

## PSBigPicture

### A self-contained JQuery function to dynamically display a large image.

I have a customer web application where I needed to display a large number of image thumbnails that can be clicked to show a larger view of the same image. This is generally, pretty easy to do, but I was concerned about the performance. Normally, all the thumbnails and their corresponding larger images would be downloaded to render the page when the page is loaded, which could take quite a bit of time due to the size of the large images.

I decided I needed to allow all the thumbnails to download and render normally, but only download and show the large image when the user actually wants to see it. To accomplish this, I developed a simple JQuery function to display the larger view dynamically in a Lightbox-like popup.

In addition to delaying the large image downloads, I also wanted to make the function completely self contained so no additional html or css is needed in the page. To use the function, all that is needed is to add the link to the script file (along with the link to the JQuery library), and add the function call in whatever event is to trigger it (most likely an onclick).

Here is a bit of markup that displays the thumbnail image which has an onclick event to show the large view:

```
...  
  
...
```

The PSBigPicture function takes three parameters. The first is the url of the large image. The second parameter is the target width of the popup display. It is the “target” because some resizing may take place to keep the images aspect ratio and keep the popup display within 80% of the current browser window size. I picked 80% so that there would still be some of the page showing around the popup. The last two parameters are the image’s title and description. Both are optional.

Let’s take a look at the code.

The first part is where the necessary html gets added to the page’s body. Notice that it is only done once (if (\$("#PSBigPicture").length == 0)). The html contains a series of divs that will be used to construct the popup. The first one (id="PSBigPicture") is the background that will overlay the entire window with a semi-transparent fill (if the browser supports opacity). This will keep the user from clicking other elements on the page while the popup is visible.

Within the PSBigPicture div is the loading div (id="PSBigPictureLoading") which was added to give the user some feedback while the large image is being downloaded. Both the background and loading divs will be shown immediately.

The next div (id="PSBigPictureShadow") will be used to produce a shadow effect behind the popup on browsers that don't support the box-shadow css tag. This div will be sized to match the div containing the image and will be offset slightly to the right and below it.

The last div is the one that will contain the image, its title and description and a close button. In order to force the divs to overlay the page and each other correctly the css z-index is used.

The following is what the actual resulting html would look like formatted in a page markup.

```
<div id="PSBigPicture" style="position:fixed;display:none; top:0px; width:100%; height:100%;  
    background-color:#FFFFFF; z-index:1">  
    <div id="PSBigPictureShadow" style="position:absolute; background-color: #000000" ></div>  
</div>  
<div id="PSBigPictureLoading" style="position:absolute;padding:12px; border:2px solid #000000;  
    background:#FFFFFF; color:#0000FF;z-index:2">  
    Loading...  
</div>  
<div id="PSBigPictureImgBox" style="position:absolute; display:none; background-color: #FFFFFF;  
    color:black; text-align:left;z-index:3">  
    <img id="PSBigPictureImg" src="" alt="" title="" />  
    <br />  
    <div style="float:left; text-align:left; width:70%">  
        <span id="PSBigPictureTitle"></span>  
    </div>  
    <div id="PSBigPictureClose" style="float:right;padding-right:8px;  
        cursor:pointer;"title="Click to close">  
        <b><u>Close</u></b>  
    </div>  
</div>
```

Once the html with in-line css is added to the page, some event handlers are created. The first is an onclick handler to cause the popup to close if the background is clicked. The second is the onclick handler to close the popup when "Close" is clicked. The last is the most important. It is the event that is triggered AFTER the large image is completely loaded. I needed to wait for the image to be completely loaded to ensure its size is valid when I resize everything. It also prevents the user from seeing the image slowly painting down the screen (that's why there is the PSBigPictureLoading div).

Before we look at the details of the load event, notice that the rest of the function call displays the background and loading divs then assigns the image's source to the img. The initial function will be done at that point. The rest will be done after the image is completely loaded in the load event handler.

Lastly, let's look at the image's load event handler. It starts with a bunch of vars containing default values for borders with, shadow size and corner radius. I put those there so they can easily be changed. Next I assemble a little html to show the optional title and description and add it to the html created earlier. Finally, the image and its container get resized and the shadows are applied. Note on the size

that if the width parameter contains 0, the target width is automatically set to 80% of the current window's width so the popup will get the maximum size (within the 80% while maintaining aspect ratio).

Enjoy.

This is the annotated code. Some lines have been wrapped to fit the page better.

```
function PSBigPicture(source, width, title, desc) {
    // Create PSBigPicture HTML
    if ($("#PSBigPicture").length == 0) { // Create this only once
        $("body").append("<div id=\"PSBigPicture\"
            style=\"position:fixed;display:none; top:0px; width:100%; height:100%;
            background-color:#FFFFFF; z-index:1\">" +

            "<div id=\"PSBigPictureShadow\" style=\"position:absolute;
            background-color: #000000\" ></div>" +
            "</div>" +

            "<div id =\"PSBigPictureLoading\" style=\"position:absolute;
            padding:12px; border:2px solid #000000; background:#FFFFFF;
            color:#0000FF;z-index:2 \">Loading...</div>" +

            "<div id=\"PSBigPictureImgBox\" style=\"position:absolute;
            display:none; background-color: #FFFFFF; color:black;
            text-align:left;z-index:3\">" +

            "<img id=\"PSBigPictureImg\" src=\"\" alt=\"\" title=\"\" />" +

            "<br />" +

            "<div style=\"float:left; text-align:left; width:70%\">
                <span id=\"PSBigPictureTitle\"></span></div>" +
            "<div id=\"PSBigPictureClose\" style=\"float:right;
            padding-right:8px;cursor:pointer;\"
            title=\"Click to close\"><b><u>Close</u></b></div>" +
            "</div>");

        // Initialize events for closing
        // Click anywhere in the PSBigPicture (background area) div to close
        $("#PSBigPicture").click(function () {
            $("#PSBigPictureImgBox").fadeOut("slow");
            $("#PSBigPicture").css("display", "none");
        });

        // Or, click fake "Close" button to close
        $("#PSBigPictureClose").click(function () {
            $("#PSBigPictureImgBox").fadeOut("slow");
            $("#PSBigPicture").css("display", "none");
        });
    } // End of first-time stuff

    // onLoad handler executed when the image is completely loaded
    $("#PSBigPictureImg").load(function () {
        var bdr = 8; // default border size
        var ds = 8; // default drop shadow size;
        var dss = 8; // default drop shadow spread;
        var cnr = 5; // default corner radius;
        var str = "";
        // Build title and description html
        if (title != "") {
```

```

        str += "<b>" + title + "</b>";
    }
    if (desc != null && desc != "") {
        str = str + "<br />" + desc;
    }
    $("#PSBigPictureTitle").html(str);
    // Size the image
    // If width param is 0 use 80% of window width
    var w = (width == 0) ? ($(window).width() * .8) : width;
    if (w > ($(window).width() * .8)) { // adjust width to fit 80% of window.
        w = ($(window).width() * .8);
    }
    $("#PSBigPictureImg").attr({ "width": w });
    // Now make sure it is inside vertical bounds.
    var h = $("#PSBigPictureImgBox").height();
    if (h > ($(window).height() * .8)) {
        // recalculate width to keep height within 80% of window height
        var dif = h - ($(window).height() * .8);
        var rat = dif / h;
        w = w - (rat * w);
        $("#PSBigPictureImg").attr({ "width": w });
    }
    // Size the image background div based on resized image size
    $("#PSBigPictureImgBox").css({ "width": w + bdr, "padding-left": bdr,
        "padding-top": bdr, "padding-bottom": bdr * 3 });
    // Set position of popup centered in window.
    var leftOffset = ($(window).width() / 2) - ($("#PSBigPictureImgBox").width() / 2);
    var topOffset = ($(window).height() / 2) - ($("#PSBigPictureImgBox").height() / 2);
    topOffset += $(window).scrollTop(); // adjust for any vertical scrolling.
    // position the shadow div offset behind the image div.
    // This may be hidden by CSS 3 shadow on newer browsers, but will show on IE 8.
    $("#PSBigPictureImgBox").css({ "left": leftOffset, "top": topOffset });
    if ($.support.opacity) { // Is browser capable of doing opacity,
        //then assume the rest of the new stuff will work too.
        var shad = ds + "px " + ds + "px " + dss + "px #3D3D3D";
        $("#PSBigPictureImgBox").css({ "box-shadow": shad });
        $("#PSBigPictureImgBox").css({ "border-radius": cnr });
    }
    else { // for older browsers
        $("#PSBigPicture").css({ "background-color": "#3D3D3D" });
        $("#PSBigPictureShadow").css({ "left": leftOffset + 8, "top": topOffset +
            8, "width": width + bdr + 8, "height": ($("#PSBigPictureImgBox").height() +
            (bdr * 3) + 8 });
    }
    $("#PSBigPictureLoading").fadeOut("fast");
    $("#PSBigPictureImgBox").fadeIn("slow");
}); // End of load event handler

// Show the background and loading div immediately
var loff = ($(window).width() / 2) - ($("#PSBigPictureLoading").width() / 2);
var toff = ($(window).height() / 2) - ($("#PSBigPictureLoading").height() / 2);
$("#PSBigPictureLoading").css({ "left": loff, "top": toff });
$("#PSBigPicture").fadeIn("fast");
$("#PSBigPicture").fadeTo("fast", 0.4);
$("#PSBigPictureLoading").fadeIn("fast");

// Load the image--actually the load process only starts here.
// The rest will be done in the load event when the image is completely loaded.
$("#PSBigPictureImg").attr({ "src": source, "title": title });

}; // End of function

```

