Gestion des images avec PHP

Le langage PHP intègre un ensemble de fonctions relatives aux images. Il est ainsi possible de créer ses propres images dans une page, mais aussi d'obtenir des informations sur des images existantes pour les redimensionner, les recadrer, etc.

Voici quelques exemples pour illustrer la puissance de PHP dans la gestion des images.

Mettre en ligne et copier une photo via un formulaire

Dans cet exemple tout simple, nous allons mettre en ligne une photo grâce à un formulaire.

photo.php

```
<!doctype html>
<html>
    <head>
        <meta charset="utf-8">
    </head>
    <body>
        <form method="post" enctype="multipart/form-data">
            <input type="file" name="photo">
            <input type="submit">
        </form>
    <?php
    if (isset($_FILES['photo']['tmp_name'])) {
        $retour = copy($_FILES['photo']['tmp_name'], $_FILES['photo']['name']);
        if($retour) {
            echo '<p>La photo a bien été envoyée.</p>';
            echo '<img src="' . $_FILES['photo']['name'] . '">';
        }
    }
    ?>
    </body>
</html>
```

Quelques explications

```
<form method="post" enctype="multipart/form-data">
```

L'attribut et la valeur *enctype='multipart/form-data'* sont ici indispensables car ils permettent au formulaire d'ajouter des données binaires, c'est à dire autre chose que du texte, comme des images, des musiques, ou autres fichiers informatiques.

```
$_FILES['photo']['tmp_name']
```

Lorsqu'on appuie sur le bouton *Envoyer*, la photo est copiée sur le serveur distant dans un dossier temporaire (nommé *tmp* sur *MAMP*) avec un nom temporaire. Cette opération peut d'ailleurs prendre un certain temps si la photo pèse plusieurs méga-octets, un peu comme quand vous envoyez une photo par email. Une fois le fichier mise en ligne sur le serveur, le nom de ce fichier est stocké dans la variable *$_FILES['photo']['tmp_name'].* Si vous faites un *echo* de cette variable, vous obtiendrez un nom comme : *fsdLDEKfdsiuovxiZERvxD*

```
$_FILES['photo']['name']
```

Cette variable contient le nom d'origine du fichier mis en ligne, par exemple *fleur.jpg*

```
copy($_FILES['photo']['tmp_name'], $_FILES['photo']['name']);
```

La fonction PHP *copy(source,destination)* va dupliquer la photo mise en ligne dans le dossier *tmp* du serveur dans le dossier où se trouve la page *photo.php*.

Ajouter une photo via un formulaire et redimensionnement

Pour tester cette page, créez un dossier nommé *miniatures* à coté du fichier *photo.php*.

photo.php

```
<html>
	<body>
		<form method="post" enctype="multipart/form-data">
			<input type="file" name="photo">
			<input type="submit" value="retailler">
		</form>
		<?php
		if (isset($_FILES['photo']['tmp_name'])) {
			$taille = getimagesize($_FILES['photo']['tmp_name']);
			$largeur = $taille[0];
			$hauteur = $taille[1];
			$largeur_miniature = 300;
			$hauteur_miniature = $hauteur / $largeur * 300;
			$im = imagecreatefromjpeg($_FILES['photo']['tmp_name']);
			$im_miniature = imagecreatetruecolor($largeur_miniature
			, $hauteur_miniature);
			imagecopyresampled($im_miniature, $im, 0, 0, 0, 0, $largeur_miniature,
$hauteur_miniature, $largeur, $hauteur);
			imagejpeg($im_miniature, 'miniatures/'.$_FILES['photo']['name'], 90);
			echo '<img src="miniatures/' . $_FILES['photo']['name'] . '">';
		}
		?>
	</body>
</html>
```

Explications

`$taille = getimagesize($fichier_courant)`

Cette fonction retourne un tableau contenant des informations intéressantes :

- $taille[0] est la largeur de l'image
- $taille[1] est la hauteur de l'image
- $taille[2] donne un numéro correspondant au format de l'image (1 => GIF, 2 => JPEG, 3 => PNG)
- $taille[3] donne une chaine de caractère utilisable directement dans la balise <img> (base64)

```
$largeur_miniature = 300;
$hauteur_miniature = $hauteur / $largeur * 300;
```

Une petite règle de trois pour retailler la largeur de l'image à 300 pixels avec une hauteur proportionnelle. Niveau de math requis : cours moyen.

`$im = imagecreatefromjpeg($_FILES['photo']['tmp_name']);`

Stocke toute la photo dans la variable *$im*. Oui c'est possible, merci PHP.

`$im_miniature = imagecreatetruecolor($largeur_miniature, $hauteur_miniature);`

Prépare une image tampon noire en 24 bits d'une largeur de 300 pixels avec une hauteur proportionnelle à la photo d'origine.

```
imagecopyresampled($im_miniature, $im, 0, 0, 0, 0, $largeur_miniature, $hauteur_miniature,
$largeur, $hauteur);
```

Redimensionnement de la grande photo temporaire et stockage dans l'image tampon *$im_miniature*.

`imagejpeg($im_miniature, 'miniatures/' . $_FILES['photo']['name'], 90);`

Création de la miniature dans le sous-dossier miniature avec le nom d'origine de la photo et en qualité jpeg 90%

Autre exemple : scan des images jpeg dans un dossier et création des vignettes

Créez un dossier *newyork*, copiez y des photos de New York au format jpeg. Tapez le code suivant et enregistrez le dans le dossier *newyork*.

newyork.php

```
<!doctype html>
<html>
    <head>
        <meta charset="utf-8">
    </head>
    <body>
    <h1>New York</h1>
    <?php
    if (!is_dir('vignettes'))
        mkdir('vignettes', 0777);
    $fichier = opendir('.');
    while ($fichier_courant = readdir($fichier)) {
        $extension = strtolower(strrchr($fichier_courant, '.'));
        if ($extension == '.jpg' || $extension == '.jpeg') {
            $nom_vignette = 'vignettes/' . $fichier_courant;
            $taille = getimagesize($fichier_courant);
            $largeur = $taille[0];
            $hauteur = $taille[1];
            if (!file_exists($nom_vignette)) {
                $im = imagecreatefromjpeg($fichier_courant);
                $largeur_vignette = 150;
                $hauteur_vignette = $hauteur / $largeur * 150;
                $im_vignette = imagecreatetruecolor($largeur_vignette, $hauteur_vignette);
                imagecopyresampled($im_vignette, $im, 0, 0, 0, 0, $largeur_vignette,
$hauteur_vignette, $largeur, $hauteur);
                imagejpeg($im_vignette, $nom_vignette, 60);
            }
            else {
                echo 'Nom de l\'image : '.$fichier_courant.'<br>
                Largeur : ' . $largeur.'<br>
                Hauteur : ' . $hauteur.'<br>
                <a href="' . $fichier_courant . '"><img src="' . $nom_vignette . '"
title="Cliquez pour agrandir"></a>
                <hr>';
            }
        }
    }
    ?>
    </body>
</html>
```

Explications

```
if (!is_dir("vignettes")) mkdir("vignettes", 0777);
```

Si le sous-dossier vignettes n'existe pas, alors on le crée avec les droits en lecture, écriture.

```
$fichier = opendir(".");
```

On ouvre le dossier courant. Le point représente le dossier où se trouve le fichier *newyork.php*.

```
while ($fichier_courant = readdir($fichier)) {…}
```

Tant qu'il y a des fichiers dans le dossier courant, on exécute la boucle. Chaque fichier lu est stocké dans la variable $fichier_courant.

```
$extension = strtolower(strrchr($fichier_courant, "."));
```

Récupération de l'extension du fichier dans la variable *$extension*.
strtolower() : Conversion en minuscule afin que l'instruction conditionnelle if suivante fonctionne.
strrchr() : Trouve tous les caractères de droite de la chaine $fichier_courant à partir du point inclus. On récolte donc l'extension du fichier (exemple : .html, .php, .gif, .jpg)

```
$im = imagecreatefromjpeg($fichier_courant)
```

Copie la photo en cours dans la variable *$im*.

```
$taille = getimagesize($fichier_courant)
```

Cette fonction renvoie un tableau contenant les 4 éléments suivants :

- $taille[0] est la largeur de l'image
- $taille[1] est la hauteur de l'image
- $taille[2] donne un numéro correspondant au format de l'image (1 => GIF, 2 => JPEG, 3 => PNG)
- $taille[3] donne une chaine de caractère utilisable directement dans la balise

  (base64)

```
$im = imagecreatetruecolor($largeur,$hauteur);
```

Crée une image vide 24 bits, soit 16 millions de couleurs, dans la variable *$im* de largeur *$largeur* et de hauteur *$hauteur*.

```
imagecopyresampled($dst_im, $src_im, $dst_x, $dst_y, $src_x, $src_y, $src_l, $src_h)
```

Copie une partie rectangulaire de l'image *$src_im* dans l'image *$dst_im*. La partie à copier est délimitée par le point (*$src_x*, *$src_y*), la largeur *$src_l* et la hauteur *$src_h*. La copie est placée dans *$dst_im* à partir du point (*$dst_x*, *$dst_y*). Une fonction équivalente existe : *imagecopyresized()*. Cette fonction sollicite moins le processeur, plus légère donc, mais l'image redimensionnée est moins jolie.

```
imagejpeg($im, $nom_image, $qualite_image);
```

Génération du fichier jpeg à partir de la variable *$im*. La variable *$qualite_image* permet de régler la qualité de l'image et la compression : entre 0 et 100.

Résultat :

Mozilla Firefox

http://localhost/newy

localhost/newyork/newyork.php

Rechercher

New York

Nom de l'image : guide-des-regles-de-savoir-vivre-new-york-times-square-taxis.jpg
Largeur : 1170
Hauteur : 780

Nom de l'image : New-York.jpg
Largeur : 2000
Hauteur : 1000

Nom de l'image : 2185_hodesti_00_p_1024x768.jpg
Largeur : 1024
Hauteur : 768