Introduction

I've been curious about how slideshows work. How much is involved in keeping them going smoothly? I decided to do it the hard way, without jQuery, so that I'd understand the inner workings better. It took a lot longer, and it was, at times, frustrating, but the process was much more rewarding.

The slideshow was built with Wordpress supplying the CMS. The idea in this version is that you make the slides as posts and add a category to them which treats them as a class of slides. The PHP then uses a modified Wordpress loop to generate the slides, and from there they are handled with Javascript.

Goals:

The slideshow should be scalable, with as few as one slide. It makes this possible by cloning the slide.
The slideshow should be easy to use. The user simply has to create the Front Page Slide category and append it to the desired post.
The slideshow uses PHP and Wordpress.

Though I feel I accomplished these goals, this version is only the beginning. For starters, it's not structured as a plug-in, which is vital to sharing it with others. There is no front end control panel, either. The next iteration should take these goals into consideration.

Lessons Learned:

This project was intimidating for me, because I had never worked so extensively with CSS animation before. This was a trial by fire, forcing me to learn timing, edge cases and how all of this had to tie in with user interaction. The user should see something very simple even though a lot is going on in the background.

Timing is absolutely vital to understanding animation. In this case, I set up timeouts that would apply classes at the appropriate time, sliding an object from one part of the screen to the next. From there, I made sure to update the slide's new position with freshly-applied styles. Did I mention that each step takes into account the current dimensions of the window's width?

It was also an opportunity to practice PHP. Though I relied on the Wordpress loop for most of the application, I did venture into more complicated PHP usage: I used loops for some processes, and found the structure is almost identical to the one used in Javascript. The transition there was easy. There was no need for object-oriented programming in PHP, since all the objects were manipulated in Javascript.

But something that has been hammered into my head with this project more than anything is DRY. Don't copy the entirety of one function into another and expect to simply refactor it for the new circumstances. The first iteration of the slideshow used one long function to assemble the slideshow elements, and I quickly realized this was a bad idea.

It took some refactoring, but I was able to transition the application into a function-focused code base. This helped make it responsive, because now every step of the application was a separate function, and calling it at the appropriate time ensured the dimensions would be the most current.