

CREATION D'IMAGES EN PHP

La création, l'utilisation, la modification d'images est largement utilisée par exemple pour générer des graphiques, ajouter un logo sur une image (copyrighting), ajouter des informations sur des images capturées par une webcam, etc.

PHP utilise des extensions (bibliothèques fournies avec PHP mais qui ne sont pas forcément activées). Vous pouvez visualiser la liste des extensions en cliquant sur "Administration". Au niveau de la partie PHP, il est indiqué le nombre d'extensions chargées. En cliquant sur "afficher", vous pouvez voir le détail des extensions chargées et les fonctions de la bibliothèque.

En ce qui concerne le graphisme, l'extension qui nous intéresse se nomme "GD". Vous pouvez constater qu'il y a un grand nombre de fonctions dans cette bibliothèque.

1 PRINCIPE DE CREATION D'UNE IMAGE

1.1 Initialisations

Lorsqu'on crée une image en PHP, on doit préciser au navigateur que le résultat du script est une image. Ceci se fait avec la fonction **header** (qui envoie un en-tête HTTP) en précisant le format de l'image :

```
header ("Content-type : image/png"); // Image au format png  
header ("Content-type : image/jpeg"); // Image au format jpg
```



La fonction **header ()** doit être appelée avant d'écrire le moindre code HTML.

Ensuite, on va créer l'image proprement dite. Initialement, on crée une image vide. Il suffit de préciser les dimensions de l'image en pixels.

```
$image=imagecreate (300 , 75) ;
```

NB : Vous verrez plus bas la fonction **imagecreatetruecolor** qui est identique à celle-ci mais qui permet d'utiliser les couleurs vraies (couleurs définies sur 24bits).

Ces fonctions renvoient une ressource, c'est à dire une variable qui contiendra toutes les informations utiles concernant l'image.

1.2 Traitements

On va ensuite réaliser le traitement sur l'image. Ces traitements peuvent être variés :

- Dans le cas de la génération d'une image, on utilisera les fonctions de tracé de formes élémentaires (cf. plus bas) qui permettront de dessiner un point, une ligne ou une forme de base (cercle, ellipse, carré, rectangle ou polygone). Ces formes peuvent être dessinées de façon (outlined ou filled) c'est à dire que l'on peut dessiner un cercle ou un disque.
- On peut choisir d'écrire un texte sur une image existante (ou sur une image préalablement générée).
- On peut redimensionner une image, fusionner plusieurs images, etc.

1.3 Affichage de l'image :

On utilisera la fonction **imagepng (\$image) ;** ou **imagejpeg (\$image) ;** selon le format choisi initialement.

Variante:

```
imagepng ($image , $filename) ; ou imagejpeg ($image , $filename) ;
```

Permet d'enregistrer l'image **\$image** dans le fichier **\$filename**.

LES PRINCIPALES FONCTIONS DE LA LIBRAIRIE GD

Vous trouverez la liste des fonctions GD sur <http://fr2.php.net/manual/fr/book.image.php>
On va se limiter ici à décrire les fonctions les plus utiles pour le TP.

```
ressource imagecreatetruecolor(int $width, int $height)
```

Cette fonction retourne une ressource représentant une image noire.

- **width et height** : Désignent respectivement la largeur et la hauteur de l'image

```
int imagecolorallocate(ressource $image ,int $red ,int $green  
,int $blue)
```

Cette fonction retourne un identifiant de couleur, représentant la couleur composée avec les couleurs RGB.

- **image** : Une ressource d'image, retourné par une des fonctions de création d'images, comme `imagecreatetruecolor()`.
- **red, green, blue** : Composantes Red, Green, Blue de la couleur. Chaque couleur est codée sur un octet donc la valeur d'une composante est comprise entre 0 et 255.

Tracés de formes élémentaires

```
bool imagesetpixel ( resource $image , int $x , int $y ,  
int $color )
```

Cette fonction dessine un pixel aux coordonnées (x,y) spécifiées.

```
bool imageline ( resource $image , int $x1 , int $y1 ,  
int $x2 , int $y2 , int $color )
```

Cette fonction dessine une ligne entre deux points fournis (x1,y1) et (x2,y2).

```
bool imageellipse ( resource $image , int $cx , int $cy ,  
int $width , int $height , int $color )
```

Cette fonction dessine une ellipse centrée sur le point spécifié.

NB : Un cercle est une ellipse particulière (**width=height**).

```
bool imagerectangle ( resource $image , int $x1 , int $y1  
, int $x2 , int $y2 , int $color )
```

Cette fonction dessine un rectangle aux coordonnées spécifiées en commençant par le sommet supérieur gauche (x1,y1) et finissant au sommet inférieur droit (x2,y2).

```
bool imagepolygon ( resource $image , array $points , int  
$num_points , int $color )
```

Cette fonction dessine un polygone dans l'image image .

- **points** : Tableau qui contient les sommets du polygone sous la forme : `points[0] = x0, points[1] = y0, points[2] = x1, points[3] = y1, etc.`

Toutes les fonctions précédentes ont leur équivalent "rempli" (**filled**). Dans ce cas, la forme est entièrement colorée et non plus délimitée par un segment de couleur. On ne reprendra donc pas toutes les fonctions précédemment détaillées mais juste une pour l'exemple :

```
bool imagefilledrectangle ( resource $image , int $x1 ,  
int $y1 , int $x2 , int $y2 , int $color )
```

Dessine un rectangle plein de couleur color dans l'image image.