**Profile challenge @ 20 June 2019**

Create-react-app profile challenge

Yarn start

App.js, add in the lesson html under here (all of the Html must be wrapped in the <div>)

function App() {

  return (

    <div>

It has be valid XML, not HTML. Self closing tags must have backslashes (i.e. <img />)

Go into app.css, remove all pre-existing css.

All css will be scoped. This style will not affect everything. It will only affect the file that it is imported into (i.e. app.js)

Create new file header.jsx

Add the following into header.jsx

import React from "react"

export default function Header(props) {

  return (

    <header>

      <h1>Welcome To Sandras Profile Page</h1>

      <nav>

        <ul>

          <li>

            <a href='#about-me'>About Me</a>

          </li>

          <li>

            <a href='#fav-foods'>Favourite Foods</a>

          </li>

          <li>

            <a href='#contact'>Contact Details</a>

          </li>

        </ul>

      </nav>

    </header>

  )

}

In App.js import Header and add the Header component

import Header from "./Header.jsx"

function App() {

  return (

    <div>

      <Header />

**Make the whole <nav> a component**

Create Nav.jsx, and add the following

import React from "react"

export default function Nav(props) {

  return (

    <nav>

      <ul>

        <li>

          <a href='#about-me'>About Me</a>

        </li>

        <li>

          <a href='#fav-foods'>Favourite Foods</a>

        </li>

        <li>

          <a href='#contact'>Contact Details</a>

        </li>

      </ul>

    </nav>

  )

}

In header.jsx, add the following

import Nav from "./Nav"

***Make a component as generic/reusable as possible in as many scenarios as possible***

* Get the information from elsewhere and ensure the component doesn’t have to be re-coded in every instance

In Header.jsx, replace “Sandra” with

{props.name}

In App.jsx, replace the <Header> component call with

      <Header name='Sandra' />

The approach: Write the part where you send the props in (set up the data structure) and then work out how to deal with the data in the component itself.

In Header.jsx where the call to the component <nav/> is made, add to the top below the import statements

Establish your data structure

const menuItems = [

  { title: "About me", link: "#about-me" },

  { title: "Favourite Foods", link: "#fav-foods" },

  { title: "Contact", link: "#contact" }

]

In Header.jsx, amend the <Nav /> call to pass the data structure through

      <Nav items={menuItems} />

To loop through, in Nav.jsx, start with {}. Can’t use forEach to loop through in JSX so we use .map . Amend the Nav function’s <nav> as

    <nav>

      <ul>

        {props.items.map(item => {

          return (

            <li>

              <a href={item.link}>{item.title}</a>

            </li>

          )

        })}

      </ul>

    </nav>

The data structure needs to be initialised where the call to the component is made.

IN app.js, take out the About me html section and replace with component call under the <Header />

      <AboutMe />

The process for building a site from scratch is still the same as this exercise. It involves breaking the wireframe down into components on react. And find situations where a component can be broken down into smaller components.

Create AboutMe.jsx, add the following

import React from "react"

export default function AboutMe(props) {

  return (

* Paste in the about me html section directly under the return statement, enclosed with round parentheses.
* Take out the imag url in the html and add it to the AboutME component call

**Working on the bio section**

In app.js add

const person = [

  "Pop culture maven. Beer fan. Award-winning music junkie. Extreme coffee  enthusiast.Thinker.Tv specialist.",

  "Friendly web maven. Bacon lover. General internet specialist. Incurable travel scholar",

  "Subtly charming twitter lover. Social media fan. Incurable travel geek. Lifelong pop culture specialist. Tv scholar.",

  "Unable to type with boxing gloves on. Proud bacon fan. Music junkie. Coffee ninja. Beer specialist"

]

In App.js, add the <AboutMe /> call under the <Header /> call

      <AboutMe image='https://randomuser.me/api/portraits/women/8.jpg' bio={person} />

Pass in the image url and also the array called person using the attribute bio.

In AboutMe.jsx, update the AboutMe function with the following

export default function AboutMe(props) {

  return (

    <section id='about-me'>

      <h2>About Me</h2>

      <img src={props.image} />

      <article>

        <ol>

          {props.bio.map(item => (

            <li>

              <p>{item}</p>

            </li>

          ))}

        </ol>

      </article>

    </section>

  )

}

In the .map expression, avoid using curly braces where there is simply an assignment rather than an expression. As in there is no need for an extra set of curly braces after the arrow operator in the .map expression above.

Update the AboutMe.jsx with the following to make the first word bold without mutating the string input from app.js

export default function AboutMe(props) {

  return (

    <section id='about-me'>

      <h2>About Me</h2>

      <img src={props.image} />

      <article>

        <ol>

          {props.bio.map(item => {

            const firstWord = item.split(" ")[0]

           item = item.replace(firstWord, "")

            return (

              <li>

                <p>

                  <strong>{firstWord}</strong>

                  {item}

                </p>

              </li>

            )

          })}

        </ol>

      </article>

    </section>

  )

}

To continue individually.