**Comparing Jousi.js to React.js**

Creating a simple button whose innerHTML changes depending on previous state:

|  |  |
| --- | --- |
| Jousi.js: | React.js: |
| let state = "ON";  const changeIt = (el) => {  el.innerHTML = state=="ON" ? "OFF" : "ON";  state = state=="ON" ? "OFF" : "ON";  }  let button = {  "d:root" : ["button-c:my-button"]  }  let buttonStats = {  "my-button" : {  innerHTML : state,  onclick : "changeIt(this)"  }  }  Create(button, buttonStats); | class Toggle extends React.Component {  constructor(props) {  super(props);  this.state = {isToggleOn: true};  this.handleClick = this.handleClick.bind(this);  }  handleClick() {  this.setState(prevState => ({  isToggleOn: !prevState.isToggleOn  }));  }  render() {  return (  <button onClick={this.handleClick}>  {this.state.isToggleOn ? 'ON' : 'hello'}  </button>  );  }  }  ReactDOM.render(  <Toggle />,  document.getElementById('root')  ); |

|  |  |
| --- | --- |
| Alternative Jousi.js: | React.js: |
| let state = "ON";  let button = {  "d:root" : ["button-c:my-button"]  }  let buttonStats = {  "my-button" : {  innerHTML : state,  onclick : `const changeIt = (el) => {  el.innerHTML = state=="ON" ? "OFF" :  "ON"; state = state=="ON" ? "OFF" : "ON";  }; changeIt(this);`  }  }  Create(button, buttonStats); | class Toggle extends React.Component {  constructor(props) {  super(props);  this.state = {isToggleOn: true};  this.handleClick = this.handleClick.bind(this);  }  handleClick() {  this.setState(prevState => ({  isToggleOn: !prevState.isToggleOn  }));  }  render() {  return (  <button onClick={this.handleClick}>  {this.state.isToggleOn ? 'ON' : 'hello'}  </button>  );  }  }  ReactDOM.render(  <Toggle />,  document.getElementById('root')  ); |

|  |  |
| --- | --- |
| OOP in Jousi.js: | React.js: |
| class ButtonChange {  constructor () {  this.state = "ON";  this.render();  }  render() {  let button = {  "d:root" : ["button-c:my-button"]  }  let buttonStats = {  "my-button" : {  innerHTML : this.state,  onclick : `const changeIt = (el) => {el.innerHTML = this.state=="ON" ? "OFF" : "ON"; this.state = this.state=="ON" ? "OFF" : "ON"; }; changeIt(this);`  }  }    Create(button, buttonStats);  }  }  new ButtonChange(); | class Toggle extends React.Component {  constructor(props) {  super(props);  this.state = {isToggleOn: true};  this.handleClick = this.handleClick.bind(this);  }  handleClick() {  this.setState(prevState => ({  isToggleOn: !prevState.isToggleOn  }));  }  render() {  return (  <button onClick={this.handleClick}>  {this.state.isToggleOn ? 'ON' : 'hello'}  </button>  );  }  }  ReactDOM.render(  <Toggle />,  document.getElementById('root')  ); |