

## فرم‌ها (Forms) در ReactJS

در این بخش می‌خواهیم به چگونگی استفاده از فرم‌ها در ری‌اکت بپردازیم.

### مثال ساده

در مثال زیر می‌خواهیم یک فرم ورودی را به کمک `value = {this.state.data}` تنظیم کنیم. از این طریق می‌توانیم هر زمان که مقدار ورودی تغییر می‌کند، حالت را به روز رسانی کنیم. ما در اینجا از رویداد `onChange` استفاده کرده ایم. این رویداد تغییرات ورودی را زیر نظر می‌گیرد و حالت را بر اساس آن به روز رسانی می‌کند.

### App.jsx

```
import React from 'react';

class App extends React.Component {

  constructor(props) {

    super(props);

    this.state = {

      data: 'Initial data...'

    }

    this.updateState = this.updateState.bind(this);

  };

  updateState(e) {

    this.setState({data: e.target.value});

  }

  render() {

    return (
```

```

<div>

  <input type = "text" value = {this.state.data}

    onChange = {this.updateState} />

  <h4>{this.state.data}</h4>

</div>

);

}

}

export default App;

```

## main.js

```

import React from 'react';

import ReactDOM from 'react-dom';

import App from './App.jsx';

ReactDOM.render(<App/>, document.getElementById('app'));

```

زمانی که مقدار متن ورودی تغییر می کند، حالت به روز رسانی می شود.



## مثال پیچیده

در مثال زیر چگونگی استفاده از فرم ها از جزء فرزند نشان داده شده است. متد `onChange` به روز رسانی حالت را فعال می کند. سپس این به روز رسانی به مقدار ورودی فرزند داده می شود و در صفحه نمایش داده می شود. مثال مشابه در بخش رویدادها آورده شده است. هر زمان که نیاز باشد حالت را از جزء فرزند به روز رسانی

کنیم، باید تابعی که به روز رسانی (updateState) را به صورت یک ویژگی (updateStateProp) مدیریت می کند را عبور دهیم.

## App.jsx

```
import React from 'react';

class App extends React.Component {

  constructor(props) {

    super(props);

    this.state = {

      data: 'Initial data...'

    }

    this.updateState = this.updateState.bind(this);

  };

  updateState(e) {

    this.setState({data: e.target.value});

  }

  render() {

    return (

      <div>

        <Content myDataProp = {this.state.data}

          updateStateProp = {this.updateState}></Content>

      </div>

    );

  }

}
```

```
class Content extends React.Component {  
  
  render() {  
  
    return (  
  
      <div>  
  
        <input type = "text" value = {this.props.myDataProp}  
  
          onChange = {this.props.updateStateProp} />  
  
        <h3>{this.props.myDataProp}</h3>  
  
      </div>  
  
    );  
  
  }  
  
}  
  
export default App;
```

## main.js

```
import React from 'react';  
  
import ReactDOM from 'react-dom';  
  
import App from './App.jsx';  
  
ReactDOM.render(<App/>, document.getElementById('app'));
```

این کار باعث می شود نتیجه ی زیر حاصل شود.

