CSCE 120: Learning To Code

Organizing Code II
Hacktivity 13.1

Introduction

Prior to engaging in this hacktivity, you should have completed all of the pre-class activities as outlined in this module. At the start of class, you will be randomly assigned a partner to work with on the entirety of this hacktivity as a *peer programming* activity. Your instructor will inform you of your partner for this class.

One of you will be the driver and the other will take on the navigator role. Recall that a driver is in charge of the keyboard and computer while the navigator is in charge of the handout and directing the activity. However, you are *both* responsible for contributing and discussing solutions. If you were a driver/navigator in the prior activity, switch roles for this activity.

1 Knowledge Check

1. What is a CDN?
2. Describe what hotlinking is; discuss the advantages and disadvantages of hotlinking versus keeping a local copy of an asset in your project.
3. What is a minifier?
4. What is a transpiler?
5. What is a polyfill?

2 Project Organization

We have provided a nearly complete version of the Albums App from a previous module. You will reorganize this project in several ways. To start, download the code we've provided from GitHub using the URL, https://github.com/cbourke/AlbumsApp and open the project in Light Table and evaluate the HTML file to reference how the page should look (its not entirely functional yet).

2.1 Organizing Assets

All of the "assets" for the application are in the same root folder. Typically, different types of assets are collected in different folders for organization.

1. Create an assets folder in your project as well as several subfolders to hold your images, CSS file(s), and JavaScript file(s). Move each type of file into its respective folder.
2. Since you moved some of the files, you will now need to make appropriate changes to the `index.html` file to reference the new locations.

2.2 Localizing Assets

The application uses the jQuery, jQuery UI, and Bootstrap libraries, but does so by loading them from a Content Delivery Network (CDN). We will now "localize" these libraries by downloading them and using local copies.

1. Create a `lib` folder in your assets folder.
2. Download the three libraries:
   - jQuery (http://jquery.com/,
   - jQuery UI (http://jqueryui.com/)
   - Bootstrap (http://getbootstrap.com/)
3. Unzip these libraries into your `lib` folder into subfolders for each library. Name the subfolders with a version attached to them, for example `jquery_2.1.4`
4. Again, you will need to make appropriate changes to the HTML file to use the local copies rather than the CDNs

2.3 Backwards Compatability

Assuming you have everything correct, try the application out. Try adding an album. You'll find that it does not work. This is because the app uses some features of EC-

MAScript 6 (ES6) and Light Table only supports ES5. To contrast, try opening the `index.html` file in your favorite browser and try it out.

Though many browsers support ES6, we want to ensure that our application will work in all browsers. However, we shouldn't let this prevent us from programming in the latest versions and using the latest features of JavaScript. One solution would be to use a transpiler such as Babel (https://babeljs.io/) to translate our ES6 code into ES5-compliant code. However, this is a bit beyond this module.

Instead, we'll use a *polyfill* to enable the functionality that we wish to use.

1. Create a new JavaScript source file in your asset's JavaScript folder.
2. Cut and paste the polyfill for the `findIndex()` function which can be found here: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/findIndex
3. Include this file in your `index.html` file and try running the application again.

2.4 JavaScript Minification

Let's now make our application's JavaScript file more compact by *minifying* it.

1. Open your browser to Google's Closure Compiler: http://closure-compiler.appspot.com/
2. Cut and paste the contents of the `album.js` source file into the compiler window (be sure to remove the content there first).
3. Click "Compile" and observe the results (be sure to use the "Simple" optimization option). How much smaller is the minified version?
4. Cut and paste the minified source into a *new* JavaScript source file named `album.min.js`
5. Change the `index.html` to use this minified version instead of the original.

2.5 CSS Minification

Similarly, minify the CSS file using the following online CSS minifier: http://cssminifier.com/

That was a lot of work. Fortunately, there are many other technologies and frameworks that make it easier. For example, build tools such as Grunt http://gruntjs.com/ can be used to automate these processes.

3 Sharing Your Project

In the next hacktivity, we'll be using Git to share and collaborate on code. With the time remaining and between now and the next hacktivity be sure to have registered with GitHub and installed GitHub's client.

1. Go to https://github.com/ and register for an account (unless you already have one)
2. Download and install GitHub's Desktop Client for your system: https://desktop.github.com