

PHP - Les images

Date de dernière mise à jour : 07/07/2007 à 15:37

Source : <http://www.vulgarisation-informatique.com/php-gd.php>.

Distribution interdite sans accord écrit d'Anthony ROSSETTO (<http://www.vulgarisation-informatique.com/contact.php>)

Présentation

PHP ne se limite pas à la génération de pages (X)HTML. En effet, celui-ci peut également afficher des images et les générer à la volée. Vous avez sans doute déjà vu des formulaires demandant un code à recopier pour être sûr que ce n'est pas un ordinateur qui remplit les champs automatiquement, et bien PHP permet de créer des images de ce style, mais heureusement, ses performances ne s'arrêtent pas là. Vous allez pouvoir créer des images aux possibilités limitées uniquement par votre créativité artistique.

Activer la librairie GD

Par défaut, la librairie GD est désactivée. Il va donc falloir l'activer car sinon vous ne pourrez pas faire ce qui va suivre. Pour l'activer, il faut que vous trouviez un fichier nommé "php.ini" qui contient les paramètres de configuration de PHP. Ce fichier se situe généralement dans le répertoire "apache2" d'easyPHP ou encore de Wamp. Essayez également dans le répertoire C:\Windows. Une fois ce fichier ouvert, localisez la ligne suivante : **;extension=php_gd2.dll**, supprimez le point virgule, enregistrez le fichier php.ini puis redémarrez Apache (ou relancez complètement le programme que vous utilisez pour travailler en PHP, easyPHP, Wamp, etc ...)

Créer une image

Avant de parler de création d'image proprement dite, il faut savoir qu'il existe plusieurs types d'images. Les types les plus répandus sur le web sont les formats gif, jpg et png. Cela tombe bien, GD permet de gérer ces trois formats (et d'autres, mais nous ne les utiliserons pas ici).

Pour que le navigateur sâche qu'il envoie une image au navigateur et non une page (X)HTML, on doit envoyer un en-tête spécial sous peine de voir s'afficher des caractères incompréhensibles lorsque l'on demandera l'image. Cet en-tête dit juste "voilà, j'envoie une image de tel type". On doit renseigner dans cet en-tête (les en-têtes sont aussi appelés headers) le type mime du fichier envoyé. Je vais vous donner les trois types mimes que nous utiliserons :

- Pour une image gif, le type mime à utiliser est **image/gif**
- Pour une image jpg, le type mime à utiliser est **image/jpeg**
- Pour une image png, le type mime à utiliser est **image/png**

Il faut maintenant que l'on envoie l'en-tête, on utilise pour cela la fonction header() en PHP. Mettons que l'on veuille générer une image jpg, on utilisera le code suivant :

```
<?php header  
(  
'Content-type: image/jpeg'  
);  
?>
```

Création d'une image de base

Pour créer une image, il y a deux solutions. Vous pouvez partir de rien du tout (et créer l'image de toutes pièces) ou partir d'une image déjà existante sur le disque dur. Pour créer une image en partant de zéro, on va utiliser la fonction imagecreate(), pour créer une image à partir d'un fichier, il existe trois fonctions en fonction du type de fichier de l'image source :

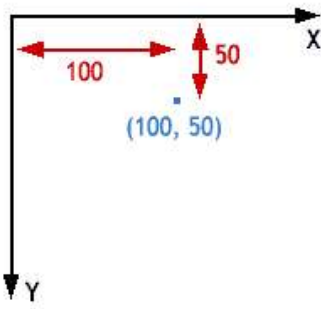
- imagecreatefromgif() : cette fonction permet de créer une image en partant d'une image gif source, on passera en paramètre le nom de l'image gif.
- imagecreatefromjpeg() : cette fonction permet de créer une image en partant d'une image jpg source, on passera en paramètre le nom de l'image jpg.
- imagecreatefrompng() : cette fonction permet de créer une image en partant d'une image png source, on passera en paramètre le nom de l'image png.

En cas d'utilisation d'une fonction `imagecreatefromX()`, les dimensions de l'image que vous allez générer seront pour le moment dictées par les dimensions de l'image que vous avez ouverte. Dans le cas contraire, il faudra préciser les dimensions de l'image à créer.

Voici un exemple créant une image de base, sans ouvrir d'image existante :

```
<?php header
(
'Content-type: image/jpeg'
);
$image
=
imagecreate
(
100
,
50
);
//Crée une image de 100 pixels de large et 50 pixels de haut
?>
```

Lorsque nous utiliserons les fonctions de la librairie GD, nous travaillerons en coordonnées X, Y selon ce plan :



Pour créer une image tout en se servant d'une image déjà existante, voici un exemple pour une image jpeg, il suffit juste de remplacer le nom de la fonction `imagecreatefromjpeg` par le nom de la fonction qui va bien pour le fichier que vous souhaitez :

```
<?php header
(
'Content-type: image/jpeg'
);
$image
=
imagecreatefromjpeg
(
'monimage.jpg'
);
//Crée une image à partir de 'monimage.jpg'
?>
```

Envoi de l'image au navigateur ou sauvegarde sur le disque

Une fois que vous avez fini de générer votre image, il faut soit l'envoyer au navigateur, soit la sauvegarder sur le disque. On utilise pour cela différentes fonctions qui diffèrent uniquement selon le type d'image que vous allez envoyer ou sauvegarder. Voici la liste des fonctions que nous

serons amenés à utiliser :

-imagegif() : cette fonction permet de générer une image gif. Si vous souhaitez sauvegarder l'image sur le disque, dans le paramètre de la fonction, entrez le nom du fichier sous lequel l'image sera sauvegardée.

-imagejpeg() : cette fonction permet de générer une image jpg. Si vous souhaitez sauvegarder l'image sur le disque, dans le paramètre de la fonction, entrez le nom du fichier sous lequel l'image sera sauvegardée.

-imagepng() : cette fonction permet de générer une image png. Si vous souhaitez sauvegarder l'image sur le disque, dans le paramètre de la fonction, entrez le nom du fichier sous lequel l'image sera sauvegardée.

Les couleurs

On travaillera en PHP avec le système de couleurs RGB (red, green, blue). Ce système permet de recomposer une quantité énorme de couleurs à partir de 256 possibilités de rouge, 256 possibilités de vert et 256 possibilités de bleu. Vous pourrez si vous le souhaitez connaître le code de votre couleur en utilisant un éditeur d'images. Windows intègre paint par défaut qui permet au moins de connaître ça ;)

Pour que PHP sâche quelle couleur il doit utiliser, on stocke les couleurs dans des variables. Les "codes couleurs" sont générés grâce à la fonction imagecolorallocate().

```
<?php
    $blanc =
imagecolorallocate
(
$image
,
255
,
255
,
255
);
//Premier 0 : quantité de rouge, second 0 : quantité de vert, troisième 0 : quantité de bleu
    $noir
=
imagecolorallocate
(
$image
,
,
,
);
?>
```

Le premier imagecolorallocate() que vous ferez correspondra à la couleur de fond de votre image. Vous pourrez toujours le changer de couleur ultérieurement via des bidouilles, mais pensez-y. Si vous souhaitez une image à fond blanc, il faut utiliser le code couleur 255, 255, 255.

Ecrire du texte

GD permet heureusement d'écrire du texte sur les images. On utilise pour cela la fonction imagestring().

```
<?php imagestring
```

```
(
$image
,
$taille_de_police
,
$x
,
$y
,
$texte
,
$couleur
);
?>
```

Nous allons maintenant détailler les paramètres de cette fonction :

- `$image` : correspond à l'image que vous avez créée avec `imagecreate()`
- `$taille_de_police` : il s'agit ici d'un nombre allant de 1 à 5 précisant la grosseur de la police. Nous n'utiliserons que pour ce chapitre des polices par défaut.
- `$x` : coordonnées en pixels sur l'axe des abscisses.
- `$y` : coordonnées en pixels sur l'axe des ordonnées.
- `$couleur` : une couleur générée avec `imagecolorallocate()`

Si nous souhaitons écrire "Salut à tous" aux coordonnées (10, 50) en police noire de taille 4, on procédera comme ceci :

```
<?php imagestring
(
$image
,
4
,
10
,
50
,
'Salut à tous !'
,
$noir
);
?>
```

Nous allons maintenant voir l'exemple complet :

```
<?php header
(
'Content-type: image/jpeg'
);
$image
=
imagecreate
(
```

```

100
,
50
);
//Crée une image de 100 pixels de large et 50 pixels de haut
$blanc
=
imagecolorallocate
(
$image
,
255
,
255
,
255
);
//Premier 0 : quantité de rouge, second 0 : quantité de vert, troisième 0 : quantité de bleu
$noir
=
imagecolorallocate
(
$image
,
,
,
,
);
imagestring
(
$image
,
4
,
10
,
50
,
'Salut à tous !'
,
$noir
);
//Affiche une chaîne sur l'image
imagejpeg
(
$image
);
//Renvoie l'image au navigateur
?>

```

Remarquez l'utilisation de la fonction `imagejpeg()` qui prend en paramètre l'image que l'on souhaite envoyer/sauvegarder. Cela permet à PHP ici

d'envoyer l'image au navigateur.

Quelques formes bien utiles

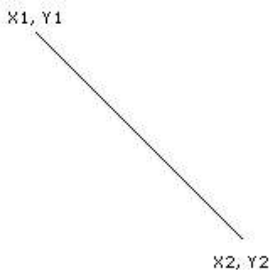
Heureusement, les possibilités de GD ne s'arrêtent pas là et il est possible de créer de nombreuses formes. Voici quelques fonctions bien utiles et leurs effets :

Créer une ligne :

La fonction **imageline()** permet de créer une ligne. Voici un exemple :

```
<?php$noir = imagecolorallocate($image, 0, 0, 0);imageline($image, $x1, $y1, $x2, $y2, $noir);?>
```

Nous venons de créer une ligne. Les coordonnées du premier point sont (x_1, y_1) et du second point (x_2, y_2). La ligne aura une couleur valant noir (ici la couleur noire, qui vaut 0,0,0 en RVB).



Modifier le style des lignes et des contours de formes :

La fonction **imagesetstyle()** permet de modifier le style des lignes et des contours de formes pour créer des lignes personnalisées. Voici un exemple :

```
<?php header('Content-type: image/gif');$image = imagecreate(200, 200);$blanc = imagecolorallocate($image, 255
```

```
,
255
,
255
);
$noir
=
imagecolorallocate
(
$image
,
,
,
);
$rouge
=
imagecolorallocate
(
$image
,
255
,
,
);
$style
= array(
$noir
,
$rouge
,
$noir
,
$rouge
,
$rouge
,
$rouge
,
$noir
);
//un pixel noir, un pixel rouge, un pixel noir ...
imagesetstyle
(
$image
,
$style
);
//Applique le style sur l'image
```

```

imagesetthickness
(
$image
,
20
);
//Modifie l'épaisseur de la ligne
imageline
(
$image
,
50
,
60
,
150
,
60
,
IMG_COLOR_STYLED
);
//On utilise la constante IMG_COLOR_STYLED qui signifie que GD doit utiliser le style défini plus haut
imagegif
(
$image
);
?>

```

Nous venons de créer une ligne. Les coordonnées du premier point sont (50, 60) et du second point (150, 60). La ligne sera personnalisée (notez la constante **IMG_COLOR_STYLED**). Il y aura respectivement un pixel noir, un pixel rouge, un pixel noir, trois pixels rouges et un pixel noir (voir le tableau \$style

).

Voilà ce que cela donne en pratique :



Créer un rectangle :

La fonction **imagecreate()** permet de créer un rectangle. On l'utilise comme ceci :

```

<?php header
(
'Content-type: image/gif'
);
$image
=
imagecreate

```

```

(
200
,
200
);
$blanc
=
imagecolorallocate
(
$image
,
255
,
255
,
255
);
$noir
=
imagecolorallocate
(
$image
,
,
,
);
imagerectangle
(
$image
,
50
,
50
,
150
,
150
,
$noir
);
imagegif
(
$image
);
?>

```

Nous venons de créer un rectangle. Les coordonnées du premier point sont (\$x1,\$y1) et du second point (\$x2,\$y2). Le rectangle sera comme tous nos exemples précédents noir : \$noir (ici la couleur noire, qui vaut 0,0,0 en RVB). Voilà ce que cela donne en pratique :

`x1, y1`



Maintenant appliquons ce que l'on a appris concernant les contours personnalisés :

```
<?php header
(
'Content-type: image/gif'
);
$image
=
imagecreate
(
200
,
200
);
$blanc
=
imagecolorallocate
(
$image
,
255
,
255
,
255
);
$noir
=
imagecolorallocate
(
$image
,
,
,
);
$rouge
=
imagecolorallocate
(
$image
,
255
,
,
);
$style
= array(
$noir
```

```
,
$rouge
,
$noir
,
$rouge
,
$rouge
,
$rouge
,
$noir
);
//un pixel noir, un pixel rouge, un pixel noir ...
imagesetstyle
(
$image
,
$style
);
//Applique le style sur l'image, ici la bordure du rectangle sera de ce style
imagesetthickness
(
$image
,
10
);
//Ici, la largeur correspondra à la largeur du contour du rectangle
imagerectangle
(
$image
,
50
,
50
,
150
,
150
,
IMG_COLOR_STYLED
);
imagegif
(
$image
);
?>
```

X1, Y1



X2, Y2

Ca a de la classe non ? Si vous souhaitez que le rectangle soit rempli, on utilise la fonction `imagefilledrectangle()` qui s'utilise de la même façon que la fonction `imagerectangle()`.

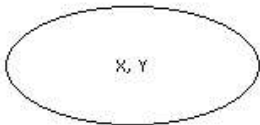
Créer une ellipse :

Vous pouvez quand même créer autre chose que des lignes et des rectangles ! Il existe par exemple une fonction vous permettant de créer des ellipses. Celle-ci porte le nom de `imageellipse()`. Voici ce que ça donne au niveau du code :

```
<?php header
(
'Content-type: image/gif'
);
$image
=
imagecreate
(
200
,
200
);
$blanc
=
imagecolorallocate
(
$image
,
255
,
255
,
255
);
$noir
=
imagecolorallocate
(
$image
,
,
,
);
imageellipse
(
$image
,
100
,
50
,
150
```

```
,  
70  
,  
$noir  
);  
imagegif  
(  
$image  
);  
?>
```

Et ce que ça donne en pratique :



Inutile que je vous montre comment créer une ellipse avec un contour personnalisé, je pense que vous savez comment faire maintenant ;)

Pour ce qui est de l'ellipse remplie intégralement, on utilise la fonction `imagefilledellipse()` qui se comporte comme la fonction `imageellipse()`.

Créer un polygone :

La fonction **`imagepolygon()`** permet de créer un polygone. On l'utilise comme ceci :

```
<?php header  
(  
'Content-type: image/gif'  
);  
$image  
=  
imagecreate  
(  
200  
,  
200  
);  
$blanc  
=  
imagecolorallocate  
(  
$image  
,  
255  
,  
255  
,  
255  
);  
$noir
```

```

=
imagecolorallocate
(
$image
,
,
,
);
$points
= array(
50
,
10
,
70
,
60
,
90
,
40
);
imagepolygon
(
$image
,
$points
,
3
,
$noir
);
imagegif
(
$image
);
?>

```

Voici ce que ça donne :



Inutile de vous dire que vous pouvez également créer des polygones aux lignes personnalisées ou/et remplis. Ici, le chiffre 3 dans les paramètres de la fonction correspond au nombre de points du polygone. Les points sont déterminés par le tableau \$points. Détaillons un peu le principe de ce tableau via cette image :



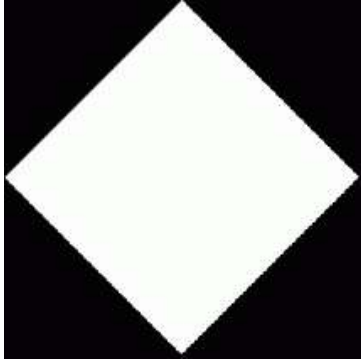
Effets sur les images

Les possibilités de GD ne s'arrêtent pas là. Vous allez pouvoir faire des rotations d'images, rendre des images transparentes, redimensionner des images ...

Voyons comment effectuer la rotation d'une image :

```
<?php header
(
'Content-type: image/gif'
);
$image
=
imagecreate
(
150
,
150
);
$blanc
=
imagecolorallocate
(
$image
,
255
,
255
,
255
);
//On a donc une image de 150*150 sur fond blanc
$noir
=
imagecolorallocate
(
$image
,
,
,
);
$image
=
imagerotate
(
$image
,
45
,
```

```
$noir
);
//L'image sur fond blanc a été tournée de 45° et le "vide" a été comblé par du noir
imagegif
(
$image
);
?>
```



Rendre une image transparente :

Les images GIF et PNG supportent la transparence. PHP et GD vont vous permettre de définir une couleur transparente sur une image. Reprenons l'image ci-dessus qui a été tournée de 45 degrés. Nous aimerions faire disparaître le noir. Voilà comment on pourrait procéder :

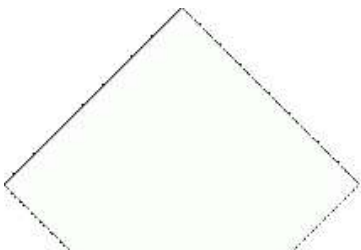
```
<?php header
(
'Content-type: image/gif'
);
$image
=
imagecreate
(
150
,
150
);
$blanc
=
imagecolorallocate
(
$image
,
255
,
255
,
255
);
//On a donc une image de 150*150 sur fond blanc
$noir
```

```

=
imagecolorallocate
(
$image
,
,
,
);
$image
=
imagerotate
(
$image
,
45
,
$noir
);
//L'image sur fond blanc a été tournée de 45° et le "vide" a été comblé par du noir
$image
=
imagecolorallocate
(
$image
,
,
,
);
//On réalloue du noir, l'image ayant été modifiée.
imagecolortransparent
(
$image
,
$noir
);
//Le noir devient transparent
imagegif
(
$image
);
?>

```

Voici ce que cela donne (il reste encore des pixels "presque noirs" voilà pourquoi ils n'ont pas été supprimés, car ils ne sont pas exactement noirs ;)) :



Redimensionner une image :

PHP et GD sont très utilisés pour redimensionner les images. Par exemple, sur ce site, votre avatar est redimensionné automatiquement si il est trop grand. J'utilise GD pour faire cela, avec une fonction très simple d'utilisation. Il s'agit de la fonction **imagecopyresampled()**. Imaginons que nous souhaitons réduire l'image à 100*100 pixels :

```
<?php header
(
'Content-type: image/gif'
);
$image
=
imagecreate
(
150
,
150
);
$destination
=
imagecreate
(
100
,
100
);
//Image qui sera l'image de destination
$blanc
=
imagecolorallocate
(
$image
,
255
,
255
,
255
);
//On a donc une image de 150*150 sur fond blanc
$noir
=
imagecolorallocate
(
$image
,
,
```

```
);
$image
=
imagerotate
(
$image
,
45
,
$noir
);
//L'image sur fond blanc a été tournée de 45° et le "vide" a été comblé par du noir
$noir
=
imagecolorallocate
(
$image
,
,
,
,
);
//On réalloue du noir, l'image ayant été modifiée.
$largeur_src
=
imagesx
(
$image
);
//Renvoie la largeur de l'image source
$hauteur_src
=
imagesy
(
$image
);
//Renvoie la hauteur de l'image source
imagecopyresampled
(
$destination
,
$image
,
,
,
,
```

```
100
,
100
,
$largeur_src
,
$hauteur_src
);
imagecolortransparent
(
$destination
,
$noir
);
//Le noir devient transparent
imagegif
(
$destination
);
?>
```

Voilà ce que l'on obtient :



Chapitre suivant



[Envoyer des mails en PHP](#)

[Présentation](#)

[Envoyer des mails au format texte](#)

[Les en-têtes \(headers\)](#)

[Envoyer des mails au format HTML](#)

[Envoyer des mails aux formats texte et HTML](#)

[Envoi de mails avec pièces jointes](#)

Source : <http://www.vulgarisation-informatique.com/php-gd.php>.

Distribution interdite sans accord écrit d'Anthony ROSSETTO (<http://www.vulgarisation-informatique.com/contact.php>)