Learning & Development

## Goal

The agreement we've come to is that it doesn't necessarily have to directly relate to WaterWorth - or if it does, there isn't pressure for it to end up with a "successful" deliverable which actually gets added to the codebase. It's great if it can be directly integrated into WaterWorth, but that's not the primary purpose for this time.

* For new concepts, start with bite-sized lessons/tutorials to see how interesting/beneficial it can really be.
* If the bite-sized chunk is proving worthwhile, have the option to move on to a larger project surrounding that concept (ex/ Udemy or Coursera course, etc).
* Continually share in this channel what we're working on.
* If it's something beneficial and someone else is interested in it, spend some time teaching each other.
* Potential for joint / parallel projects if both of us are interested in the same concept.

## New Learning Ideas

* Windows Subsystem for Linux (WLS) ✓
* React.js
  + JSX
  + Lit-html / (git:developit/htm) ✓
  + Redux
* Chai.js / Mocha.js
* Typescript & Node.js
  + What other frameworks?
  + Angular
  + Vue.js
  + Nest
* Ruby
* Go
* Android Dev
* Python
  + Django
* Machine Learning
* .Net FWs
  + ?
* Docker
  + Kubernetes
* Programming challenges
  + Leetcode - https://leetcode.com/problemset/top-100-liked-questions/
  + Khan Academy
  + <https://www.cscareerhackers.org/>
* Math
  + Pre-Calc class?

## Refactor Ideas

* Cypress Tests
  + Single Files
  + Independent
  + Clear Datasets
* JS Unit tests?
* Promises: Bluebird / Q / JQuery ?! can we pick one to stick with??
* Table Tools
  + Componentize
  + Evaluate what's actually applicable & where
    - See if these chunks of functionality could be separated
* Solution + SolutionTool + SolutionDashboard
  + Re-evaluate what belongs where
    - Lots of functionality is misplaced
    - Redo CheckToolConditions => tools have configs which optionally trigger publishes of events to enable/disable next etc
    - Solidify control of next button
* Analytic Service Controllers
  + Computation Model
  + Model Parameters?
  + Cache Entities?
  + Simultaneous Requests => mutexes?
* Dropzone : create custom component for panel, solution-tool, solution-dashboard
* Alert-element : custom component, created but not used?
* model.ts => mechanism for linking tools for refreshes
* npm dependencies => audit & clean