Ryan Snell

Web Development II

Exercise I

1. How to bind a function to a Component event?

You accomplish this through the use of hooks. They look as shown below.  
  
class Hello extends React.Component {

constructor(props) {

super(props);

}

render() {

return(

<div>

Hello {props}

</div>

)

}

}

1. What is a stateful variable?  
     
   Stateless: means there is no memory of the past. Every transaction is performed as if it were being done for the very first time.

Stateful: means that there is memory of the past. Previous transactions are remembered and may affect the current transaction.

1. How to create a stateful variable?  
     
   const [count,setCount] = useState(0);
2. How to pass stateful variable to a child Component?

handleCount(value) {

this.setState({count: this.state.count+100});

this.setState({count: this.state.count+value});

this.setState({count: this.state.count-100});

}

Or

(prevState, props) => stateChange

1. How to change a stateful variable from a child Component?  
     
   class Parent extends Component {

constructor(props){

super(props);

this.state = {

value: ""

}

}

changeValue(value){

this.setState({value});

}

render(){

return (

<div>

<Child1 value={this.state.value}/>

<Child2 changeValue={changeValue}/>

</div>

)

}

}

class Child2 extends Component {

constructor(props) {

super(props);

this.state={

input: ""

}

}

handleChange(e){

var {value} = e.target;

this.setState({

input: value

},() => this.props.changeValue(value));

}

render(){

return (

<div>

<input value={this.state.input} onChange={this.handleChange}/>

</div>

)

}

}

class Child1 extends Component {

constructor(props) {

super(props);

this.state={value:''}

}

componentWillReceiveProps(nextProps) {

this.setState({value: nextProps.value})

}

render(){

return (

<div>

{this.props.value}

</div>

)

}

}