="photos/Fosse.JPG" target="_blank">
</a>
  <h3>Fosse.JPG</h3>
  <p>photos/Fosse.JPG</p>
</div>
</div>
</body>
</html>

```

Le fichier f1.html pourra être visualisé par un navigateur de la manière suivante :



L'appel suivant :

```

miniatures_liste_photos_html.sh -d "Galerie Photos"
                                minia/index_miniatures.txt f2.html

```

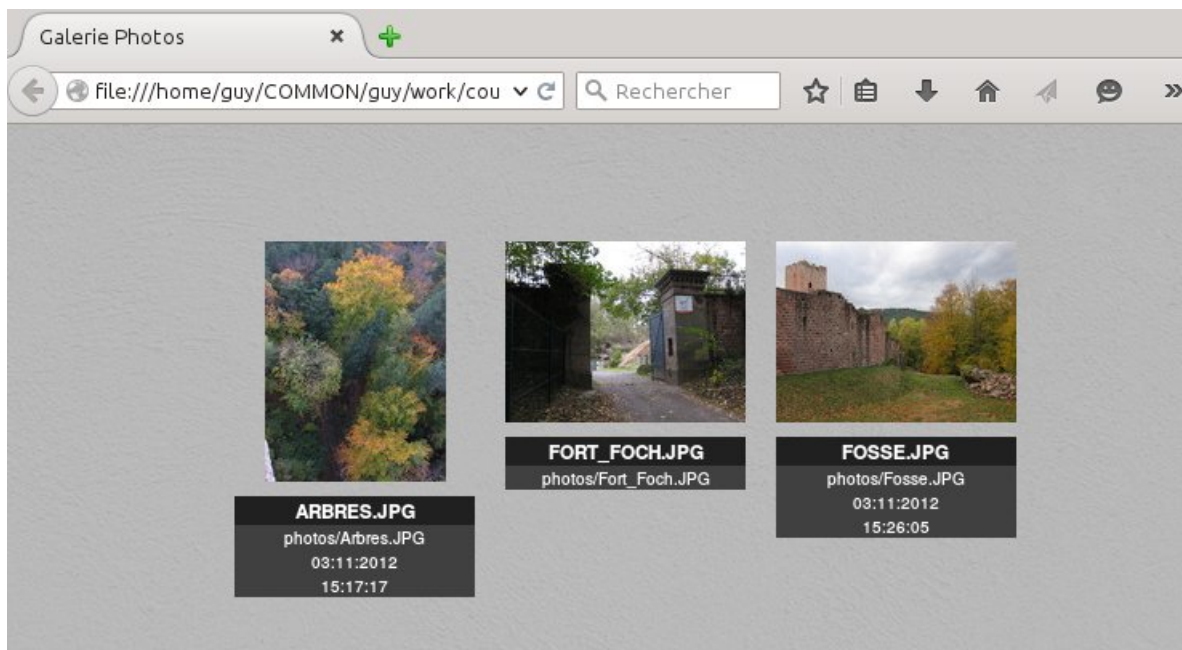
Écrira dans le fichier f2.html le code suivant :

```

<!doctype html>
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>Galerie Photos</title>
  <link rel="stylesheet" href="styles/styles.css" media="all">
</head>
<body>
<div class="container">
<div class="galleryItem">
  <a href="photos/Arbres.JPG" target="_blank">
</a>
  <h3>Arbres.JPG</h3>
  <p>photos/Arbres.JPG</p>
<p>03:11:2012<br/>15:17:17</p>
</div>
<div class="galleryItem">
  <a href="photos/Fort_Foch.JPG" target="_blank">
</a>
  <h3>Fort_Foch.JPG</h3>
  <p>photos/Fort_Foch.JPG</p>
</div>
<div class="galleryItem">
  <a href="photos/Fosse.JPG" target="_blank">
</a>
  <h3>Fosse.JPG</h3>
  <p>photos/Fosse.JPG</p>
<p>03:11:2012<br/>15:26:05</p>
</div>
</div>
</body>
</html>

```

Le fichier f2.html pourra être visualisé par un navigateur de la manière suivante :



Attention : Si le fichier fichier_destination n'existe pas il devra être créé. S'il existe, il devra être effacé avant d'être rempli par le nouveau code HTML.

