**Friday May 3rd, 2019**

8:48 — I’m awake and I’ve successfully made it to my computer without any distractions.

9:08 — I’ve been trying what feels like forever to figure out the correct syntax for if else statements in React. Finally, I decided that searching anymore would be less productive than simply getting the answer straight from the course. Then… the instructor decided to simply use a ternary operator in this video lol.

9:11 — Here is his syntax: {{display: props.question ? "block" : "none"}}

9:16 — Here’s a code sample I just wrote to make it easier to distinguish whether my jokes are delivered as standalone punchlines or delivered in question + punchline format:

<h3 style={{color: props.question ? "black" : "green"}}>Answer: {props.punchLine}</h3>

9:17 — The above code will render standalone punchlines as green and all other text as black.

9:24 — I just finished watching a video about using Map in React. While I have a decent understanding of Map in JavaScript, using it in this new context makes me feel a little bit uneasy. When I come back for another learning session later on today, I’m going to rewatch the video and try to apply something similar from scratch.

11:12 — Let’s get back to it.

11:15 — I’m rewatching the Mapping Components video. The author says that in an app, most of the data you’ll be displaying to the page won’t be hard coded into your components. In reality, most of the data will be coming from some type of HTTP call to an API that will reach out to a server and database that will return JSON (JavaScript Object Notation) to you.

11:23 — When we are using repeated components such as in the case of a higher order function like .map ( ), React prefers that we give each child in an array a unique key prop.

11:29 — Now the instructor (who I’ll refer to as ‘Bob’ or ‘Bob Ziroll’ from this point forward) wants me to practice using .map ( ) on my own.

11:50 — I’ve been diligently working the last 20 or so minutes. I made a couple mistakes I’ve since corrected. One was I was trying to work with props, but I forgot to give the appropriate function a props parameter. DUH. Another thing I was doing wrong was I was trying to access the props of a function by using dot notation on the name of the function itself rather than the props parameter. Below is the proper code:

function Product (props) {

return (

<div>

<h2> {props.name} </h2>

<h3> {props.price} </h3>

<h4> {props.description} </h4>

</div>

)

}

11:53 — Previously, however, I was trying to nest Product.name within my h2 which only returned the string, “Product” within my App.js file rather than the name I passed.

12:11 — Just came back from a 15 minute bathroom/meditation break. Let’s continue working.

12:22 — I went back and watched and old FunFunFunction video on .map ( ) to refresh my knowledge. Then I went to [repl.it](http://repl.it) and practiced doing some work with .map ( ) from scratch.

12:30 — I’m now successfully console logging the product data I need. I just need to work it into my map function now.

12:37 — So now I’ve got a .map ( ) function that looks like this:

const productComponents = vschoolProducts.map(function(product){

return product

})

console.log(productComponents[0])

12:59 — I’m confused beyond confused. I’ve been working on this for an hour. Even going so far as to use the exact same template as Bob. Even with that being the case, I’m still getting errors.

1:04 — I watched his method for completing the exercise and it looks like the issue was here:

function Product(props) {

return (

<div>

<h2>{props.product.name}</h2>

<p>{props.product.price.toLocaleString("en-US", { style: "currency", currency: "USD" })} - {props.product.description}</p>

</div>

)

}

In his code he writes props.product.name whereas in my code I just wrote props.name which basically caused all of my values to be null when mapped them. Why did this behavior occur? I honestly don’t know, and remain somewhat confused. With that being said, I’m going to exercise and come back to all of this with a clear head later when my attention is sharp.

**Total time spent coding today: N/A**

**Total time spent coding in May 2019: N/A**