

LEARNING FACEBOOK'S



React.js

React Components Props & State

Lesson 03

Lesson Objectives

At the end of this module you will be able to Learn:

- Understanding and using Props and State
- Handling Events with methods
- Manipulating the State
- Two way data-binding
- Functional (Stateless) VS Class (Stateful) Components
- Parent – Child Communication
- Dynamically rendering contents
- Showing Lists, List and keys

LEARNING FACEBOOK'S



React.js



Understanding and using Props and State

- Props:
 - Its shorthand for properties.
 - Using props, components talk to each other i.e., they share data between each other.
 - Props are immutable.
 - All Props can be accessible with `this.props`
- `React.PropTypes` :
- `React.PropTypes` are used to run type checking on the props for a component.
- Allows you to control the presence, or types of certain props passed to the child component.
- **Default Props:**

```
App.defaultProps = {  tech: 'React Js'  };
```



In PropEg.js file

```
import React from 'react';
import ReactDOM from 'react-dom';

class PropEg extends React.Component {
  render() {
    return(
      <div>
        <h1>{this.props.appName}</h1>
        <img src={this.props.prod.img}/>
        <h2>{this.props.prod.prodName}</h2>
        <h2>{this.props.prod.price}</h2>
        <h2>{this.props.prod.qty}</h2>
      </div>
    );
  }
}
export default PropEg;
```

In index.js file

```
import React from 'react';
import ReactDOM from 'react-dom';
import PropEg from './PropEg';
import imag from './dot.png';

var product1={prodName:"Echo Dot", price:"3500", qty:"6",img:imag};

ReactDOM.render(<PropEg appName="Shopping cart!!!" prod={product1}/>
  , document.getElementById('root'));
```



- React components can be made dynamic by adding state to it.
- State is used when a component needs to change independently of its parent.
- React component's initial state can be populated using `getInitialState()` with an object map of keys to values.
- React component's state can be accessed using `this.state`
- React component's state can be updated using `this.setState()` with an object map of keys which can be updated with new values. Keys that are not provided are not affected.
- `setState()` merges the new state with the old state.
- As a best practice the nested React components should be stateless. They should receive the state data from their parent components via `this.props` and render that data accordingly.

Demo



State:

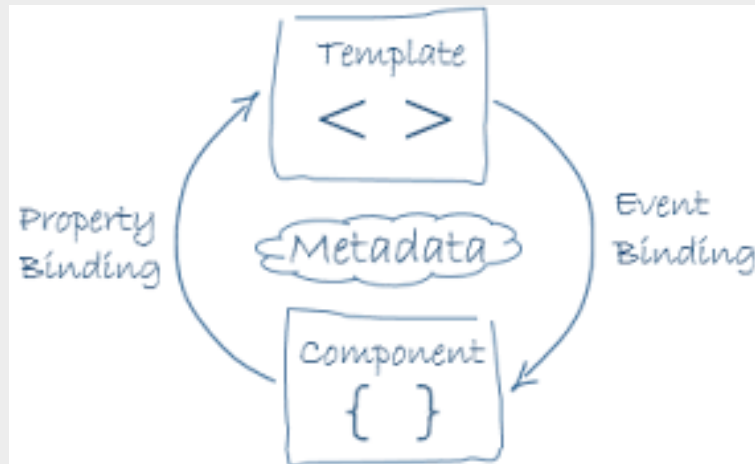
react-state-2-2019
react-state-2019



Two way data-binding



- Two Binding is a mechanism for coordinating parts of a template with parts of a component.
- When properties in the model get updated, so does the UI.
- When UI elements get updated, the changes get propagated back to the model.



- Angular supports two way binding
- React Js supports one way binding only (uni-directional flow).
- React is more performant than angular.

Functional (Stateless) VS Class (Stateful) Components



Functional component :

- A functional component is just a plain JavaScript function which accepts props as an argument and returns a React element.
- Generally component doesn't have its own state.
- Easy write, understand and test.
- Avoid complete use of '**this**' keyword
- Major 2 drawback
 1. Lifecycle hooks is not supported
 2. Refs are also not supported.

Class component [StateFul Components] :

- A Class Component can be created by extending React.Component
- Inside which we create a render function which returns a React element
- State can be initialized by constructor
- Advantage is we can use lifecycle hook

Demo



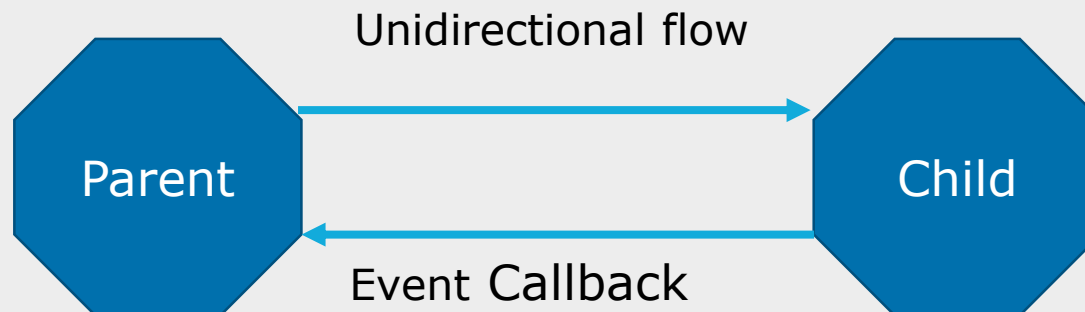
Demo of
react-functional-class-component2019



Parent – Child Communication



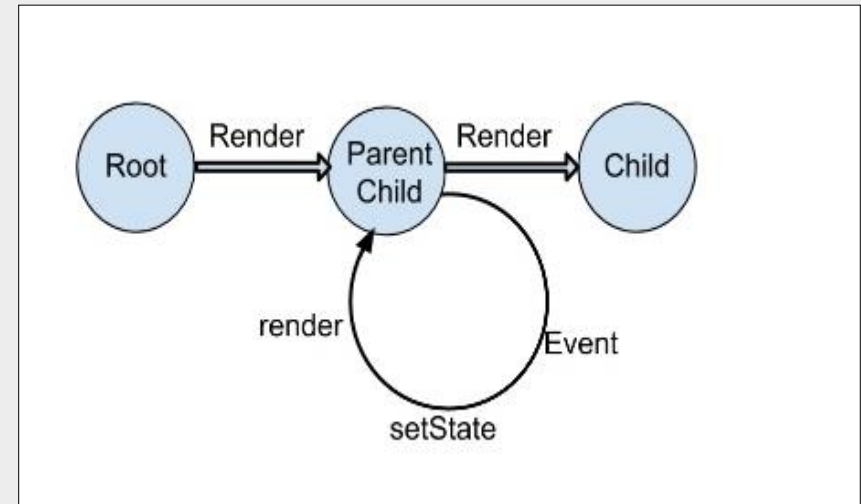
- In React greatest advantage is using 'uni-directional flow'
- Data is passed from Parent to Child components
- A child component can never send a prop back to the parent component that is called unidirectional data flow.
- If we can pass the data from child to parent. You can use callback and state in your application.





Unidirectional data flow

- In React, application data flow is unidirectional via the state and props objects, as opposed to the two-way binding of libraries like Angular.
- In a multi component hierarchy, a common parent component will manage the state and pass it down the chain via props.
- State should be updated using the `setState` method to ensure that a UI to update and the resulting values should be passed down to child components using attributes that are accessible in said children via `this.props`



Dynamically rendering contents



Conditional rendering in React works the same way conditions work in JavaScript. Use JavaScript operators like if or the conditional operator to create elements representing the current state, and let React update the UI to match them.

```
function Product(props) {  
  const available= props.available;  
  if (available){  
    return <SubProduct />;  
  }  
  return <NoProduct />;  
}  
  
ReactDOM.render(  
  <Product available={false} />,  
  document.getElementById('root')  
);
```

We can also make use of switch cases, if else, ternary operator etc,.

Showing Lists and Keys



Lists:

Assume that we have a huge set of numbers, say some 2000 numbers and we have to find the square of number and get it on the screen. Old way is to make use is using normal for loop.

So we have a `Array.map()` method function to create a new array that has the same number of elements, and where each element is the result of calling the function you provide.

```
<div>
  {props.items.map((item, index) => ( <Item key={index} item={item} /> ))}
</div>
```

Keys :

Keys help React identify which items have changed, are added, or are removed.

Keys should be given to the elements inside the array to give the elements a stable identity.

Rules for keys :

unique : Every item in the list must have a unique key

Permanent: An item's key must not change between re-renders, unless that item is different

Demo



Demo of

react-list_keys_2019
react-list-keys-2-2019





Summary

By now you would have got clear idea of

- Understanding and using Props and State
- Manipulating the State
- Two way data-binding
- Functional (Stateless) VS Class (Stateful) Components
- Parent – Child Communication
- Dynamically rendering contents
- Showing Lists, List and keys



Review Questions



- Question 1:

- _____question 1

- A)
- B)
- C)
- D)

- Question 2:

- q2

- A)
- B)
- C)
- D)