



Billede klasse

Nu har jeg efterhånden lavet lidt artikler om diverse billede behandlinger, så tænkte det var på tide at smide alle funktioner i en klasse. Det er der kommet denne klasse ud af.

Skrevet den **02. Feb 2009** af **jakobdo** | kategorien **Programmering / PHP** | ★★★★★

Jeg ved godt det ikke en artikel, men i kan jo bruge den hvis i vil. Klassen formål er at gøre det nemt at arbejde med billeder i php. Klassen kan resize, croppe, rotere, grayscale, flippe, vandmærke og selvfølgelig kombinationer af dem alle. Rettelser/forbedringer/tilføjelser og lignende modtages med kys hånd.

```
<?php

/*****\
* USAGE *
* $image = new image('picture.ext'); *
\*****/

class image{
    private $image;
    private $quality = 100;

    public function __construct($filename)    {
        $this->imagecreatefrom($filename);
    }

    private function getExtension($filename)    {
        return strtolower(pathinfo($filename,PATHINFO_EXTENSION));
    }

    private function imagecreatefrom($filename) {
        switch($this->getExtension($filename)){
            case 'jpg':
            case 'jpe':
            case 'jpeg':
                $this->resource = imagecreatefromjpeg($filename);
                break;
            case 'gif':
                $this->resource = imagecreatefromgif($filename);
                break;
            case 'png':
                $this->resource = imagecreatefrompng($filename);
                break;
        }
    }
}
```

```

        case 'bmp':
            $this->resource = $this->imagecreatefrombmp($filename);
            break;
        default:
            throw new Exception('Format not supported');
    }
}
//
//http://dk.php.net/gd - shd at earthling dot net - 28-Mar-2006 02:44
//START
//
private function imagebmp($im, $fn = false){
    if (!$im) return false;

    if ($fn === false) $fn = 'php://output';
    $f = fopen ($fn, "w");
    if (!$f) return false;

    //Image dimensions
    $biWidth = imagesx ($im);
    $biHeight = imagesy ($im);
    $biBPLine = $biWidth * 3;
    $biStride = ($biBPLine + 3) & ~3;
    $biSizeImage = $biStride * $biHeight;
    $bfOffBits = 54;
    $bfSize = $bfOffBits + $biSizeImage;

    //BITMAPFILEHEADER
    fwrite ($f, 'BM', 2);
    fwrite ($f, pack ('VvvV', $bfSize, 0, 0, $bfOffBits));

    //BITMAPINFO (BITMAPINFOHEADER)
    fwrite ($f, pack ('VVVvvVVVVV', 40, $biWidth, $biHeight, 1, 24, 0,
$biSizeImage, 0, 0, 0, 0));

    $numpad = $biStride - $biBPLine;
    for ($y = $biHeight - 1; $y >= 0; --$y)
    {
        for ($x = 0; $x < $biWidth; ++$x)
        {
            $col = imagecolorat ($im, $x, $y);
            fwrite ($f, pack ('V', $col), 3);
        }
        for ($i = 0; $i < $numpad; ++$i)
            fwrite ($f, pack ('C', 0));
    }
    fclose ($f);
    return true;
}
//
//http://dk.php.net/gd - shd at earthling dot net - 28-Mar-2006 02:44
//END
//
//

```

```

//http://dk.php.net/gd - DHKold - 15-Jun-2005 11:52
//START
//
private function imagecreatefrombmp($filename){
    if (! $f1 = fopen($filename,"rb")) return FALSE;

    $FILE = unpack("vfile_type/Vfile_size/Vreserved/Vbitmap_offset",
fread($f1,14));
    if($FILE['file_type'] != 19778) return FALSE;

    $BMP = unpack('Vheader_size/Vwidth/Vheight/vplanes/vbits_per_pixel'.
'/Vcompression/Vsize_bitmap/Vhoriz_resolution'.
'/Vvert_resolution/Vcolors_used/Vcolors_important', fread($f1,40));
    $BMP['colors'] = pow(2,$BMP['bits_per_pixel']);
    if ($BMP['size_bitmap'] == 0)
        $BMP['size_bitmap'] = $FILE['file_size'] - $FILE['bitmap_offset'];
    $BMP['bytes_per_pixel'] = $BMP['bits_per_pixel']/8;
    $BMP['bytes_per_pixel2'] = ceil($BMP['bytes_per_pixel']);
    $BMP['decal'] = ($BMP['width']*$BMP['bytes_per_pixel']/4);
    $BMP['decal'] -= floor($BMP['width']*$BMP['bytes_per_pixel']/4);
    $BMP['decal'] = 4-(4*$BMP['decal']);
    if ($BMP['decal'] == 4)
        $BMP['decal'] = 0;

    $PALETTE = array();
    if ($BMP['colors'] < 16777216)
        $PALETTE = unpack('V'.$BMP['colors'],
fread($f1,$BMP['colors']*4));

    $IMG = fread($f1,$BMP['size_bitmap']);
    $VIDE = chr(0);

    $res = imagecreatetruecolor($BMP['width'],$BMP['height']);
    $P = 0;
    $Y = $BMP['height']-1;
    while ($Y >= 0){
        $X=0;
        while ($X < $BMP['width']){
            if ($BMP['bits_per_pixel'] == 24)
                $COLOR = unpack("V",substr($IMG,$P,3).$VIDE);
            elseif ($BMP['bits_per_pixel'] == 16){
                $COLOR = unpack("v",substr($IMG,$P,2));
                $blue = ($COLOR[1] & 0x001f) << 3;
                $green = ($COLOR[1] & 0x07e0) >> 3;
                $red = ($COLOR[1] & 0xf800) >> 8;
                $COLOR[1] = $red * 65536 + $green * 256 + $blue;
            }elseif ($BMP['bits_per_pixel'] == 8){
                $COLOR = unpack("n",$VIDE.substr($IMG,$P,1));
                $COLOR[1] = $PALETTE[$COLOR[1]+1];
            }elseif ($BMP['bits_per_pixel'] == 4){
                $COLOR = unpack("n",$VIDE.substr($IMG,floor($P),1));
                if (($P*2)%2 == 0)
                    $COLOR[1] = ($COLOR[1] >> 4);
            }
        }
        $Y--;
        $P += $BMP['width']*$BMP['bytes_per_pixel'];
    }
    return imagecolorset($res,$X,$Y,$COLOR[1]);
}

```

```

        else
            $COLOR[1] = ($COLOR[1] & 0x0F);
            $COLOR[1] = $PALETTE[$COLOR[1]+1];
        }elseif($BMP['bits_per_pixel'] == 1){
$COLOR = unpack("n",$VIDE.substr($IMG,floor($P),1));
            if(($P*8)%8 == 0)
                $COLOR[1] = $COLOR[1]>>7;
            elseif(($P*8)%8 == 1)
                $COLOR[1] = ($COLOR[1] & 0x40)>>6;
            elseif (($P*8)%8 == 2)
                $COLOR[1] = ($COLOR[1] & 0x20)>>5;
            elseif (($P*8)%8 == 3)
                $COLOR[1] = ($COLOR[1] & 0x10)>>4;
            elseif (($P*8)%8 == 4)
                $COLOR[1] = ($COLOR[1] & 0x8)>>3;
            elseif (($P*8)%8 == 5)
                $COLOR[1] = ($COLOR[1] & 0x4)>>2;
            elseif (($P*8)%8 == 6)
                $COLOR[1] = ($COLOR[1] & 0x2)>>1;
            elseif (($P*8)%8 == 7)
                $COLOR[1] = ($COLOR[1] & 0x1);
            $COLOR[1] = $PALETTE[$COLOR[1]+1];
        }elseif{
return FALSE;
    }
        imagesetpixel($res,$X,$Y,$COLOR[1]);
        $X++;
        $P += $BMP['bytes_per_pixel'];
    }
        $Y--;
        $P+=$BMP['decal'];
    }

    fclose($f1);
    return $res;
}
//
//http://dk.php.net/gd - DHKold - 15-Jun-2005 11:52
//START
//

/*****\
* USAGE *
* $image->getWidth(); *
* Returns width as integer/pixel *
\*****/
    public function getWidth(){
        return imagesx($this->resource);
    }

/*****\
* USAGE *
* $image->getHeight(); *
* Returns height as integer/pixel *
\*****/

```

```

public function getHeight(){
    return imagesy($this->resource);
}

/*****\
* USAGE *
* $image->quality(); *
* Set quality of image *
* 0 = worst *
* 100 = best *
\*****/
public function quality($quality){
    if(!is_numeric($quality))
        throw new Exception('Quality should be integer, 0 - 100');
    $this->quality = intval($quality);
}

/*****\
* USAGE *
* $image->show(FORMAT,DEBUG); *
* Shows the image in a given format *
* Format: jpg, png, gif and bmp *
* Debug: true/false *
\*****/
public function show($format,$debug=false){
    switch($format){
        case 'jpg':
        case 'jpe':
        case 'jpeg':
            if($debug){
                header('Content-Type: text/html');
            }else{
                header('Content-Type: image/jpeg');
            }
            imagejpeg($this->resource, NULL, $this->quality);
            break;
        case 'gif':
            if($debug){
                header('Content-Type: text/html');
            }else{
                header('Content-Type: image/gif');
            }
            imagegif($this->resource, NULL);
            break;
        case 'png':
            if($debug){
                header('Content-Type: text/html');
            }else{
                header('Content-Type: image/png');
            }
            imagepng($this->resource, NULL, floor( abs( $this->quality /
10 - 9.9 ) ) );
            break;
        case 'bmp':
            if($debug){

```

```

        header('Content-Type: text/html');
    }else{
        header('Content-Type: image/bmp');
    }
    $this->imagebmp($this->resource);
    break;
default:
    throw new Exception('Format not supported');
}
}

/*****\
* USAGE *
* $image->resize(WIDTH,HEIGHT,RATIO); *
* Resize an image *
* Width: integer/pixel *
* Height: integer/pixel *
* Ratio: True = (keep ratio) *
* Ratio: False = (skip ratio) *
\*****/
public function resize($width,$height,$ratio=true){

    $ow = $this->getWidth();
    $oh = $this->getHeight();
    if($ratio){
        $wscale = $width / $ow;
        $hscale = $height / $oh;
        $scale = min( $hscale, $wscale );
        $nw = round( $ow * $scale, 0 );
        $nh = round( $oh * $scale, 0 );
    }else{
        $nw = $width;
        $nh = $height;
    }
    $tempImage = imagecreatetruecolor( $nw, $nh );
    imagecopyresampled( $tempImage, $this->resource, 0, 0, 0, 0, $nw, $nh,
    $ow, $oh );
    imagedestroy($this->resource);
    $this->resource = $tempImage;
}

/*****\
* USAGE *
* $image->crop(X,Y,WIDTH,HEIGHT); *
* Crop an image *
* X: X-Position - integer/pixel *
* Y: Y-Position - integer/pixel *
* Width: integer/pixel *
* Height: integer/pixel *
\*****/
public function crop($x, $y, $width,$height){
    $ow = $this->getWidth();

```

```

        $oh = $this->getHeight();
        if($ow < ($width + $x) || $oh < ($height + $y))
            throw new Exception('Crop failed, picture is too small');
        $cropImage = imagecreatetruecolor( $width, $height );
        imagecopyresampled( $cropImage, $this->resource, 0, 0, $x, $y, $width,
$height, $width, $height );
        imagedestroy($this->resource);
        $this->resource = $cropImage;
    }

/*****\
* USAGE *
* $image->flipHorizontal(); *
* Flip an image horizontal *
\*****/
//
//http://dk.php.net/gd - php at synvb dot com - 22-Apr-2006 05:19
//START
//
function flipHorizontal(){
    $ow = $this->getWidth();
    $oh = $this->getHeight();
    $flipImage = imagecreatetruecolor($ow, $oh);
    for($x = 0; $x < $ow; $x++) {
        imagecopy($flipImage, $this->resource, $x, 0, $ow - $x - 1, 0, 1,
$oh);
    }
    imagedestroy($this->resource);
    $this->resource = $flipImage;
}

/*****\
* USAGE *
* $image->flipVertical(); *
* Flip an image vertical *
\*****/
function flipVertical(){
    $ow = $this->getWidth();
    $oh = $this->getHeight();

    $flipImage = imagecreatetruecolor($ow, $oh);

    for($y = 0; $y < $oh; $y++) {
        imagecopy($flipImage, $this->resource, 0, $y, 0, $oh - $y - 1,
$ow, 1);
    }
    imagedestroy($this->resource);
    $this->resource = $flipImage;
}
//
//http://dk.php.net/gd - php at synvb dot com - 22-Apr-2006 05:19
//END
//

/*****\

```

```

* USAGE *
* $image->rotate(DEGREE,HEXCOLOR); *
* Rotate an image *
* Degree: 0-359 integer *
* HexColor: #fff or #ffffff *
\*****/
//

//http://www.anyexample.com/programming/php/php_convert_rgb_from_to_html_hex_color.xml
//START
//
function rotate($degrees,$hex_bgcolor='#000'){
    if($hex_bgcolor[0] == '#')
        $hex_bgcolor = substr($hex_bgcolor, 1);

    if (strlen($hex_bgcolor) == 6)
        list($r, $g, $b) = array($hex_bgcolor[0].$hex_bgcolor[1],
$hex_bgcolor[2].$hex_bgcolor[3], $hex_bgcolor[4].$hex_bgcolor[5]);
    elseif (strlen($hex_bgcolor) == 3)
        list($r, $g, $b) = array($hex_bgcolor[0].$hex_bgcolor[0],
$hex_bgcolor[1].$hex_bgcolor[1], $hex_bgcolor[2].$hex_bgcolor[2]);
    else
        return false;

    $r = hexdec($r);
    $g = hexdec($g);
    $b = hexdec($b);
    //

//http://www.anyexample.com/programming/php/php_convert_rgb_from_to_html_hex_color.xml
//END
//

$bgcolor = imagecolorallocate($this->resource, $r, $g, $b);

    $rotateImage = imagerotate($this->resource, $degrees, $bgcolor, 0);

    imagedestroy($this->resource);
    $this->resource = $rotateImage;
}

\*****\
* USAGE *
* $image->grayscale(); *
* Grayscale an image *
\*****/
//
//http://bubble.ro/How_to_convert_an_image_to_grayscale_using_PHP.html
//START
//
function grayscale(){
    $ow = $this->getWidth();
    $oh = $this->getHeight();

```



```

        for ($i = 0; $i < $ow; $i++){
            for ($j = 0; $j < $oh; $j++){
                // get the rgb value for current pixel
                $rgb = ImageColorAt($this->resource, $i, $j);
                // extract each value for r, g, b
                $rr = ($rgb >> 16) & 0xFF;
                $gg = ($rgb >> 8) & 0xFF;
                $bb = $rgb & 0xFF;
                // get the Value from the RGB value
                $g = round(($rr + $gg + $bb) / 3);
                // grayscale values have r=g=b=g
                $val = imagecolorallocate($this->resource, $g, $g, $g);
                // set the gray value
                imagesetpixel ($this->resource, $i, $j, $val);
            }
        }
    }
    //
    //http://bubble.ro/How_to_convert_an_image_to_grayscale_using_PHP.html
    //END
    //

/*****\
* USAGE *
* $image->save(FILENAME); *
* Save an image *
* Filename: Save image as filename *
\*****/
    public function save($filename){
        if(!is_writable(pathinfo($filename,PATHINFO_DIRNAME))){
            throw new Exception('Directory is not writable');
        }
        switch($this->getExtension($filename)){
            case 'jpg':
            case 'jpe':
            case 'jpeg':
                imagejpeg($this->resource, $filename, $this->quality);
                break;
            case 'gif':
                imagegif($this->resource, $filename);
                break;
            case 'png':
                imagepng($this->resource, $filename, floor( abs(
$this->quality / 10 - 9.9 ) ) );
                break;
            default:
                throw new Exception('Format not supported');
        }
    }
}

/*****\
* USAGE *
* $image->watermark(WATERMARK,X,Y); *
* Watermark an image *
* Watermark: Image to use as watermark *

```

```

* X: Left, Center, Right or integer      *
* Y: Top, Center, Bottom or integer     *
\*****/
    public function watermark($watermark,$x='right',$y='bottom'){
        $watermark = new image($watermark);
        if($watermark->getWidth() > $this->getWidth() ||
$watermark->getHeight() > $this->getHeight()){
            throw new Exception('Watermark is too large');
        }
        $positionX = 0;
        $positionY = 0;
        switch(strtolower($x)){
            case 'right':
                break;
            case 'center':
                $positionX = round($this->getWidth() / 2) -
round($watermark->getWidth() / 2);
                break;
            case 'right':
                $positionX = $this->getWidth() - $watermark->getWidth();
                break;
            default:
                $positionX = (is_numeric($x)) ? intval($x) :
($this->getWidth() - $watermark->getWidth());
        }

        switch(strtolower($y)){
            case 'top':
                break;
            case 'center':
                $positionY = round($this->getHeight() / 2) -
round($watermark->getHeight() / 2);
                break;
            case 'bottom':
                $positionY = $this->getHeight() - $watermark->getHeight();
                break;
            default:
                $positionY = (is_numeric($y)) ? intval($y) :
($this->getHeight() - $watermark->getHeight());
        }

        if(($positionX+$watermark->getWidth()) > $this->getWidth() ||
($positionY+$watermark->getHeight()) > $this->getHeight()){
            throw new Exception('Watermark is outside the image');
        }
        imagecopy($this->resource, $watermark->resource, $positionX,
$positionY, 0, 0, $watermark->getWidth(), $watermark->getHeight());
    }
}
?>

```

Brug af klassen, kunne se sådan ud:

```
[/pre]
<?php
require_once('class.image.php');

try{
    $image = new image('image.jpg');
    $image->quality(100);
    $image->crop(400,133,300,400);
    $image->flipHorizontal();
    $image->flipVertical();
    $image->resize(100,100,false);
    $image->grayscale();
    $image->resize(200,100);
    $image->watermark('watermark.png', 'RIGHT','BOTTOM');
    $image->watermark('transparent.png', 20,200);
    $image->rotate(90,'#fff');
    $image->show('gif');
    //$image->save('image.gif');
}catch(Exception $e){
    echo 'Exception Caught : <b >' . $e->getMessage() . '[/b]';
}

?>
[/pre]
```

Sourcen kan hentes her: <http://www.codebreaker.dk/exp/artikler/1201.phps>

Og til dem som har problemer med sourcen ikke vises korrekt i deres editor efter copy/paste, så skyldes det IE7, brug derfor Firefox.

Kommentar af casualty d. 04. Jul 2008 | 1

Nice...Tak

Kommentar af human d. 15. Jun 2008 | 2

Måske kunne du fortælle hva det gør? Hva er formålet med klassen?

Kommentar af gedekid d. 06. Aug 2008 | 3

Kommentar af fizk d. 16. Jun 2008 | 4

Ahhhh ... Det ser ganske fornuftigt ud og ud fra de eksempler du giver kan man da også regne det meste ud, men en lidt mere udvidet api kunne være rart :)

Kommentar af qtax87 (nedlagt brugerprofil) d. 10. Sep 2008 | 5

Human dette står i toppen. "Klassen formål er at gøre det nemt at arbejde med billeder i php"

Utrolig god opsætning og kodning, lækkert med referencer via. links til php.net til folk der ikke kan finde ud af at søge på php.net :o)
Stor, stor 5 tal herfra.

Kommentar af jih d. 19. Jun 2008 | 6

som sædvanligt - ikke en artikel, men utroligt godt stykke kode!-)

Kommentar af tcp d. 21. Sep 2008 | 7

Kommentar af cronaldo d. 24. Dec 2008 | 8

Sjovt qtax87 har givet en 5'er i sin tekst og kun en 3'er i karakter ;-)
.. Frem-, frem-, fremragende kodelykke!
Jeg giver til gengæld både 5 her i teksten og i karakter.. perfekt!

Kommentar af zurekk d. 29. Aug 2008 | 9

Kommentar af zhx d. 13. Jun 2008 | 10

Satme lækkert, 5 her fra...!

Kommentar af vejgaard82 d. 22. Jun 2008 | 11

Kan man ikke få et program som deller det op i linjer da mit prg bare smider det i en lang tekst :(