**Monday May 13th, 2019 Daily Coding Journal**

NOTE: The following journal is working through exercises from this course: <https://scrimba.com/playlist/p7P5Hd>and today is focused mostly if not entirely on the following video: <https://scrimba.com/p/p7P5Hd/cgDqBHP>

9:58 — A pomodoro session a day keeps the poor man’s house away… or something like that. Anyway, here we are again ready to do some more learning!

10:07 — I’ve been trying to rebuild the solution from scratch. Adding a console log in my handleChange function is easy enough. Adding a prop of *handleChange={this.handleChange}* is also easy enough.

10:08 — The thing I don’t get however, is exactly how the following line of code should go:

onChange={() => this.handleChange(props.item.id)}

10:09 — I spent probably 30 minutes on just this line of code yesterday but I don’t seem to have fully retained what needs to be done here. I know that first we have our onChange event handler that’s listening for some event to occur. Then we have an empty event with a function that should return the values being returned from our handleChange method. I’m making some small error in syntax, however, that resulting in an error that handleChange is undefined, however.

10:14 — Apparently the working solution is:

onChange={() => props.handleChange(props.item.id)}

10:15 — I find this pretty bizarre though as I’m almost sure I went through the following logic at least once. Oh well, I’ll get it next time.

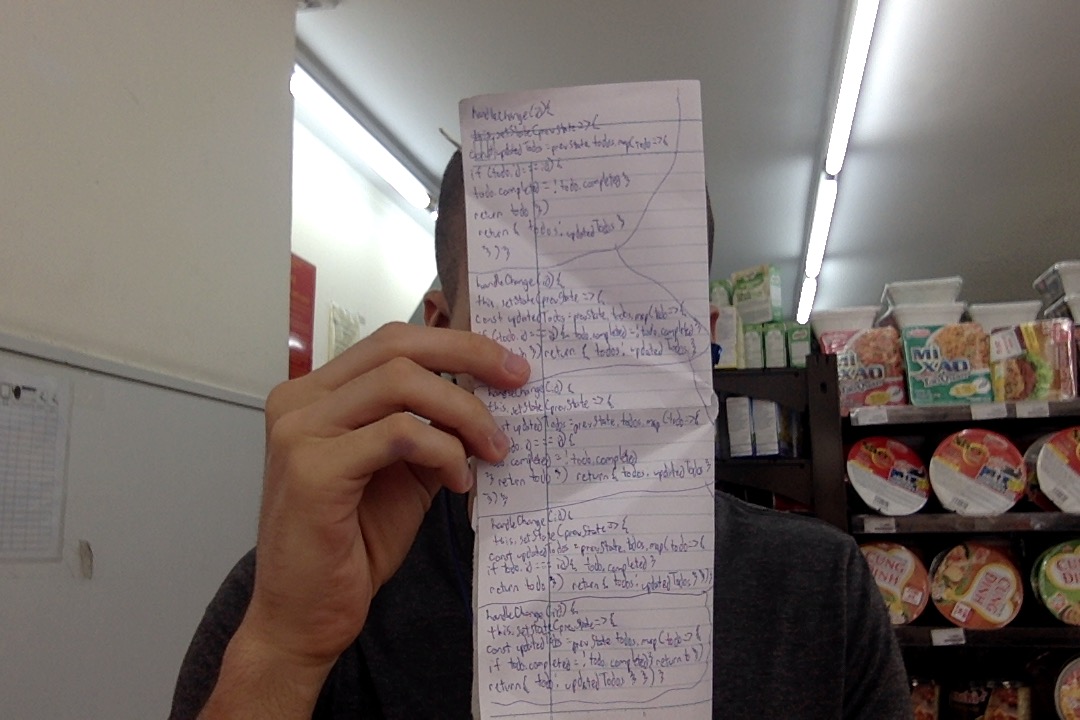
10:19 — Now I’m doing some work with .map( ) to see if I can change the state of my todo-items from true to false.

10:21 — If I console.log( ) the following:

console.log(this.state.todos)

I’m returned a huge list of data, which is exactly what I’m looking for.

10:25 — While console.logging the data I need is easy enough, I need to figure out how to use the map method without it returning a bunch of null values.

10:30 — I’m going to spend some time copying code by hand to help meditate on and gain a better understanding of both syntax, and *why* the code is doing what it is doing.

10:44 — While I kind of understand the above code, I’m not yet sure that I could replicate it on my own. I think I’ll need some more practice to do so. On the bright side, while this pomodoro session is over, I’m going to commit to coding for another hour or so today!

22:22 — Well, I’m certain that I’m not going to get in another hour of coding in today. I’m not sure what’s going on, but the ball on the back of my neck is swollen up and it feels quite painful for me to move around at all. I’m not suuuper worried, because this is the first time I can remember enduring this kind of pain. With that being said, I’m not going to be programming too much tonight because of that. Slow and steady wins the race. Get me up to an hour or so of total programming today, and then I’ll rest up and come back stronger for tomorrow.

22:27 — I’m working through the problem again. I’ve already gotten halfway through.

22:34 — I’ve gotten everything fairly solid except for the handleChange function. That part I haven’t figured out yet:

import React from "react"

import TodoItem from "./TodoItem"

import todosData from "./todosData"

class App extends React.Component {

constructor() {

super()

this.state = {

todos: todosData

}

this.handleChange = this.handleChange.bind(this)

}

handleChange(id) {

this.setState(prevState => {

const updatedTodos = this.todos.map(todoItem => {

if (todoItem.id == id) {

console.log("its a match")

}

return todoItem

})

})

}

render() {

const todoItems = this.state.todos.map(item => <TodoItem key={item.id} item={item} handleChange={this.handleChange}/>)

return (

<div className="todo-list">

{todoItems}

</div>

)

}

22:39 — I got some advice from an old Saigon friend earlier today. Amongst some other pieces of career advice, one thing he mentioned was that to improve this journal, I should change up the formatting. He suggested that to make this journal more programmer-friendly that I should use a web-readable format like markdown (.md). He also recommended that I can use VS Code or sublime to edit my journal in. I’ll probably roll with sublime-text as that’s something I’ve already used on at least one occasion. Apart from this, he also suggested that using a sortable date format would make it easier to find relevant files down the road.

22:47 — The above were some great tips. I’ll get started with markdown (.md) and sublime-text tomorrow!

**Total time spent coding today: 1 hour 15 minutes**

**Total time spent coding thus far in May 2019: 24 hours 38 minutes**

**Total lifetime hours of coding: 520 hours 32 minutes**