

Advanced Thumbnail Creator

1 The problem

When working with the news feeds or just with information that contains graphic images, there is often a necessity to generate a preview. At a first sight, nothing can be simpler than to create a reduced in size image of an arbitrary picture. However, it's only at a first sight.

2 Existing solutions

There are few methods for solving this problem:

- a) Use of the original image as a thumbnail by means of reducing it in `` properties.
- b) Reduction of the image size with the help of the `imagecopyresampled()` and `imagecopyresized()` functions (GD library).
- c) Selection of the central region of the image and its subsequent transformation by the functions of the GD library.
- d) Generation of the thumbnail with the help of graphic applications on a local machine and its subsequent uploading to the server.

Obvious advantage of the first three methods is the complete process automation. Disadvantages: in the first case, the image is loaded in its original size (Fig. 2a); in the second - the original image proportions are distorted in the thumbnail (Fig. 2b); in the third - the central part of an image isn't always a "meaningful" part (Fig. 2c).

The fourth method achieves necessary results (Fig. 2d), but adds superfluous difficulties during switching between applications, since there is a necessity of loading of at least two files in place of one.



Figure 1 - Initial image



Figure 2 – Thumbnail generation result (70x70)

3 The solution

Essence of the method is an automatic thumbnail generation with manual selection of the "meaningful" region in a web-browser.

4 Used technologies

PHP, Javascript, DOM, CSS, HTML

5 Cross-browser

Internet Explorer 6.0,7.0; FireFox 1.0,2.0; Opera 9.0; Safari 2.0;

6 Input and output parameters

Input parameters:

\$_GET[filename] - is a string which determines the file name and the file path in relation to a root directory on the server. Variable \$_GET[filename] must be pre-coded by urlencode() function;

\$_GET[thumbwidth] - is the pre-set width of the thumbnail. If this parameter is equal to "0" or not defined, the thumbnail width is determined from the chosen region width.

\$_GET[thumbheight] - is the pre-set height of the thumbnail. If this parameter is equal to "0" or not defined, the thumbnail height is determined from the chosen region height.

Output parameters:

Thumbnail images in the web-browser.

JPEG-file placed in the same catalogue where the initial image was.

7 Implementation

The developed program complex consists of two files.

1. thumbcreate.js – a set of functions to work with the selected region and a function for thumbnail generation.

2. thumbcreate.php - contains the preview generation function, the working area block and the input-output form .

7.1. Thumbcreate.js functions description.

```
//initialization
function init()
{
    document.getElementById("image").onclick=mouseHandler;
    document.getElementById("image").onmousemove=mouseHandler;
    document.getElementById('th_w').value=thumbWidth;
    document.getElementById('th_h').value=thumbHeight;
}

//mouse handler
function mouseHandler(mouseEvent)
{
    if (!mouseEvent) mouseEvent = window.event;
    if (mouseEvent.button == 2) return;
    var element = (mouseEvent.target)?mouseEvent.target:mouseEvent.srcElement;

    //for a clique we begin to draw a selection rectangle
    if (mouseEvent.type=="click")
    {
        var x = mouseEvent.clientX - document.getElementById("image").offsetLeft;
```

```

        var at = mouseEvent.clientY - document.getElementById("image").offsetTop;
        pointSet(x,y);
        rectangleDraw('area');
    };

//draw the selection region during mouse motion
    if (mouseEvent.type=="mousemove")
    {
        .
    }
}

//setting coordinates for top left and right bottom corner
function pointSet(x,y)
{
    if (!ptype)
    {
        x1=x+document.body.scrollLeft;
        y1=y+document.body.scrollTop;
        rectangleHide('area');
        inputUpdate();
    }
    else
    {
        x2=x+document.body.scrollLeft;
        y2=y+document.body.scrollTop;
        pointCorrect();
        inputUpdate();
    }
    ptype = !ptype;
}

//correcting TL BR coordinates if they are switched
function pointCorrect(x1c,y1c,x2c,y2c)
{
    .
}

//rectangle drawing (x1,y1); (x2,y2)
function rectangleDraw(rectId)
{
    .
}

// rectangle drawing from input fields
function rectangleDrawInput(rectId)
{
    .
}

//rectangle hiding
function rectangleHide(rectId)
{
    .
}

//input fields update functions
function inputUpdate()
{
    .
}

function inputWidthUpdate()
{
    .
};

function inputHeightUpdate()
{
    .
};

```

```

function inputXYUpdate()
{
    .
};

//prepare for thumbnail generation
function generateImageThumb()
{
    var previewclass='preview';
    var previewimage='/preview.gif';

    var links=document.getElementsByTagName('a');
    var prevlinks=new Array();
    var c=0;

    var previewTest = new RegExp("(^|\\s)" + previewclass + "(\\s|$)");

    for(I=0; I<links.length; I++)
    {
        if (previewTest.test(links[I].className))
            prevlinks[c]=links[I]; c++; }

    for(I=0; I<prevlinks.length; I++)
    {

        var newa=document.createElement('a');
        newa.style.textDecoration="none";

        var newbutton=document.createElement('input');
        newbutton.type="button";
        newbutton.value="Generate Thumbnail"

        newa.appendChild(newbutton);
        newa.href="#";

        var newbr=document.createElement('br');
        newa.appendChild(newbr);

        newa.onclick=function()
        {
            if(this.getElementsByTagName('img')[0])
                this.removeChild(this.getElementsByTagName('img')[0]);

            var newimg=document.createElement('img');
            newimg.style.border="0";
            newimg.vspace="10";
            this.appendChild(newimg);

            var rand=parseInt(1000*Math.random());
            newimg.src="?action=generate&r="+rand+"&x1="+document.getElementById('x1_inp').value+"&y1="+document.getElementById('y1_inp').value+"&x2="+document.getElementById('x2_inp').value+"&y2="+document.getElementById('y2_inp').value+"&w="+document.getElementById('th_w').value+"&h="+document.getElementById('th_h').value+"&fn="+document.getElementById('fileName').value;

            return false;
        }

        prevlinks[I].parentNode.insertBefore(newa,prevlinks[I].nextSibling);
    }

}

//initialization
window.onload=function()
{
    init();
    generateImageThumb();
}

```

7.2. Thumbcreate.php functions description.

```
<?php
    if (!$_GET[thumbwidth]) $_GET[thumbwidth]=0;
    if (!$_GET[thumbheight]) $_GET[thumbheight]=0;

//thumbnail creation function
    function createthumb($new_w,$new_h,$x1,$y1,$x2,$y2,$fn)
    {
        $src_img=imagecreatefromjpeg($_GET['fn']);
        $dst_img=ImageCreateTrueColor($new_w,$new_h);
        imagecopyresampled($dst_img,$src_img,0,0,$x1,$y1,$new_w,$new_h,$x2-$x1,$y2-$y1);

//output to browser
        imagejpeg($dst_img);
//output to file
        imagejpeg($dst_img,substr($_GET['fn'],0,-4)."_th.jpg");

        imagedestroy($dst_img);
        imagedestroy($src_img);
    }
    if ($_GET[action]=="generate")
    {
        header('Content-type:image/jpeg');
        createthumb($_GET['w'],$_GET['h'],$_GET['x1'],$_GET['y1'],$_GET['x2'],$_GET['y2'],
$_GET['fn']);
    }
?>

//working area
<div id="image" style="cursor: crosshair; width:<?= $imgWidth ?>px; height:<?=
$imgHeight ?>px; border-width:0; background-image: url('<?= $_GET[filename] ?>')
"><div id="area"></div></div>

//input-output form
<form action="" method="post">
<table width="<?=$imgWidth?>px">
    <tr>
        <td width="33%" align=center>
            TL (x1, y1) = (<input type="text" id="x1_inp" value="0" class="atc1"
onChange="inputXYUpdate();">,<input type="text" id="y1_inp" value="0" class="atc1"
onChange="inputXYUpdate();">)<br>
            BR (x2, y2) = (<input type="text" id="x2_inp" value="0" class="atc1"
onChange="inputXYUpdate();">,<input type="text" id="y2_inp" value="0" class="atc1"
onChange="inputXYUpdate();">)<br></td>
        <td width="33%" align=center>
            Width = <input type="text" id="th_width" value="0" class="atc2"
onChange="inputWidthUpdate();"><br>
            Height = <input type="text" id="th_height" value="0" class="atc2"
onChange="inputHeightUpdate();"><br></td>
        <td width="33%" align=center>
            Thumbnail width = <input type="text" id="th_w" value="" class="atc2"><br>
            Thumbnail height = <input type="text" id="th_h" value=""
class="atc2"><br></td></tr>
        <tr><td colspan=3 align=center><BR><a href="/none.gif"
class="preview"></a></td></tr>
</table><input type="hidden" value="<?= $fileInput ?>" id="fileName"></form>
```

8 Example

For the “meaning” region selection you must determine two angular points of rectangle (fig. 3). During the pointer motion on a working area values in formfields are changed: TL (x_1, y_1) are coordinates of the first angular point, BR (x_2, y_2) are coordinates of the second angular point, Width is width of the selection region, Height is the height of selection region. Thumbnail Width, Thumbnail Height – width and height of generated thumbnail. All mentioned parameters can be edited. After pressing “Generate Thumbnail” button you’ll get thumbnail in your browser and thumbnail file on server.



TL (x_1, y_1) = (211 , 66)
BR (x_2, y_2) = (677 , 532)

Width = 466
Height = 466

Thumbnail width = 150
Thumbnail height = 150

Generate Thumbnail



Figure 3 - Example

9 References

1. Image previews with DOM JavaScript - <http://icant.co.uk/articles/imagepreview/>
2. Advanced scaling of images in PHP (rus) - <http://www.codenet.ru/webmast/php/Image-Resize-GD/>

Andrew Zhupanenko
research@zhupanenko.com
<http://research.zhupanenko.com/atc/>

February, 2007