**React @ 20 Jun 2019**

Create-react-app jsx

Modify app.js with

const message = "App is running!"

function App() {

  return <div>{message}</div>

}

Amend it further

  return <div style={{ color: "red", fontSize: "30px" }}>{message}</div>

It is double braces

* Outer brace are script tags for JSX
* Inner brace is the style object

To contain the styles, app.js can be amended as per follows

function App() {

  const styles = {

    color: "red",

    fontSize: "30px",

    backgroundColor: "black",

    border: "5px solid yellow"

  }

  return <div style={styles}>{message}</div>

}

Alternatively, you can add the styles into the app.css and then call it in the app.js by declaring the style as a class attribute

function App() {

  return <div className='container'>{message}</div>

}

Create Comment.jsx

Inside Comment.jsx add the following

import React from "react"

const Comment = props => {

  return (

    <div className='comment'>

      <a href='/' className='avatar'>

        <img alt='avatar' src={faker.image.avatar()} />

      </a>

      <div className='content'>

        <a href='/' className='author'>

          {props.author}

        </a>

        <p className='date'>11/12 6:00pm</p>

        <p className='text'>I think you are really cool!</p>

      </div>

    </div>

  )

}

This code uses Faker. Faker is a package that generates random data for testing purposes with a number of categories, profiles for individuals.

Install faker

* yarn add faker
* npm i faker

ON Comment.jsx, import faker

import faker from "faker"

In Comment.jsx, amend the code for date and text with props call

        <p className='date'>{props.date}</p>

        <p className='text'>{props.text}</p>

Add at the bottom of comment.jsx, export

export default Comment

In App.js, import comment

import Comment from "./Comment"

in app.js, update the app function as per below

function App() {

  return (

    <div className='comments'>

      <Comment author='Mary Smith' date={"11/12 6:00pm"} text={"I think you are really cool!"} />

      <Comment author='Jude Henry' date={"09/12 1:30pm"} text={"Hey this is great!"} />

      <Comment author='Bobby Miller' date={"07/12 9:10am"} text={"Simply spectacular."} />

    </div>

  )

}

**Rendering react components as children**

Redner the component called <Card> and passing through the enclosed HTMl as children

A component could pass through other components (Nesting components)

In Comment.jsx, amending the comment function to

import Card from "./Card"

const Comment = props => {

  return (

    <Card>

      <a href='/' className='avatar'>

        <img alt='avatar' src={faker.image.avatar()} />

      </a>

      <div className='content'>

        <a href='/' className='author'>

          {props.author}

        </a>

        <p className='date'>{props.date}</p>

        <p className='text'>{props.text}</p>

      </div>

    </Card>

  )

}

Create Card.jsx

import React from "react"

in Card.jsx, add styles variable

const styles = {

  backgroundColor: "gray",

  padding: "1rem",

  margin: "1rem",

  border: "1px solid black",

  borderRadius: "0.5rem"

}

In Card.jsx, create the actual cards,

const Card = props => {

  return <div style={styles}>Card Component</div>

}

export default Card

But currently not displaying the children (the details of each person)

React will pass in the html as children. Amend Card.jsx as per follows

const Card = props => {

  return <div style={styles}>{props.children}</div>

}

Nested components don’t follow a hierarchy. Once you have created it you can use it anywhere within the application as long as you import it into the file. So Card can be used in app.js or Comment.jsx even if it is nested.

import Card from "./Card"