GD Library (Graphics Draw)

The GD library is used for dynamic image creation. From PHP we use with the GD library to create GIF, PNG or JPG images instantly from our code. This allows us to do things such as create charts on the fly, created an an anti-robot security image, create thumbnail images, or even build images from other images.

imagecreate():

```
resource imagecreate ( int $width , int $height )
```

imagecolorallocate():

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

imagestring():

```
bool imagestring ( resource $image , int $font , int $x , int $y , string $string , int $color )
```

imagerotate():

```
resource imagerotate ( resource $image , float $angle , int $bgd_color [, int $ignore_transparent = 0 ] )
```

imageline():

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

imageellipse():

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

imagefilledarc():

```
bool imagefilledarc ( resource $image , int $cx , int $cy , int $width , int $height , int $start , int $end , int $color , int $style )
```

Example:

```
<?php header ("Content-type: image/png");
 $handle = ImageCreate (130, 50) or die ("Cannot Create image");
 $bg_color = ImageColorAllocate ($handle, 255, 0, 0);
 ImagePng ($handle);
 ?>
```

Example :

```
<?php
 header ("Content-type: image/png");
 $handle = ImageCreate (130, 50) or die ("Cannot Create image");
 $bg_color = ImageColorAllocate ($handle, 255, 0, 0);
 $txt_color = ImageColorAllocate ($handle, 0, 0, 0);
 ImageString ($handle, 5, 5, 18, "Welcome", $txt_color);
 ImagePng ($handle);
 ?>
```

Example : Rotate an image

```
<?php
//Your image, or your user's image that you want to rotate
$image = '1.jpg';

//Number of degrees you wish to rotate the image, this could be a drop down value
$degrees = 90;

// This sets the image type to .jpg but can be changed to png or gif
header('Content-type: image/jpeg') ;

// Makes the Canvas
$source = imagecreatefromjpeg($image) ;

// Rotates the image
$rotate = imagerotate($source, $degrees, 0) ;

// Outputs the new jpg image, you could change this to gif or png if needed
imagejpeg($rotate) ;
?>
```

Example : Playing with Fonts

```
<?php
 header ("Content-type: image/png");
 $handle = ImageCreate (130, 50) or die ("Cannot Create image");
 $bg_color = ImageColorAllocate ($handle, 255, 0, 0);
 $txt_color = ImageColorAllocate ($handle, 0, 0, 0);
 ImageTTFText ($handle, 20, 15, 30, 40, $txt_color, "uploads/BRLNSB.ttf", "TEST");
 ImagePng ($handle);
 ?>
```

The first parameter is our handle, then font size, rotation, starting X, starting Y, text color, font, and finally our text. For the font parameter, you need to include the path to font file. For our example I have placed the font Quel in a folder called Fonts. As you can see from our example, we have also set the text to print at a 15 degree angle.

Example : Drawing Lines

```
<?php header ("Content-type: image/png");
 $handle = ImageCreate (130, 50) or die ("Cannot Create image");
 $bg_color = ImageColorAllocate ($handle, 255, 0, 0);
 $txt_color = ImageColorAllocate ($handle, 255, 255, 255);
 $line_color = ImageColorAllocate ($handle, 0, 0, 0);
 ImageLine($handle, 65, 0, 130, 50, $line_color);
 ImageString ($handle, 5, 5, 18, "Welcome", $txt_color);
 ImagePng ($handle);
 ?>
```

To make a cool volcano like we have in our example, we simply put this into a loop, keeping our starting coordinates the same, but moving along the x axis with our finishing coordinates.

```
<?php
 header ("Content-type: image/png");
 $handle = ImageCreate (130, 50) or die ("Cannot Create image");
 $bg_color = ImageColorAllocate ($handle, 255, 0, 0);
 $txt_color = ImageColorAllocate ($handle, 255, 255, 255);
 $line_color = ImageColorAllocate ($handle, 0, 0, 0);
 for($i=0;$i<=129;$i=$i+5)
 {
 ImageLine($handle, 65, 0, $i, 50, $line_color);
 }
 ImageString ($handle, 5, 5, 18, "Welcome", $txt_color);
 ImagePng ($handle);
 ?>
```

Example : Drawing An Ellipse

```
 <?php
 header ("Content-type: image/png");
 $handle = ImageCreate (130, 50) or die ("Cannot Create image");
 $bg_color = ImageColorAllocate ($handle, 255, 0, 0);
 $txt_color = ImageColorAllocate ($handle, 255, 255, 255);
 $line_color = ImageColorAllocate ($handle, 0, 0, 0);
 imageellipse ($handle, 65, 25, 100, 40, $line_color);
 ImageString ($handle, 5, 5, 18, "Welcome", $txt_color);
 ImagePng ($handle);
 ?>
```

The parameters we use with Imageellipse () are the handle, the X and Y center coordinates, the width and height of the ellipse, and the color. Like we did with our line, we can also put our ellipse into a loop to create a spiral effect.

```
 <?php
 header ("Content-type: image/png");
 $handle = ImageCreate (130, 50) or die ("Cannot Create image");
 $bg_color = ImageColorAllocate ($handle, 255, 0, 0);
 $txt_color = ImageColorAllocate ($handle, 255, 255, 255);
 $line_color = ImageColorAllocate ($handle, 0, 0, 0);
 for($i=0;$i<=130;$i=$i+10)
 {
 imageellipse ($handle, $i, 25, 40, 40, $line_color);
 }
 ImageString ($handle, 5, 5, 18, "Welcome", $txt_color);
 ImagePng ($handle);
 ?>
```

Example : Arcs & Pies

```
 <?
 header('Content-type: image/png');
 $handle = imagecreate(100, 100);
 $background = imagecolorallocate($handle, 255, 255, 255);
```

```
$red = imagecolorallocate($handle, 255, 0, 0);
$green = imagecolorallocate($handle, 0, 255, 0);
$blue = imagecolorallocate($handle, 0, 0, 255);
imagefilledarc($handle, 50, 50, 100, 50, 0, 90, $red, IMG_ARC_PIE);
imagefilledarc($handle, 50, 50, 100, 50, 90, 225 , $blue, IMG_ARC_PIE);
imagefilledarc($handle, 50, 50, 100, 50, 225, 360 , $green, IMG_ARC_PIE);
imagepng($handle);
?>
```

Using imagefilledarc we can create a pie, or a slice. The parameters are: handle, center X & Y, width, height, start, end, color, and type. The start and end points are in degrees, starting from the 3 o'clock position.

The types are:

1. IMG_ARC_PIE- Filled arch
2. IMG_ARC_CHORD- filled with straight edge
3. IMG_ARC_NOFILL- when added as a parameter, makes it unfilled
4. IMG_ARC_EDGED- Connects to center. You will use this with nofill to make an unfilled pie.

Example : Using imagesx() and imagesy()

imagesx() : return the width of the specified image

imagesy() : return the height of the specified image

```
<?php

// create a 300*200 image
$img = imagecreatetruecolor(300, 200);

echo imagesx($img); // 300
Echo imagesy($img); // 200

?>
```