

JavaScript Popup Generator



Disclaimer

This SOFTWARE PRODUCT is provided by El Condor "as is" and "with all faults." El Condor makes no representations or warranties of any kind concerning the safety, suitability, lack of viruses, inaccuracies, typographical errors, or other harmful components of this SOFTWARE PRODUCT. There are inherent dangers in the use of any software, and you are solely responsible for determining whether this SOFTWARE PRODUCT is compatible with your equipment and other software installed on your equipment. You are also solely responsible for the protection of your equipment and backup of your data, and El Condor will not be liable for any damages you may suffer in connection with using, modifying, or distributing this SOFTWARE PRODUCT.

You can use this SOFTWARE PRODUCT freely, if you would you can credit me in program comment:

El Condor – CONDOR INFORMATIQUE – Turin

Comments, suggestions and criticisms are welcomed: mail to rossati@libero.it

Conventions

Commands syntax, instructions in programming language and examples are with font COURIER NEW. The optional parties of syntactic explanation are contained between [square parentheses], alternatives are separated by | and the variable parties are in *italics*.

Contents table

1 PopUp generator.....	1
1.1 Using the Popup generator.....	1
1.2 Compatibility.....	1
1.3 How works.....	1
1.3.1 Popup creation.....	1
1.3.2 Popup closing.....	1
1.4 Parameters.....	1
1.4.1 Cancel.....	1
1.4.2 Class.....	2
1.4.3 Content.....	2
1.4.4 Fade.....	2
1.4.5 Id.....	2
1.4.6 Left and Top.....	2
1.4.7 Movable.....	2
1.4.8 Style.....	2
1.4.9 Width and Height.....	2
2 Examples.....	2
2.1 Center.....	2
2.2 Cancel by Escape key.....	2
2.3 Cancel a permanent popup.....	2
2.4 Fading.....	3
2.5 Show random image.....	3
2.6 Popup of text and images.....	3
3 Annexes.....	4
3.1 Sandbox.....	4
3.2 PHP script for get a random image.....	4
4 History.....	5

1 PopUp generator

This is a documentation of the JavaScript `popUpGen.js` script for generate popups.

1.1 Using the Popup generator

The popup is created by calling the function:

```
link = popUp(parameters)
```

where `parameters` is an associative array that contains the data necessary to personalize the popup; the function returns an object, `link`, that points to the popup generated.

Below a list of parameters,  the key parameters are indifferent to the case.

Name	Default value	Note
cancel	click	Cancel by mouse click or key (see here for values), or mouseout or number of milliseconds
class	empty	
content	empty	Is the content that must be shown
fade	empty	From 0 to two numeric values, the first one is fade in, the second one the fade out; the values are milliseconds
height	-1	For images, on default the height is automatically determined
id	empty	
left	-1	The default value forces horizontal centering
movable	false	To make the popup movable
top	-1	The default value forces vertical centering
style	empty	
width	-1	For images, on default the width is automatically determined

1.2 Compatibility

Needs HTML5 for access the size of images.

1.3 How works

1.3.1 Popup creation

The popup is inserted in the HTML tag indicated by the parameter `id` or in a `div` tag newly created if the `id` is not indicated or it does not exist; the `div` is inserted as child of body:

```
var link = document.body.appendChild(document.createElement("div"));
```

and style:

```
position:absolute;display:inline-block;
```

1.3.2 Popup closing

The popup is removed, unless the `cancel` parameter is `no`, when a key is typed or mouse event occurs (see paragraph [1.4.1 Cancel](#)).

If the popup is into an existent HTML tag, i.e. the `id` exists, the inner content is cleared, otherwise the `div` created is removed.

1.4 Parameters

1.4.1 Cancel

`click|no|mouseout|key|number`

This indicates how the popup can be closed:

- `number` is a milliseconds of the popup life, `number` is inclusive of any `fade` duration;

- `no` means that the popup remains on the page (see paragraph 2.3 *Cancel a permanent popup*);
- `mouseout` indicates the popup disappears when the mouse comes out of the popup.
- key see [here](#) for values; some possible keys are: `Delete`, `Enter`, `Escape`, `Tab`.

`Click`, `mouseout` and `no` are indifferent to case.

1.4.2 Class

Adds a `class` to the popup container.

1.4.3 Content

Is the popup content that is inserted on the `div` created.

Content can be a name of a file image of type `.jpg`, `.jpeg`, `.png` or `.gif` or any HTML fragment.

1.4.4 Fade

Can contains up to two numbers space separated: the first one is fade in time, the second one the fade out time; the time values are in milliseconds; if only fade out is requested the fade in must be present with value 0, for example: `{content: '...', fade: "0 2000", ...}`

1.4.5 Id

Is the `id` of the element where the `content` of the popup will be inserted.

 The lack of `id` or an `id` nonexistent is the necessary condition for a movable popup.

1.4.6 Left and Top

These parameters are taken into account if a container `id` is not indicated or this doesn't exist.

If the value are `-1` (the default) the popup is centered.

1.4.7 Movable

The parameter is taken into consideration if the parameter `id` is omitted or the `id` is not existent; if `movable:true` the popup can be manually moved.

1.4.8 Style

Add a style to a popup container, for example: `"style":"text-align: center"`.

1.4.9 Width and Height

They determine the size of the popup,  it makes sense for popups with images.

A value `-1` doesn't change the relative dimension.

Setting `width` and `height` at the same time can possibly distort image, vice versa, setting one of the parameters maintains the proportions of a possible image.

2 Examples

2.1 Center

Due to the default values of `top` and `left` parameters it is only necessary the `content`. The popup disappears by a mouse `click`.

```
popUp({content: 'images/rabbitLake.png'})
```

2.2 Cancel by Escape key

In addition to the `content`, the `cancel` parameter must be set to `Escape`.

```
popUp({content: '...', cancel:'Escape', ...})
```

2.3 Cancel a permanent popup

```
PopupDiv = popUp({content: '...', cancel:'No', ...})
...
var remove = x => x.remove();
```

```
...
remove(popUpDiv);
```

2.4 Fading

The fade can affect both the display and the disappearance of the popup.

```
popUp({'content': '...', 'fade':'2000',...})      // fade in two seconds
popUp({'content': '...', 'fade':'0 2000',...})    // fade out two seconds
```

2.5 Show random image

The example presupposes an application on the server, in the example below a PHP script.

```
Var parms = {...}
parms.content = "<img src='getImage.php?R="+Date.now()+"'>";
```

 The ?R="+Date.now() is to avoid the image caching.

See the paragraph 3.2 for an example of sending a random image by a PHP script on server.

2.6 Popup of text and images

```
var complex =
<table>
<thead>
<tr>
    <th colspan=3>Form Generator for B4A</th>
</tr>
</thead>
<tr>
<td></td>
<td><ul>
<li>Has been added a Timer management,
</li><li>A view Toggle, possibly with image with more than two states,
</li><li>Added CKL, a check boxes list;
</li><li>%#months returns a month number from 1 to 12,
</li><li>an error on Reset button in Tabs and Section has been corrected.
</li></ul></td>
<td></td>
</table>
...
Var parms = {...}
parms.content = complex;
```

3 Annexes

3.1 Sandbox

In the site <https://www.condorinformatique.com/PopUp/index.html> there is a demo useful for test the capabilities of the package.

3.2 PHP script for get a random image

```
<?php
function randomImg($folder) {
    $aImages = chooseImages($folder);
    return $aImages[rand(0, count($aImages)-1)];
}
function chooseImages($folder) {
    $aImages = glob($folder . '/*.{jpg,jpeg,png,gif}', GLOB_BRACE);
    $aBigImages = [];
    foreach($aImages as $k => $file) {
        if (filesize($file) > 10000) array_push($aBigImages, $aImages[$k]);
    }
    return $aBigImages;
}
if (isset($_REQUEST["List"])){
    $aImages = chooseImages("images");
    $re = '/^.*\/(.*)\..*$mi';
    foreach($aImages as $k => $file) {
        preg_match($re, $aImages[$k], $matches);
        $aImages[$k] .= ".$matches[1]";
    }
    echo implode(", ", $aImages);
} else {
    $fileImage = randomImg("images");
    $aMimeTypes = array('jpeg' => 'image/jpeg',
                        'jpg' => 'image/jpeg',
                        'png' => 'image/png',
                        'gif' => 'image/gif');
    $path_parts = pathinfo($fileImage);
    header("Content-type: ".$aMimeTypes[$path_parts['extension']]);
    readfile("$fileImage");
}
exit(0);
?>
```

4 History

0.1.0	30 March 2020	First version
0.1.1	26 April 2020	Correct resizing of image
0.1.2	September 2024	<ul style="list-style-type: none">• Added the parameter <code>movable</code>,• the function now returns the object div created that contains the popup.• the center of popup is correct.
0.1.3	October 2024	<ul style="list-style-type: none">• Correct positioning of the popup on the screen.