Dart Productivity App

Problem Statement
In Middle School and High Schools students receive Agendas and planners to use to keep track of school work. Students have a place to record their assignments, when things are due, and prioritize tasks.

The goal of this project will be to create a simple productivity app to record student data and use tools to help them at school. The main feature of the app will be its To Do List.
The language that will be used to solve this problem is Dart programming language, developed by Google.

A brief survey of alternatives
The main alternative to Dart is JavaScript. Google decided that JavaScript should be rewritten with cleaner code and optimized performance. The Dart language does not stray too far from the coding elements of JavaScript but it does have some unique features which makes the language productive. JavaScript has spin offs that makes the code simplified like jQuery, Note.js, etc. which extend its features.

Dart offers code that combines jQuery and JavaScript for selecting html elements and modifying them.

Build Tools
There are various tools available for Dart. Firstly, the Dart SDK needs to be downloaded for development purposes. The Dart lang site also allows developers to use their DartPad tool for simple app testing inside browsers without the need for the SDK. But more complex applications require the SDK. Next an IDE or text editors is needed. Dart recommends the WebStorm IDE and the environment has to be set up. I briefly experimented with this IDE and the also another one called Dart Editor, but I preferred using Atom.

I use Atom for my web development projects because it has a minimalist design, and has lots of plugins. To support the Dart language on Atom I had to add Dart plugins to recognize the SDK, and highlight text support.

Another tool that was used for testing was Dartium, a version of chrome that supports the Dart VM. Dart also offers other tools such as pub, for adding libraries to packages; dart2Js for for converting Dart code to JavaScript so it renders on IE, FireFox, etc.,

Language Features used in Solution
- querySelector()
- event listeners

Relating to Course
Dart has a lot of similar features in common with functional programming languages.
- Functions:
    - Functions in Dart are objects anehave their own type, called Function. Dart uses first-class functions which is also used by Scala and Clojure. First-class functions allows functions to be passed as parameters to other functions. Functions can also be assigned to values. Essentially functions symbols follow the same life-cycle as values.

        - offers anonymous functions, functions short cuts if only one expression, and optional parameters for functions, something similar to what is offered in clojure and scala

- Lexical Scoping:
    - Dart uses lexical scoping, nested functions inherit from the variables of upper functions. As we learned in class lexical scoping is when the function body inherits from the context of the expression that defines the function
    - Dart also supports closures. Functions having access to variables outside of the its original scope. Closures were used in Scala to reduce multiple parameter functions to single parameter functions.

Similar to Scala string interpolation can be used to output the value of a variable inside a string using the '$' symbol.
Dart does not support the in secure eval() function used in JavaScript to evaluate a an expression or execute a statement.

Type inference is not part of the Dart language, unlike Scala. Dart has a dynamic type system, which executes its code at runtime.

Dart supports mixins like Scala for polymorphism using traits, which can not be done in Java.