

prettyphoto Plugin

Es gab Probleme (1.10.2020), so dass die Bilder nicht mehr richtig angezeigt wurden. Das Problem wurde behoben und kann auf der GitHub Seite herunter geladen werden:

Aktuelle Seite:

<https://github.com/ssahara/dw-plugin-prettyPhoto>

Info über das Plugin:

<https://www.dokuwiki.org/plugin:prettyphoto>

Aktuelle Version 02.10.2020:

[dw-plugin-prettyphoto-master.zip](#)

Forum:

<https://forum.dokuwiki.org/d/18308-prettyphoto-do-not-work-anymore/5>

Info:

Evtl. in den Einstellungen „defer_js“ deaktivieren.

modifizierte jquery.prettyPhoto.js

Verzeichnis /dokuwiki/lib/plugins/prettyphoto/js/jquery.prettyPhoto.js

[jquery.prettyPhoto.js](#)

```
/* -----  
----  
Class: prettyPhoto  
Use: Lightbox clone for jQuery  
Copyright: GPLv2 or Creative Commons 2.5 license  
Author: Stephane Caron (http://www.no-margin-for-errors.com)  
Version: 3.1.7 - replaced size() with length property  
-----  
-- */  
(function($) {  
    $.prettyPhoto = {version: '3.1.7'};  
  
    $.fn.prettyPhoto = function(pp_settings) {  
        pp_settings = jQuery.extend({  
            hook: 'rel', /* the attribute tag to use for prettyPhoto  
hooks. default: 'rel'. For HTML5, use "data-rel" or similar. */  
            animation_speed: 'fast', /* fast/slow/normal */  
            ajaxcallback: function() {},  
            slideshow: 5000, /* false OR interval time in ms */  
            autoplay_slideshow: false, /* true/false */  
            opacity: 0.80, /* Value between 0 and 1 */  
            show_title: false, /* true/false */  
            allow_resize: true, /* Resize the photos bigger than
```

```

viewport. true/false */
    allow_expand: true, /* Allow the user to expand a resized
image. true/false */
    default_width: 500,
    default_height: 344,
    counter_separator_label: '/', /* The separator for the
gallery counter 1 "of" 2 */
    theme: 'pp_default', /* light_rounded / dark_rounded /
light_square / dark_square / facebook */
    horizontal_padding: 5, /* The padding on each side of the
picture */
    hideflash: false, /* Hides all the flash object on a page,
set to TRUE if flash appears over prettyPhoto */
    wmode: 'opaque', /* Set the flash wmode attribute */
    autoplay: true, /* Automatically start videos: True/False
*/
    modal: false, /* If set to true, only the close button will
close the window */
    deeplinking: true, /* Allow prettyPhoto to update the url
to enable deeplinking. */
    overlay_gallery: false, /* If set to true, a gallery will
overlay the fullscreen image on mouse over */
    overlay_gallery_max: 30, /* Maximum number of pictures in
the overlay gallery */
    keyboard_shortcuts: true, /* Set to false if you open forms
inside prettyPhoto */
    changepicturecallback: function({}), /* Called everytime an
item is shown/changed */
    callback: function({}), /* Called when prettyPhoto is
closed */
    ie6_fallback: true,
    markup: '<div class="pp_pic_holder"> \
        <div class="ppt">&nbsp;</div> \
        <div class="pp_top"> \
            <div class="pp_left"></div> \
            <div class="pp_middle"></div> \
            <div class="pp_right"></div> \
        </div> \
        <div class="pp_content_container"> \
            <div class="pp_left"> \
            <div class="pp_right"> \
                <div class="pp_content"> \
                    <div class="pp_loaderIcon"></div> \
                    <div class="pp_fade"> \
                        <a href="#" class="pp_expand"
style="left: 30px;" title="Expand the image">Expand</a> \
                        <a class="pp_close" style="z-
index:99999; right:30px; top:10px;" href="#">Close</a> \
                        <div class="pp_hoverContainer">
\
                            <a class="pp_next"

```

```

href="#">next</a> \
                                <a class="pp_previous"
href="#">previous</a> \
                                </div> \
                                <div id="pp_full_res"></div> \
                                <div class="pp_details"> \
                                    <div class="pp_nav"> \
                                        <a href="#"
class="pp_arrow_previous">Previous</a> \
                                        <p
class="currentTextHolder">0/0</p> \
                                        <a href="#"
class="pp_arrow_next">Next</a> \
                                        </div> \
                                        <p
class="pp_description"></p> \
                                        <div
class="pp_social">{pp_social}</div> \
                                        <!--<a class="pp_close"
href="#">Close</a>--> \
                                            </div> \
                                            </div> \
                                            </div> \
                                            </div> \
                                            </div> \
                                            <div class="pp_bottom"> \
                                                <div class="pp_left"></div> \
                                                <div class="pp_middle"></div> \
                                                <div class="pp_right"></div> \
                                            </div> \
                                        </div> \
                                        <div class="pp_overlay"></div>',
gallery_markup: '<div class="pp_gallery"> \
    <a href="#"
class="pp_arrow_previous">Previous</a> \
        <div> \
            <ul> \
                {gallery} \
            </ul> \
        </div> \
        <a href="#"
class="pp_arrow_next">Next</a> \
            </div>',
image_markup: '',
flash_markup: '<object classid="clsid:D27CDB6E-
AE6D-11cf-96B8-444553540000" width="{width}" height="{height}"><param
name="wmode" value="{wmode}" /><param name="allowfullscreen"
value="true" /><param name="allowscriptaccess" value="always" /><param
name="movie" value="{path}" /><embed src="{path}" type="application/x-
shockwave-flash" allowfullscreen="true" allowscriptaccess="always"

```

```

width="{width}" height="{height}" wmode="{wmode}"></embed></object>',
    quicktime_markup: '<object
classid="clsid:02BF25D5-8C17-4B23-BC80-D3488ABDDC6B"
codebase="http://www.apple.com/qtactivex/qtplugin.cab"
height="{height}" width="{width}"><param name="src"
value="{path}"><param name="autoplay" value="{autoplay}"><param
name="type" value="video/quicktime"><embed src="{path}"
height="{height}" width="{width}" autoplay="{autoplay}"
type="video/quicktime"
pluginspage="http://www.apple.com/quicktime/download/"></embed></object
>',
    iframe_markup: '<iframe src="{path}" width="{width}"
height="{height}" frameborder="no"></iframe>',
    inline_markup: '<div class="pp_inline">{content}</div>',
    custom_markup: '',
    social_tools: false /* html or false to disable */
}, pp_settings);

// Global variables accessible only by prettyPhoto
var matchedObjects = this, percentBased = false, pp_dimensions,
pp_open,

// prettyPhoto container specific
pp_contentHeight, pp_contentWidth, pp_containerHeight,
pp_containerWidth,

// Window size
windowHeight = $(window).height(), windowWidth =
$(window).width(),

// Global elements
pp_slideshow;

doresize = true, scroll_pos = _get_scroll();

// Window/Keyboard events
$(window).unbind('resize.prettyphoto').bind('resize.prettyphoto',function(){
    _center_overlay(); _resize_overlay(); });

if(pp_settings.keyboard_shortcuts) {
$(document).unbind('keydown.prettyphoto').bind('keydown.prettyphoto',function(e){
    if(typeof $pp_pic_holder != 'undefined'){
        if($pp_pic_holder.is(':visible')){
            switch(e.keyCode){
                case 37:
                    $.prettyPhoto.changePage('previous');
                    e.preventDefault();
                    break;
                case 39:
                    $.prettyPhoto.changePage('next');
            }
        }
    }
});
}

```

```

        e.preventDefault();
        break;
    case 27:
        if(!settings.modal)
            $.prettyPhoto.close();
        e.preventDefault();
        break;
    };
    // return false;
};
});
});
});

/**
 * Initialize prettyPhoto.
 */
$.prettyPhoto.initialize = function() {

    settings = pp_settings;

    if(settings.theme == 'pp_default')
settings.horizontal_padding = 16;

    // Find out if the picture is part of a set
    theRel = $(this).attr(settings.hook);
    galleryRegExp = /\[(?:.*)\]/;
    isSet = (galleryRegExp.exec(theRel)) ? true : false;

    // Put the SRCs, TITLES, ALTs into an array.
    pp_images = (isSet) ? jQuery.map(matchedObjects,
function(n, i){ if($(n).attr(settings.hook).indexOf(theRel) != -1)
return $(n).attr('href'); }) : $.makeArray($(this).attr('href'));
    pp_titles = (isSet) ? jQuery.map(matchedObjects,
function(n, i){ if($(n).attr(settings.hook).indexOf(theRel) != -1)
return ($(n).find('img').attr('alt')) ? $(n).find('img').attr('alt') :
""; }) : $.makeArray($(this).find('img').attr('alt'));
    pp_descriptions = (isSet) ? jQuery.map(matchedObjects,
function(n, i){ if($(n).attr(settings.hook).indexOf(theRel) != -1)
return $(n).attr('title') ? $(n).attr('title') : ""; }) :
$.makeArray($(this).attr('title'));

    if(pp_images.length > settings.overlay_gallery_max)
settings.overlay_gallery = false;

    set_position = jQuery.inArray($(this).attr('href'),
pp_images); // Define where in the array the clicked item is
positionned
    rel_index = (isSet) ? set_position :
$("a["+settings.hook+"^='"+theRel+"'']").index($(this));

```

```

        _build_overlay(this); // Build the overlay {this} being the
caller

        if(settings.allow_resize)
            $(window).bind('scroll.prettyphoto',function(){
        _center_overlay(); });

        $.prettyPhoto.open();

        return false;
    }

    /**
    * Opens the prettyPhoto modal box.
    * @param image {String,Array} Full path to the image to be
open, can also be an array containing full images paths.
    * @param title {String,Array} The title to be displayed with
the picture, can also be an array containing all the titles.
    * @param description {String,Array} The description to be
displayed with the picture, can also be an array containing all the
descriptions.
    */
    $.prettyPhoto.open = function(event) {
        if(typeof settings == "undefined"){ // Means it's an API
call, need to manually get the settings and set the variables
            settings = pp_settings;
            pp_images = $.makeArray(arguments[0]);
            pp_titles = (arguments[1]) ? $.makeArray(arguments[1])
: $.makeArray("");
            pp_descriptions = (arguments[2]) ?
$.makeArray(arguments[2]) : $.makeArray("");
            isSet = (pp_images.length > 1) ? true : false;
            set_position = (arguments[3])? arguments[3]: 0;
            _build_overlay(event.target); // Build the overlay
{this} being the caller
        }

        if(settings.hideflash)
        $('object,embed,iframe[src*=youtube],iframe[src*=vimeo]').css('visibili
ty','hidden'); // Hide the flash

        _checkPosition($(pp_images).length); // Hide the
next/previous links if on first or last images.

        $('.pp_loaderIcon').show();

        if(settings.deeplinking)
            setHashtag();

```

```

        // Rebuild Facebook Like Button with updated href
        if(settings.social_tools){
            facebook_like_link =
settings.social_tools.replace('{location_href}',
encodeURIComponent(location.href));
$pp_pic_holder.find('.pp_social').html(facebook_like_link);
        }

        // Fade the content in
        if($ppt.is(':hidden')) $ppt.css('opacity',0).show();
$pp_overlay.show().fadeTo(settings.animation_speed,settings.opacity);

        // Display the current position
$pp_pic_holder.find('.currentTextHolder').text((set_position+1) +
settings.counter_separator_label + $(pp_images).length);

        // Set the description
        if(typeof pp_descriptions[set_position] != 'undefined' &&
pp_descriptions[set_position] != ""){
$pp_pic_holder.find('.pp_description').show().html(unescape(pp_descript
ions[set_position]));
        }else{
            $pp_pic_holder.find('.pp_description').hide();
        }

        // Get the dimensions
        movie_width = (
parseFloat(getParam('width',pp_images[set_position])) ) ?
getParam('width',pp_images[set_position]) :
settings.default_width.toString();
        movie_height = (
parseFloat(getParam('height',pp_images[set_position])) ) ?
getParam('height',pp_images[set_position]) :
settings.default_height.toString();

        // If the size is % based, calculate according to window
dimensions
        percentBased=false;
        if(movie_height.indexOf('%') != -1) { movie_height =
parseFloat(($window).height() * parseFloat(movie_height) / 100) -
150); percentBased = true; }
        if(movie_width.indexOf('%') != -1) { movie_width =
parseFloat(($window).width() * parseFloat(movie_width) / 100) - 150);
percentBased = true; }

        // Fade the holder
$pp_pic_holder.fadeIn(function(){
    // Set the title
        (settings.show_title && pp_titles[set_position] != ""
&& typeof pp_titles[set_position] != "undefined") ?
$ppt.html(unescape(pp_titles[set_position])) : $ppt.html('&nbsp;');

```

```
imgPreloader = "";
skipInjection = false;

// Inject the proper content
switch(_getFileType(pp_images[set_position])){
    case 'image':
        imgPreloader = new Image();

        // Preload the neighbour images
        nextImage = new Image();
        if(isSet && set_position < $(pp_images).length
-1) nextImage.src = pp_images[set_position + 1];
        prevImage = new Image();
        if(isSet && pp_images[set_position - 1])
prevImage.src = pp_images[set_position - 1];

$pp_pic_holder.find('#pp_full_res')[0].innerHTML =
settings.image_markup.replace(/{\path}/g,pp_images[set_position]);

        imgPreloader.onload = function(){
            // Fit item to viewport
            pp_dimensions =
_fitToViewport(imgPreloader.width,imgPreloader.height);

            _showContent();
        };

        imgPreloader.onerror = function(){
            alert('Image cannot be loaded. Make sure
the path is correct and image exist.');
```

```
            $.prettyPhoto.close();
        };

        imgPreloader.src = pp_images[set_position];
        break;

        case 'youtube':
            pp_dimensions =
_fitToViewport(movie_width,movie_height); // Fit item to viewport

            // Regular youtube link
            movie_id =
getParam('v',pp_images[set_position]);

            //youtu.be link
            if(movie_id == ""){
                movie_id =
pp_images[set_position].split('youtu.be/');
                movie_id = movie_id[1];
                if(movie_id.indexOf('?') > 0)
```

```

        movie_id =
movie_id.substr(0,movie_id.indexOf('?')); // Strip anything after the ?

        if(movie_id.indexOf('&') > 0)
            movie_id =
movie_id.substr(0,movie_id.indexOf('&')); // Strip anything after the &
        }

        movie =
'http://www.youtube.com/embed/'+movie_id;
        (getParam('rel',pp_images[set_position])) ?
movie+="?rel="+getParam('rel',pp_images[set_position]) :
movie+="?rel=1";

        if(settings.autoplay) movie += "&autoplay=1";

        toInject =
settings.iframe_markup.replace(/width/g,pp_dimensions['width']).replace(/height/g,pp_dimensions['height']).replace(/wmode/g,settings.wmode).replace(/path/g,movie);
        break;

        case 'vimeo':
            pp_dimensions =
_fitToViewport(movie_width,movie_height); // Fit item to viewport

            movie_id = pp_images[set_position];
            var regExp =
/http(s?):\\/(www\\.)?vimeo\\.com/(\\d+)/;
            var match = movie_id.match(regExp);

            movie = 'http://player.vimeo.com/video/'+
match[3] +'?title=0&amp;byline=0&amp;portrait=0';
            if(settings.autoplay) movie += "&autoplay=1";

            vimeo_width = pp_dimensions['width'] +
'/embed/?moog_width='+ pp_dimensions['width'];

            toInject =
settings.iframe_markup.replace(/width/g,vimeo_width).replace(/height/g,pp_dimensions['height']).replace(/path/g,movie);
            break;

            case 'quicktime':
                pp_dimensions =
_fitToViewport(movie_width,movie_height); // Fit item to viewport
                pp_dimensions['height']+=15;
pp_dimensions['contentHeight']+=15;
pp_dimensions['containerHeight']+=15; // Add space for the control bar

                toInject =

```

```
settings.quicktime_markup.replace(/{\width}/g,pp_dimensions['width']).re
place(/{\height}/g,pp_dimensions['height']).replace(/{\wmode}/g,settings.
wmode).replace(/{\path}/g,pp_images[set_position]).replace(/{\autoplay}/g
,settings.autoplay);
        break;

        case 'flash':
            pp_dimensions =
_fitToViewport(movie_width,movie_height); // Fit item to viewport

            flash_vars = pp_images[set_position];
            flash_vars =
flash_vars.substring(pp_images[set_position].indexOf('flashvars') +
10,pp_images[set_position].length);

            filename = pp_images[set_position];
            filename =
filename.substring(0,filename.indexOf('?'));

            toInject =
settings.flash_markup.replace(/{\width}/g,pp_dimensions['width']).replac
e(/{\height}/g,pp_dimensions['height']).replace(/{\wmode}/g,settings.wmod
e).replace(/{\path}/g,filename+'?'+flash_vars);
            break;

            case 'iframe':
                pp_dimensions =
_fitToViewport(movie_width,movie_height); // Fit item to viewport

                frame_url = pp_images[set_position];
                frame_url =
frame_url.substr(0,frame_url.indexOf('iframe')-1);

                toInject =
settings.iframe_markup.replace(/{\width}/g,pp_dimensions['width']).repla
ce(/{\height}/g,pp_dimensions['height']).replace(/{\path}/g,frame_url);
                break;

                case 'ajax':
                    doresize = false; // Make sure the dimensions
are not resized.

                    pp_dimensions =
_fitToViewport(movie_width,movie_height);
                    doresize = true; // Reset the dimensions

                    skipInjection = true;
$.get(pp_images[set_position],function(responseHTML){
            toInject =
settings.inline_markup.replace(/{\content}/g,responseHTML);
$pp_pic_holder.find('#pp_full_res')[0].innerHTML = toInject;
            _showContent();
```

```

        });

        break;

        case 'custom':
            pp_dimensions =
            _fitToViewport(movie_width,movie_height); // Fit item to viewport

            toInject = settings.custom_markup;
            break;

        case 'inline':
            // to get the item height clone it, apply
default width, wrap it in the prettyPhoto containers , then delete
            myClone =
            $(pp_images[set_position]).clone().append('<br clear="all"
            />').css({'width':settings.default_width}).wrapInner('<div
            id="pp_full_res"><div
            class="pp_inline"></div></div>').appendTo($('body')).show();
            doresize = false; // Make sure the dimensions
are not resized.

            pp_dimensions =
            _fitToViewport($(myClone).width(),$(myClone).height());
            doresize = true; // Reset the dimensions
            $(myClone).remove();
            toInject =
            settings.inline_markup.replace(/{content}/g,$(pp_images[set_position]).
            html());

            break;
        };

        if(!imgPreloader && !skipInjection){
            $pp_pic_holder.find('#pp_full_res')[0].innerHTML =
toInject;

            // Show content
            _showContent();
        };
    });

    return false;
};

/**
 * Change page in the prettyPhoto modal box
 * @param direction {String} Direction of the paging, previous
or next.
 */
$.prettyPhoto.changePage = function(direction){
    currentGalleryPage = 0;

```

```

        if(direction == 'previous') {
            set_position--;
            if (set_position < 0) set_position =
$(pp_images).length - 1;
        }else if(direction == 'next'){
            set_position++;
            if(set_position > $(pp_images).length - 1) set_position
= 0;
        }else{
            set_position=direction;
        };

        rel_index = set_position;

        if(!doresize) doresize = true; // Allow the resizing of the
images

        if(settings.allow_expand) {
            $($('.pp_contract').removeClass('pp_contract').addClass('pp_expand'));
        }

        _hideContent(function(){ $.prettyPhoto.open(); });
    };

    /**
     * Change gallery page in the prettyPhoto modal box
     * @param direction {String} Direction of the paging, previous
or next.
     */
    $.prettyPhoto.changeGalleryPage = function(direction){
        if(direction=='next'){
            currentGalleryPage ++;

            if(currentGalleryPage > totalPage) currentGalleryPage =
0;
        }else if(direction=='previous'){
            currentGalleryPage --;

            if(currentGalleryPage < 0) currentGalleryPage =
totalPage;
        }else{
            currentGalleryPage = direction;
        };

        slide_speed = (direction == 'next' || direction ==
'previous') ? settings.animation_speed : 0;

        slide_to = currentGalleryPage * (itemsPerPage * itemWidth);

        $pp_gallery.find('ul').animate({left:-

```

```
slide_to},slide_speed);
};

/**
 * Start the slideshow...
 */
$.prettyPhoto.startSlideshow = function(){
    if(typeof pp_slideshow == 'undefined'){
        $pp_pic_holder.find('.pp_play').unbind('click').removeClass('pp_play').
        addClass('pp_pause').click(function(){
            $.prettyPhoto.stopSlideshow();
            return false;
        });
        pp_slideshow =
        setInterval($.prettyPhoto.startSlideshow,settings.slideshow);
    }else{
        $.prettyPhoto.changePage('next');
    }
}

/**
 * Stop the slideshow...
 */
$.prettyPhoto.stopSlideshow = function(){
    $pp_pic_holder.find('.pp_pause').unbind('click').removeClass('pp_pause')
    .addClass('pp_play').click(function(){
        $.prettyPhoto.startSlideshow();
        return false;
    });
    clearInterval(pp_slideshow);
    pp_slideshow=undefined;
}

/**
 * Closes prettyPhoto.
 */
$.prettyPhoto.close = function(){
    if($pp_overlay.is(":animated")) return;

    $.prettyPhoto.stopSlideshow();

    $pp_pic_holder.stop().find('object,embed').css('visibility','hidden');

    $('div.pp_pic_holder,div.ppt,.pp_fade').fadeOut(settings.animation_speed,function(){ $(this).remove(); });

    $pp_overlay.fadeOut(settings.animation_speed, function(){
```

```

        if(settings.hideflash)
$('object,embed,iframe[src*=youtube],iframe[src*=vimeo]').css('visibili
ty','visible'); // Show the flash

        $(this).remove(); // No more need for the prettyPhoto
markup

        $(window).unbind('scroll.prettyphoto');

        clearHashtag();

        settings.callback();

        doresize = true;

        pp_open = false;

        delete settings;
    });
};

/**
content in.
 * Set the proper sizes on the containers and animate the
 */
function _showContent(){
    $('.pp_loaderIcon').hide();

    // Calculate the opened top position of the pic holder
    projectedTop = scroll_pos['scrollTop'] + ((windowHeight/2)
- (pp_dimensions['containerHeight']/2));
    if(projectedTop < 0) projectedTop = 0;

    $ppt.fadeTo(settings.animation_speed,1);

    // Resize the content holder
    $pp_pic_holder.find('.pp_content')
        .animate({
            height:pp_dimensions['contentHeight'],
            width:pp_dimensions['contentWidth']
        },settings.animation_speed);

    // Resize picture the holder
    $pp_pic_holder.animate({
        'top': projectedTop,
        'left': ((windowWidth/2) -
(pp_dimensions['containerWidth']/2) < 0) ? 0 : (windowWidth/2) -
(pp_dimensions['containerWidth']/2),
        width:pp_dimensions['containerWidth']
    },settings.animation_speed,function(){
        $pp_pic_holder.find('.pp_hoverContainer,#fullResImage').height(pp_dimen

```

```

sions['height']).width(pp_dimensions['width']);

$pp_pic_holder.find('.pp_fade').fadeIn(settings.animation_speed); //
Fade the new content

        // Show the nav
        if(isSet &&
_getFileType(pp_images[set_position])=="image") {
$pp_pic_holder.find('.pp_hoverContainer').show(); }else{
$pp_pic_holder.find('.pp_hoverContainer').hide(); }

        if(settings.allow_expand) {
            if(pp_dimensions['resized']){ // Fade the resizing
link if the image is resized
                $('a.pp_expand,a.pp_contract').show();
            }else{
                $('a.pp_expand').hide();
            }
        }

        if(settings.autoplay_slideshow && !pp_slideshow &&
!pp_open) $.prettyPhoto.startSlideshow();

        settings.changepicturecallback(); // Callback!

        pp_open = true;
    });

    _insert_gallery();
    pp_settings.ajaxcallback();
};

/**
 * Hide the content...DUH!
 */
function _hideContent(callback){
    // Fade out the current picture
    $pp_pic_holder.find('#pp_full_res object,#pp_full_res
embed').css('visibility','hidden');
    $pp_pic_holder.find('.pp_fade').fadeOut(settings.animation_speed,function(){
        $('a.pp_loaderIcon').show();

        callback();
    });
};

/**
 * Check the item position in the gallery array, hide or show
the navigation links
 * @param setCount {integer} The total number of items in the

```

```

set
    */
    function _checkPosition(setCount){
        (setCount > 1) ? $(''.pp_nav').show() : $(''.pp_nav').hide();
// Hide the bottom nav if it's not a set.
    };

    /**
    * Resize the item dimensions if it's bigger than the viewport
    * @param width {integer} Width of the item to be opened
    * @param height {integer} Height of the item to be opened
    * @return An array containin the "fitted" dimensions
    */
    function _fitToViewport(width,height){
        resized = false;

        _getDimensions(width,height);

        // Define them in case there's no resize needed
        imageWidth = width, imageHeight = height;

        if( ((pp_containerWidth > windowWidth) ||
(pp_containerHeight > windowHeight)) && doresize &&
settings.allow_resize && !percentBased) {
            resized = true, fitting = false;

            while (!fitting){
                if((pp_containerWidth > windowWidth)){
                    imageWidth = (windowWidth - 200);
                    imageHeight = (height/width) * imageWidth;
                }else if((pp_containerHeight > windowHeight)){
                    imageHeight = (windowHeight - 200);
                    imageWidth = (width/height) * imageHeight;
                }else{
                    fitting = true;
                };

                pp_containerHeight = imageHeight, pp_containerWidth
= imageWidth;
            };

            if((pp_containerWidth > windowWidth) ||
(pp_containerHeight > windowHeight)){
                _fitToViewport(pp_containerWidth,pp_containerHeight)
            };

            _getDimensions(imageWidth,imageHeight);
        };
    }

```

```

        return {
            width:Math.floor(imageWidth),
            height:Math.floor(imageHeight),
            containerHeight:Math.floor(pp_containerHeight),
            containerWidth:Math.floor(pp_containerWidth) +
(settings.horizontal_padding * 2),
            contentHeight:Math.floor(pp_contentHeight),
            contentWidth:Math.floor(pp_contentWidth),
            resized:resized
        };
    };

    /**
     * Get the containers dimensions according to the item size
     * @param width {integer} Width of the item to be opened
     * @param height {integer} Height of the item to be opened
     */
    function _getDimensions(width,height){
        width = parseFloat(width);
        height = parseFloat(height);

        // Get the details height, to do so, I need to clone it
        since it's invisible
        $pp_details = $pp_pic_holder.find('.pp_details');
        $pp_details.width(width);
        detailsHeight = parseFloat($pp_details.css('marginTop')) +
parseFloat($pp_details.css('marginBottom'));

        $pp_details =
$pp_details.clone().addClass(settings.theme).width(width).appendTo($('body')).css({
            'position':'absolute',
            'top':-10000
        });
        detailsHeight += $pp_details.height();
        detailsHeight = (detailsHeight <= 34) ? 36 : detailsHeight;
        // Min-height for the details
        $pp_details.remove();

        // Get the titles height, to do so, I need to clone it
        since it's invisible
        $pp_title = $pp_pic_holder.find('.ppt');
        $pp_title.width(width);
        titleHeight = parseFloat($pp_title.css('marginTop')) +
parseFloat($pp_title.css('marginBottom'));
        $pp_title = $pp_title.clone().appendTo($('body')).css({
            'position':'absolute',
            'top':-10000
        });
        titleHeight += $pp_title.height();
        $pp_title.remove();
    }

```

```

        // Get the container size, to resize the holder to the
        right dimensions
        pp_contentHeight = height + detailsHeight;
        pp_contentWidth = width;
        pp_containerHeight = pp_contentHeight + titleHeight +
$pp_pic_holder.find('.pp_top').height() +
$pp_pic_holder.find('.pp_bottom').height();
        pp_containerWidth = width;
    }

    function _getFileType(itemSrc){
        if (itemSrc.match(/youtube\.com\/watch/i) ||
itemSrc.match(/youtu\.be/i)) {
            return 'youtube';
        }else if (itemSrc.match(/vimeo\.com/i)) {
            return 'vimeo';
        }else if(itemSrc.match(/\b.mov\b/i)){
            return 'quicktime';
        }else if(itemSrc.match(/\b.swf\b/i)){
            return 'flash';
        }else if(itemSrc.match(/\biframe=true\b/i)){
            return 'iframe';
        }else if(itemSrc.match(/\bajax=true\b/i)){
            return 'ajax';
        }else if(itemSrc.match(/\bcustom=true\b/i)){
            return 'custom';
        }else if(itemSrc.substr(0,1) == '#'){
            return 'inline';
        }else{
            return 'image';
        }
    };

    function _center_overlay(){
        if(doresize && typeof $pp_pic_holder != 'undefined') {
            scroll_pos = _get_scroll();
            contentHeight = $pp_pic_holder.height(), contentwidth =
$pp_pic_holder.width();

            projectedTop = (windowHeight/2) +
scroll_pos['scrollTop'] - (contentHeight/2);
            if(projectedTop < 0) projectedTop = 0;

            if(contentHeight > windowHeight)
                return;

            $pp_pic_holder.css({
                'top': projectedTop,
                'left': (windowWidth/2) + scroll_pos['scrollLeft']
- (contentwidth/2)

```

```

        });
    };
};

function _get_scroll(){
    if (self.pageYOffset) {
        return
    {scrollTop:self.pageYOffset,scrollLeft:self.pageXOffset};
    } else if (document.documentElement &&
document.documentElement.scrollTop) { // Explorer 6 Strict
        return
    {scrollTop:document.documentElement.scrollTop,scrollLeft:document.docum
entElement.scrollLeft};
    } else if (document.body) {// all other Explorers
        return
    {scrollTop:document.body.scrollTop,scrollLeft:document.body.scrollLeft}
;
    };
};

function _resize_overlay() {
    windowHeight = $(window).height(), windowHeight =
$(window).width();

    if(typeof $pp_overlay != "undefined")
    $pp_overlay.height($(document).height()).width(windowWidth);
};

function _insert_gallery(){
    if(isSet && settings.overlay_gallery &&
_getFileType(pp_images[set_position])=="image") {
        itemWidth = 52+5; // 52 beign the thumb width, 5 being
the right margin.
        navWidth = (settings.theme == "facebook" ||
settings.theme == "pp_default") ? 50 : 30; // Define the arrow width
depending on the theme

        itemsPerPage =
Math.floor((pp_dimensions['containerWidth'] - 100 - navWidth) /
itemWidth);
        itemsPerPage = (itemsPerPage < pp_images.length) ?
itemsPerPage : pp_images.length;
        totalPage = Math.ceil(pp_images.length / itemsPerPage)
- 1;

        // Hide the nav in the case there's no need for links
        if(totalPage == 0){
            navWidth = 0; // No nav means no width!
            $pp_gallery.find('.pp_arrow_next,.pp_arrow_previous').hide();
        }else{
            $pp_gallery.find('.pp_arrow_next,.pp_arrow_previous').show();

```

```
};

galleryWidth = itemsPerPage * itemWidth;
fullGalleryWidth = pp_images.length * itemWidth;

// Set the proper width to the gallery items
$pp_gallery
    .css('margin-left', -((galleryWidth/2) +
(navWidth/2)))
    .find('div:first').width(galleryWidth+5)
    .find('ul').width(fullGalleryWidth)
    .find('li.selected').removeClass('selected');

goToPage = (Math.floor(set_position/itemsPerPage) <
totalPage) ? Math.floor(set_position/itemsPerPage) : totalPage;

$.prettyPhoto.changeGalleryPage(goToPage);

$pp_gallery_li.filter(':eq('+set_position+')').addClass('selected');
}else{
    $pp_pic_holder.find('.pp_content').unbind('mouseenter
mouseleave');
    // $pp_gallery.hide();
}
}

function _build_overlay(caller){
    // Inject Social Tool markup into General markup
    if(settings.social_tools)
        facebook_like_link =
settings.social_tools.replace('{location_href}',
encodeURIComponent(location.href));

    settings.markup =
settings.markup.replace('{pp_social}','');

    $('body').append(settings.markup); // Inject the markup

    $pp_pic_holder = $('.pp_pic_holder') , $ppt = $('.ppt'),
    $pp_overlay = $('div.pp_overlay'); // Set my global selectors

    // Inject the inline gallery!
    if(isSet && settings.overlay_gallery) {
        currentGalleryPage = 0;
        toInject = "";
        for (var i=0; i < pp_images.length; i++) {
if(!pp_images[i].match(/\b(jpg|jpeg|png|gif)\b/gi)){
            classname = 'default';
            img_src = '';
        }else{
            classname = '';
        }
    }
}
```

```

        img_src = pp_images[i];
    }
    toInject += "<li class='"+classname+"'><a
href='#'><img src='" + img_src + "' width='50' alt='' /></a></li>";
    };

    toInject =
settings.gallery_markup.replace(/{\gallery}/g,toInject);

    $pp_pic_holder.find('#pp_full_res').after(toInject);

    $pp_gallery = $('.pp_pic_holder .pp_gallery'),
    $pp_gallery_li = $pp_gallery.find('li'); // Set the gallery selectors

    $pp_gallery.find('.pp_arrow_next').click(function(){
        $.prettyPhoto.changeGalleryPage('next');
        $.prettyPhoto.stopSlideshow();
        return false;
    });

    $pp_gallery.find('.pp_arrow_previous').click(function(){
        $.prettyPhoto.changeGalleryPage('previous');
        $.prettyPhoto.stopSlideshow();
        return false;
    });

    $pp_pic_holder.find('.pp_content').hover(
        function(){
    $pp_pic_holder.find('.pp_gallery:not(.disabled)').fadeIn();
        },
        function(){
    $pp_pic_holder.find('.pp_gallery:not(.disabled)').fadeOut();
        });

    itemWidth = 52+5; // 52 beign the thumb width, 5 being
the right margin.
    $pp_gallery_li.each(function(i){
        $(this)
            .find('a')
            .click(function(){
                $.prettyPhoto.changePage(i);
                $.prettyPhoto.stopSlideshow();
                return false;
            });
    });
};

// Inject the play/pause if it's a slideshow
if(settings.slideshow){
    $pp_pic_holder.find('.pp_nav').prepend('<a href="#"

```

```
class="pp_play">Play</a>')
    $pp_pic_holder.find('.pp_nav
.pp_play').click(function(){
        $.prettyPhoto.startSlideshow();
        return false;
    });
}

$pp_pic_holder.attr('class', 'pp_pic_holder ' +
settings.theme); // Set the proper theme

$pp_overlay
    .css({
        'opacity':0,
        'height':$(document).height(),
        'width':$(window).width()
    })
    .bind('click',function(){
        if(!settings.modal) $.prettyPhoto.close();
    });

$('a.pp_close').bind('click',function(){
$.prettyPhoto.close(); return false; });

if(settings.allow_expand) {
    $('a.pp_expand').bind('click',function(e){
        // Expand the image
        if($(this).hasClass('pp_expand')){
$(this).removeClass('pp_expand').addClass('pp_contract');
            doresize = false;
        }else{
$(this).removeClass('pp_contract').addClass('pp_expand');
            doresize = true;
        };
        _hideContent(function(){ $.prettyPhoto.open(); });

        return false;
    });
}

$pp_pic_holder.find('.pp_previous, .pp_nav
.pp_arrow_previous').bind('click',function(){
    $.prettyPhoto.changePage('previous');
    $.prettyPhoto.stopSlideshow();
    return false;
});

$pp_pic_holder.find('.pp_next, .pp_nav
.pp_arrow_next').bind('click',function(){
```

```

        $.prettyPhoto.changePage('next');
        $.prettyPhoto.stopSlideshow();
        return false;
    });

    _center_overlay(); // Center it
};

if(!pp_alreadyInitialized && getHashtag()){
    pp_alreadyInitialized = true;

    // Grab the rel index to trigger the click on the correct
element
    hashIndex = getHashtag();
    hashRel = hashIndex;
    hashIndex =
hashIndex.substring(hashIndex.indexOf('/')+1,hashIndex.length-1);
    hashRel = hashRel.substring(0,hashRel.indexOf('/'));

    // Little timeout to make sure all the prettyPhoto
initialize scripts has been run.
    // Useful in the event the page contain several init
scripts.
    setTimeout(function(){
    $("a["+pp_settings.hook+"^='"+hashRel+"']:eq("+hashIndex+"").trigger('
click'); },50);
    }

    return
this.unbind('click.prettyphoto').bind('click.prettyphoto',$.prettyPhoto
.initialize); // Return the jQuery object for chaining. The unbind
method is used to avoid click conflict when the plugin is called more
than once
};

function getHashtag(){
    var url = location.href;
    hashtag = (url.indexOf('#prettyPhoto') !== -1) ?
decodeURI(url.substring(url.indexOf('#prettyPhoto')+1,url.length)) :
false;
    if(hashtag){ hashtag = hashtag.replace(/<|>/g,''); }
    return hashtag;
};

function setHashtag(){
    if(typeof theRel == 'undefined') return; // theRel is set on
normal calls, it's impossible to deeplink using the API
    location.hash = theRel + '/' + rel_index + '/';
};

function clearHashtag(){

```

```
        if ( location.href.indexOf('#prettyPhoto') !== -1 )
location.hash = '';
    }

    function getParam(name,url){
        name = name.replace(/\[/, "\\[").replace(/\]/, "\\]");
        var regexS = "[\\?&]+" + name + "=( [^&#]*) ";
        var regex = new RegExp( regexS );
        var results = regex.exec( url );
        return ( results == null ) ? "" : results[1];
    }

})(jQuery);

var pp_alreadyInitialized = false; // Used for the deep linking to make
sure not to call the same function several times.
```

From:

<https://wiki.hennweb.de/> - HennWeb

Permanent link:

https://wiki.hennweb.de/doku.php?id=wiki_anleitung:prettyphoto

Last update: **12/11/2020 15:15**

