



Using JQuery to Make a Photo Slideshow

This exercise was modified from the slideshow code at <http://slidesjs.com/>.

You will need the following folders and files. You will need to ftp all the folders to your site under the directory for the project. This must be done properly for your slideshow to work.

- index.html - this can be renamed, or the code within can be copied and placed within another page on your site. Make sure you copy all associated divs and head content to make the slideshow work. See below.
- img folder - this has images associated with the slideshow, arrows, backgrounds, etc. These items can be modified or replaced. If you name your images with this convention, slide1.jpg, they will automatically be inserted in slideshow. Otherwise, you will have to modify code with your own filenames. Make sure all photos are optimized first in Photoshop and that they are size properly for your interface. The current photo size is 576px x 432px (8 inches x 6 inches). If your images are different sizes, you will have to modify the interface.
- css folder - this contains the global.css file. You can modify this file to change the look of your slideshow. You will also have the main stylesheet that you use for the project.
- js folder - this contains slides.min.jquery.js, the jquery file referenced in the html page. You will not modify this, but just make sure you have it uploaded in the js folder on your site.

You can see a link to the example files on the course outline, but you will receive the files on the Desktop in a folder called photo_slideshow.

ACL Fest 2011



The html page must have the reference to the stylesheet in the <head> section. This is a separate stylesheet than the main one you are using for your site. You can use the stylesheet to change the way the slideshow looks.

```
<link rel="stylesheet" href="css/global.css">
```

The html page must also include javascript references to external and internal JQuery libraries as well as the slides function. These must also be included in the head. The function will allow you to change things about the way the slideshow operates.

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js"></script>
<script src="js/slides.min.jquery.js"></script>
```

```
<script>
    $(function(){
        $('#slides').slides({
            preload: true,
            preloadImage: 'img/loading.gif',
            play: 10000,
            pause: 2500,
            effect: 'fade',
            crossfade: true,
            fadeSpeed: 1000,
            hoverPause: true,
            animationStart: function(current){
                $('.caption').animate({
                    bottom:-35
                },100);
                if (window.console && console.log) {
                    // example return of current slide number
                    console.log('animationStart on slide: ', current);
                }
            },
            animationComplete: function(current){
                $('.caption').animate({
                    bottom:0
                },200);
                if (window.console && console.log) {
                    // example return of current slide number
                    console.log('animationComplete on slide: ', current);
                }
            },
            slidesLoaded: function() {
                $('.caption').animate({
                    bottom:0
                },200);
            }
        });
    });
</script>
```

The JQuery libraries provide the functionality so that you can write simpler code to create your effects. These often get updated, so try out the newest JQuery library and test out the code. Accessing it from the Google API page makes it easier to maintain.

<http://code.google.com/apis/libraries/devguide.html>

There are several nested divs for this slideshow in the css. All divs must be copied and pasted into a div on the page where you want the slideshow to appear.

CSS - ignore all the reset code at the top of the stylesheet. Just leave that there. It resets the code to make sure there are no default variables set by the browser.

Divs

- `slideshow_title` - this is a div that simply holds a title for your slideshow. You can use it or remove it. You can modify how the `<h2>` of the slideshow title looks in your main css file.
- `container` - id for slideshow; find div in css; you can adjust width, padding, margins, etc.
- `slides` - id for slides - allows you to position photos within container; adjust top, left
- `slides_container` - class for slides container. Allows you to set width. Don't change anything else.
- `slide` - class for slides. Each photo is in a slide class.
- `#frame` - this controls the size and location of the frame behind the images (looks like a polaroid, white border with a shadow). You can remove this frame or you can modify the image to create a custom look. You may have to do some advanced customization if your photos are not the same size as the ones in the example.
- `#slides .next`, `#slides .prev` - this handles the location of the arrows.
- `.pagination` - this handles the dots below the slideshow so you can advance that way. You may use this to adjust the location of the pagination.
- `.caption` - This controls the way that the captions look. You can change the for size, color, etc.

Images

The images in this slideshow are 576px x 432px (8 inches x 6 inches). If your images are different, then you will need to make adjustments to your `#slideshow-title`, `#frame`, `#container`, `.slides_container`, `div.slide`

Slides

Each slide has an image reference (i.e. ``). Adjust the filename (if necessary), the height, width (if necessary) and the alt tag. You can also provide (and should change) a link location for the file. `` This will surround the img tag and allow the user to click on the image to open it in a new window. Leave `target="_blank"`. Close `` after the img tag. You can modify what is in the slideshow, but if you need more, just copy and paste one of the entire `<div class="slide">` section and make changes.

Captions

Under `<div class="caption">`, change the paragraph for that image to your caption. Make sure you also provide an alt tag for each image (for accessibility).

In the html, there are a few things you can change in the function:

- `preload: true`; leave that, preloads images so you don't have to wait during slideshow.
- `preloadImage: 'img/loading.gif'`; leave that, too. That's the little animation that shows up while loading.
- `play: 10000`, speed of the transition of each slide when it is on auto play (in milliseconds),

- 10,000 = 10 seconds); if you don't want it to auto play, set this value to 0.
- pause: 2500, pause when you use a button or pagination, adds extra time to the transition.
- effect: 'fade', you can use one of two effects, fade or slide. Slide slides the image across. Try them both.
- crossfade: true, Crossfade goes with fade. It lets one image fade out while the other fades in.
- fadeSpeed: 1000, (speed of the fade in milliseconds, 1000 = 1 second).
- hoverPause: true, means it will pause when someone is hovering over the slideshow.

Leave the animation functions. They make it work!

Exercise:

You will be using this Slide plugin to make a photo slideshow with captions.

1. Use Photoshop to optimize 10 images. If you make them 576px x 432px (8in x 6in - landscape view), they will fit in the slideshow. If not, you will need to make appropriate changes to the slideshow to accommodate the new size.
2. Using Save for Web, name each image slide1.jpg, slide2.jpg, etc, naming all 10 photos.
3. Download all the files associated with the slideshow. You will need the index.html, css folder, js folder and img folder. Save these in their own project folder for this exercise.
3. Copy those images to the img folder. They will replace the ones that are already in there, if you used the naming convention. If you changed the names, you will have to adjust the image references in the html code.
4. Try the slideshow by opening the html page in the browser. How does it look, do you need to change anything? Does it work? You may need to make changes to the divs in the stylesheet.
5. Add captions and alt tags for all the images in the html page. Save and test it again in the browser. Make sure you tell a story with your captions, not just identify what is in the photo. You can control how the captions look with the #caption div in the stylesheet. Change the background color (adjust rgba to see how the "alpha" works), font size, font family and color to see what happens.
6. You can use the frame image provided or create your own, or even remove it. You can also change the image used for the arrows. Leave the pagination image as is, unless you feel like changing the colors. Make sure when you resave it that you keep the same name.
7. Use the stylesheet to move the arrows to the bottom of the slideshow, instead of the side. Find the #slides .next and #slides .prev styles and adjust the top and left locations of each.
8. Play with a few of the variables in the function in the html page. Try the 'slide' effect instead of the 'fade' effect. Return it to the 'fade' effect before completing. Adjust the play and fadeSpeed times. Decide if you want the slideshow to also auto play or not (if not, make play: 0). For each of those items, make sure there is a comma at the end of the line. Do not touch anything below the animationStart function.
9. Don't forget to give your slideshow a title in the #slideshow_title div. You can adjust the <h2> styles in the stylesheet. Feel free to add additional <body> styles to control things like background color or margins.
10. Provide an overview description of the stylesheet in the #main div. You can adjust the <p> styles in the stylesheet.
11. Complete your slideshow. You will need to make a new folder on fetch then upload all the files and folders to that new folder. Make a link from your assignment page to the slideshow (i.e. slideshow/index.html).