

[to.pqina.nl](https://pqina.nl)

Create thumbnails with PHP

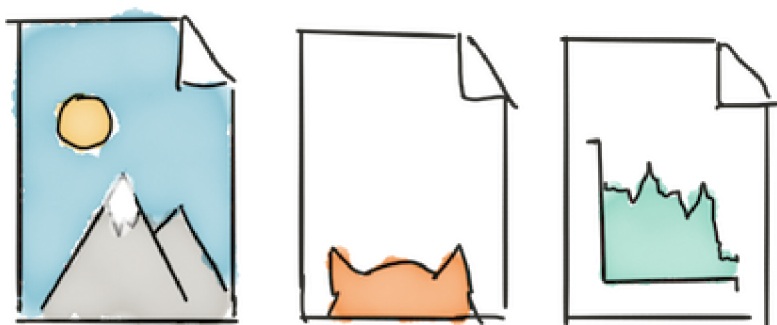
November 3rd, 2016

If you've got photo galleries on your website or need to generate avatars from uploaded profile pictures you'll most likely want to create thumbnails. In this article we'll look at a function that handles various image types and resizes them to a set width and height.

Want to get the code and get back to your project? [Download the script here](#).

For this article I'll assume you've got some images uploaded on your PHP server. The tutorial will not discuss uploading files to the server. Alright. Images ready? Launch your editor and let's do some coding!

PHP has a [long list of functions that can load images](#). It has one function for each image type. As it's really cumbersome to have a thumbnail function per image type our mission is to write one function to handle all the usual image types.



We'll go with JPEGs, PNGs and Cats. I mean GIFs. For those we'll need the following PHP methods:

- `imagecreatefromjpeg` & `imagejpeg`
- `imagecreatefrompng` & `imagepng`
- `imagecreatefromgif` & `imagegif`

Our steps:

1. Load the image (using the correct PHP function)
2. Create a thumbnail by duplicating the image and resizing it
3. Save the thumbnail to disk (using the correct PHP function)

This gives us the following script. It's well documented so each sub-step is explained.

```
// Link image type to correct image loader and saver
// - makes it easier to add additional types later on
// - makes the function easier to read
const IMAGE_HANDLERS = [
  IMAGETYPE_JPEG => [
    'load' => 'imagecreatefromjpeg',
    'save' => 'imagejpeg',
    'quality' => 100
  ],
  IMAGETYPE_PNG => [
    'load' => 'imagecreatefrompng',
```

```
'save' => 'imagepng',
'quality' => 0
],
IMAGETYPE_GIF => [
    'load' => 'imagecreatefromgif',
    'save' => 'imagegif'
]
];

/**
 * @param $src - a valid file location
 * @param $dest - a valid file target
 * @param $targetWidth - desired output width
 * @param $targetHeight - desired output height or null
 */
function createThumbnail($src, $dest, $targetWidth, $targetHeight = null) {

    // 1. Load the image from the given $src
    // - see if the file actually exists
    // - check if it's of a valid image type
    // - load the image resource

    // get the type of the image
    // we need the type to determine the correct loader
    $type = exif_imagetype($src);

    // if no valid type or no handler found -> exit
    if (!$type || !IMAGE_HANDLERS[$type]) {
        return null;
    }

    // load the image with the correct loader
    $image = call_user_func(IMAGE_HANDLERS[$type]['load'], $src);

    // no image found at supplied location -> exit
    if (!$image) {
        return null;
    }

    // 2. Create a thumbnail and resize the loaded $image
    // - get the image dimensions
    // - define the output size appropriately
    // - create a thumbnail based on that size
    // - set alpha transparency for GIFs and PNGs
    // - draw the final thumbnail

    // get original image width and height
    $width = imagesx($image);
    $height = imagesy($image);

    // maintain aspect ratio when no height set
    if ($targetHeight == null) {

        // get width to height ratio
        $ratio = $width / $height;

        // if is portrait
        // use ratio to scale height to fit in square
        if ($width > $height) {
            $targetHeight = floor($targetWidth / $ratio);
        }
    }
}
```

```
// if is landscape
// use ratio to scale width to fit in square
else {
    $targetHeight = $targetWidth;
    $targetWidth = floor($targetWidth * $ratio);
}
}

// create duplicate image based on calculated target size
$thumbnail = imagecreatetruecolor($targetWidth, $targetHeight);

// set transparency options for GIFs and PNGs
if ($type == IMAGETYPE_GIF || $type == IMAGETYPE_PNG) {

    // make image transparent
    imagecolortransparent(
        $thumbnail,
        imagecolorallocate($thumbnail, 0, 0, 0)
    );

    // additional settings for PNGs
    if ($type == IMAGETYPE_PNG) {
        imagealphablending($thumbnail, false);
        imagesavealpha($thumbnail, true);
    }
}

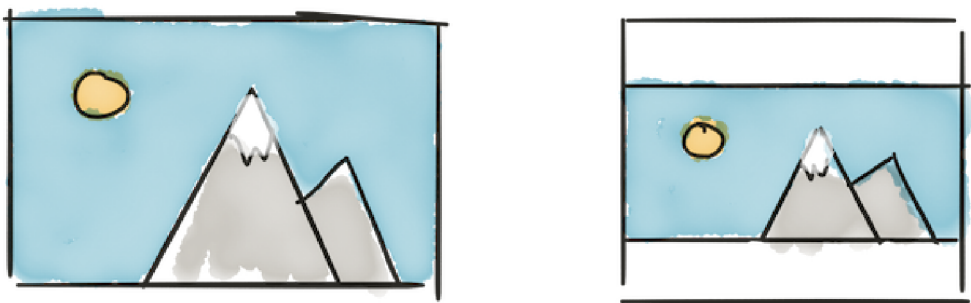
// copy entire source image to duplicate image and resize
imagecopyresampled(
    $thumbnail,
    $image,
    0, 0, 0, 0,
    $targetWidth, $targetHeight,
    $width, $height
);

// 3. Save the $thumbnail to disk
// - call the correct save method
// - set the correct quality level

// save the duplicate version of the image to disk
return call_user_func(
    IMAGE_HANDLERS[$type]['save'],
    $thumbnail,
    $dest,
    IMAGE_HANDLERS[$type]['quality']
);
}
```

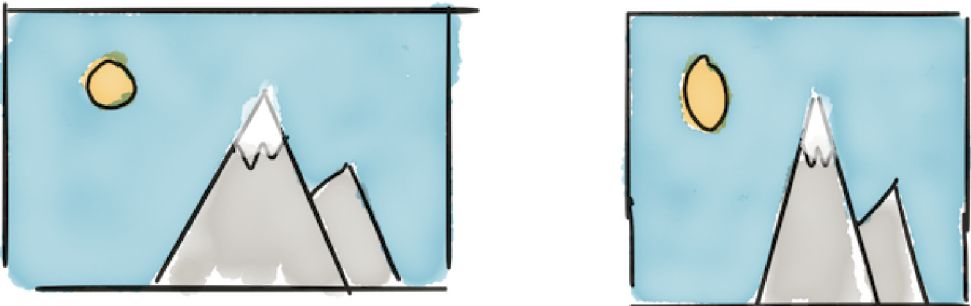
Using this script you can now for example generate square 160 by 160 pixel thumbnails with the following command.

```
createThumbnail('profile.jpg', 'profile_thumb.jpg', 160);
```



If you don't mind your image being skewed you can also supply the fourth argument (which is the output height). The script will then force the output size to the given given width & height values. See the example below.

```
createThumbnail('sunset.jpg', 'sunset_thumb.jpg', 160, 160);
```



That's it!

I use Twitter to share new webdevelopment tips and tricks, so [Follow me there](#) if you found this interesting and want to learn more.



Rik Schennink
Indie Product Developer

Latest PQINA Plugins



High-performance JavaScript image editing in the browser. Cropping, rotating, flipping, filtering, resizing, annotating, and client-side transforming of images, it's all effortless with Pintura Image Editor.

[Pintura Image Editor product page](#)



A JavaScript library that uploads anything you throw at it, optimizes images for faster uploads, and offers a great, accessible, silky smooth user experience.

[FilePond product page](#)