Props:

To pass props:

import Header from "./components/Header";

**function** App() {

  return (

    <div>

      <Header text={Hello} />

    </div>

  );

}

export default App;

Header:

import PropTypes from 'prop-types'

**function** Header({ text }) {

  return (

    <header>

      <div className="container">

        <h1>{text}</h1>

</div>

    </header>

  )

}

Header.defaultProps = {

  text: "Feedback UI"

}

Header.propTypes = {

  text: PropTypes.string.isRequired

}

export default Header

To set default props:

Header.defaultProps = {

  text: "Feedback UI"

}

To define prop types, first import PropTypes (impt snippet):

import PropTypes from 'prop-types'

And define them:

Header.propTypes = {

  text: PropTypes.string.isRequired

}

To add inline style:

    <header style={{backgroundColor:'blue', color:"red"}}>

Defining using object:

**const** headerStyles = {

    backgroundColor: 'blue',

    color: "red"

  }

  return (

    <header style={headerStyles}>

Defining style using props:

App.js:

import Header from "./components/Header";

**function** App() {

  return (

    <div>

      <Header bgColor="red" textColor="blue" />

    </div>

  );

}

export default App;

Header.jsx:

**const** headerStyles = {

    backgroundColor: bgColor,

    color: textColor

  }

  return (

    <header style={headerStyles}>

State

Two types:

* Global state
* Component level state

useState example of changing the value:

**const** handleClick = () **=>**{

    setRating((prev) **=>** {

      return prev + 1

    })

  }

prev – is the previous value in state. So if you need it you have declare a function inside the setState.

Otherwise you can just write it like this:

**const** handleClick = () **=>**{

    setRating(

      return 1

    )

  }

props.children – allows to render the content included in between opening and closing brackets:

import Card from "./shared/Card"

**function** FeedbackItem({item}) {

  return (

    <Card reverse={true}>

      <div className="num-display">{ item.rating}</div>

      <div className="text-display">{item.text}</div>

    </Card>

  )

}

export default FeedbackItem

Card component:

**function** Card({children, reverse}) {

  return (

    <div className="card">{ children}</div>

  )

}

export default Card

If you need to pass argument on the click event write it like this:

      <button onClick={() **=>** handleClick(item.id)} className="close">

Example of calculating average:

**let** average = feedback.reduce((acc, cur) **=>** {

    return acc + cur.rating

  }, 0) / feedback.length

 average = average.toFixed(1).replace(/[.,]0$/,'')

regular expression removes trailing zeroes.

Context – way to pass data without having to pass it manually (from one component to another)