Code.org App Lab and “Real World” Coding

**Purpose and Scope**

The purpose of this document is to start to explain where an App Lab program fits in the real world of app development and JavaScript programming.

Some will be interested in this because they want to publish their apps. Others will be interested in this because they want to learn “real” JavaScript and want to know how the skills they are developing in App Lab will apply in their future careers.

**Key Points**

1. App Lab creates hybrid apps – not native apps. These apps live inside an html page (more about this later). Connection to a web server is needed to allow the app to run. If you want native apps, there are many other ways to create them. Keep in mind, however, that many apps (such as eCommerce, credit card, and airline/flight apps) have very limited usefulness if not connected to the web anyway. Also keep in mind that internet connectivity “everywhere” is becoming the norm for many users.
2. While App Lab is a JavaScript IDE, Code.org has created extensive additional capabilities to allow you to quickly create nice apps. When you toggle between the block mode and the text mode, the text that appears is not native JavaScript.
3. App Lab depends on several custom api’s, frameworks, libraries – whatever you want to call them. While this means you aren’t learning “native” JavaScript, very few developers create everything totally from scratch. App Lab uses, for example, jQuery. This is very commonly used by almost all JavaScript developers – so get used to it and be happy taking advantage of the fact that others have done a lot of hard work for you.
4. You can easily share your App Lab apps with others without knowing much of what’s covered in this paper.
	1. Code.org has provided a nice “Share” button at the top of the IDE window. You can merely share the link with others via email, text, etc. and they will be able to access your app from their phones – and it will work with both Android and IOS (because it is not a native app as described above).
	2. Your user can place an icon on their Android screen simply by using the dropdown menu in the upper right corner when the app is open in the browser and select “Add to Home Screen”.
5. Most of what is covered in this paper comes from examining the contents of the zipped package created automatically via the Share>Show Advanced Options>Export feature of App Lab:



**How It Works**

The unzipped files from the Export above are as follows:





The main web page is index.html and the code is as follows:



Note that word wrap is turned off and that the div id=”divApplab” actually contains very many elements not show as they run off the screen to the right.

The index.html page above links to the fonts (provided in a local file) and two css files contained in the unzipped main folder. The applab.css file contains multiple references to Droplet <http://droplet-editor.github.io/> -- which provides the toggling between block and text mode for us in App Lab. The style.css file is a normal css file.

The index.html page also contains four script tags. The first is a cloud-based copy of jQuery. The second is a function – and I don’t know exactly what it is doing. The third is the url for the specific App Lab code a student woulf write (although I don’t understand why it isn’t referencing the local copy of the file. The fourth references a special applab.csi file that contains the App Lab “special sauce” that shields us from a lot of “real world” JavaScript – whether we want to be or not.

All of these files work together to provide a dynamic hybrid app hosted within a single html document.

**Screenshots of Portions of Key Files**

1. index.html: included above
2. styles.css:



1. applab.css



1. code.js



1. applab-api.js



Note that almost all of the above continue on for many lines of code and aren’t word-wrapped, either.

Obviously, this code is well beyond the scope of abilities for any beginning programmer.