DEPLOYING YOUR APP

DEPLOYING TO GITHUB PAGES:

To set up your GitHub Pages repo (one time only):

1. *On github.com, create a repo called* `username.github.io`, *where* `username` *is your GitHub username.*
2. *Click the* Clone or Download *button for the new repo and copy the address*
3. *In your terminal, navigate to the parent folder where you want to add the new repo, then type* `git clone`, *paste the copied address, and press* Enter
4. *Create or move the files for your app into the newly created* `username.github.io` *folder*

To update your GitHub Pages site (every time you're ready to push new changes):

In your Terminal app, `cd` *into the* `username.github.io` *folder, then use the standard commands for updating a git repo:*

```
git add .
git commit -m "description of change"
git push origin master
```

(remember that these are 3 separate commands, which you should execute one at a time)

DEPLOYING TO FIREBASE HOSTING:

To set up Firebase Hosting (one time only):

1. *On firebase.google.com, in the console for your app, click* Hosting, *then click* Get Started.
2. *Execute the command shown using the terminal on your computer, then return to the browser and click Continue.*
3. *In your Terminal app, execute the* `firebase login` *command.*
4. *In your Terminal app, navigate to the folder containing your app files, then execute the* `firebase init` *command.*
5. *Move the files for your app into the newly created* `public` *subfolder.*
6. *In your Terminal app, execute the* `firebase deploy` *command.*

To update the front end (HTML/CSS/JS) of your Firebase site (every time you're ready to push new changes):

In your Terminal app, `cd` *into the main folder of your Firebase project, then execute the* `firebase deploy` *command.*

TRANSPILING WITH BABEL:

To set up Babel for a project (one time only per project):

1. At the command line, navigate to the main directory for your app.
2. Enter the following command:
   ```
   touch package.json
   ```
3. Open the package.json file you just created in your editor, type
   ```
   {}
   ```
   then save your changes and close the file.
4. Enter the following command:
   ```
   touch .babelrc
   ```
5. Open the .babelrc file you just created in your editor, type
   ```
   {
     "presets": ["env"]
   }
   ```
   then save your changes and close the file.
6. Enter the following command:
   ```
   npm install babel-cli --save-dev
   ```
   Note: As long as you receive no error messages (marked ERR), you can ignore any warnings (marked WARN).
7. Enter the following command:
   ```
   npm install babel-preset-env --save-dev
   ```
8. Open the package.json file in your editor, add the following key-value pair
   ```
   "scripts": {
     "build": "babel js -d lib"
   }
   ```
   then save your changes and close the file.
   Note: Be sure to include a comma between key-value pairs, but not after the final key-value pair — remember the JSON rules!

To transpile the contents of the js folder:

Enter the following command:

```
npm run build
```

In your editor, open the .js file from the lib folder, which should be the transpiled version, and verify that the code has been transpiled.