

## انیمیشن‌ها (Animation) در ReactJS

در این بخش می‌خواهیم به چگونگی متحرک کردن عناصر با استفاده از ReactJS بپردازیم.

### مرحله 1: نصب React CSS Transitions Group

این برنامه افزونه‌ای برای ReactJS بوده که با کمک آن می‌توان انیمیشن‌ها و گذارهای CSS اولیه را ایجاد کرد. برای نصب آن از پنجره ی cmd کمک می‌گیریم.

```
C:\Users\username\Desktop\reactApp>npm install react-addons-css-transition-group
```

### مرحله 2: اضافه کردن یک فایل CSS

بیا یک فایل جدید style.css را ایجاد کنیم.

```
C:\Users\Tutorialspoint\Desktop\reactApp>type nul > css/style.css
```

برای آن که بتوانیم در برنامه ی خود از این فایل استفاده کنیم، باید آن را به عنصر هد index.html پیوند دهیم.

```
<!DOCTYPE html>
<html lang = "en">
  <head>
    <link rel = "stylesheet" type = "text/css" href = "./style.css">
    <meta charset = "UTF-8">
    <title>React App</title>
  </head>
  <body>
    <div id = "app"></div>
    <script src = 'index_bundle.js'></script>
  </body>
</html>
```

### مرحله 3: ظاهر کردن انیمیشن

حالا می‌خواهیم یک جزء ابتدایی ReactJS را ایجاد کنیم. از عنصر ReactCSSTransitionGroup به عنوان پوشاننده ی جزئی که ما می‌خواهیم آن را متحرک کنیم استفاده می‌شود. در اینجا از transitionAppear و

، transitionLeave و transitionEnter در حالی که transitionAppearTimeout false هستند.

## App.jsx

```
import React from 'react';

var ReactCSSTransitionGroup = require('react-addons-css-transition-group');

class App extends React.Component {

  render() {

    return (

      <div>

        <ReactCSSTransitionGroup transitionName = "example"

          transitionAppear = {true} transitionAppearTimeout = {500}

          transitionEnter = {false} transitionLeave = {false}>

          <h1>My Element...</h1>

        </ReactCSSTransitionGroup>

      </div>

    );

  }

}

export default App;
```

## main.js

```
import React from 'react'
import ReactDOM from 'react-dom';
import App from './App.jsx';

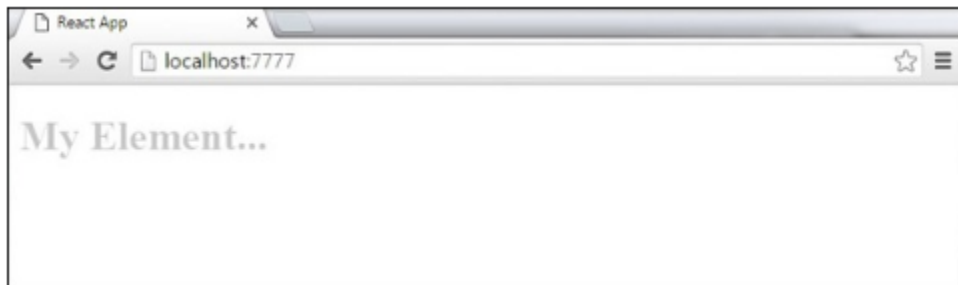
ReactDOM.render(<App />, document.getElementById('app'));
```

متحرک سازی CSS خیلی ساده است.

## css/style.css

```
.example-appear {  
  opacity: 0.04;  
}  
.example-appear.example-appear-active {  
  opacity: 2;  
  transition: opacity 50s ease-in;  
}
```

بعد از باز کردن برنامه عنصر مورد نظر ما محو می شود.



## مرحله 4: متحرک سازی ورود و خروج

از این متحرک سازی ها می توانیم در زمانی استفاده کنیم که نیاز است عناصری را از لیست حذف و یا به آن اضافه کنیم.

## App.jsx

```
import React from 'react';  
  
var ReactCSSTransitionGroup = require('react-addons-css-transition-group');  
  
class App extends React.Component {  
  
  constructor(props) {  
  
    super(props);  
  
  }  
  
}
```

```
this.state = {  
  items: ['Item 1...', 'Item 2...', 'Item 3...', 'Item 4...']  
}  
  
this.handleClick = this.handleClick.bind(this);  
};  
  
handleAdd() {  
  var newItems = this.state.items.concat([prompt('Create New Item')]);  
  this.setState({items: newItems});  
}  
  
handleRemove(i) {  
  var newItems = this.state.items.slice();  
  newItems.splice(i, 1);  
  this.setState({items: newItems});  
}  
  
render() {  
  var items = this.state.items.map(function(item, i) {  
    return (  
      <div key = {item} onClick = {this.handleClick.bind(this, i)}>  
        {item}  
      </div>  
    );  
  }).bind(this);  
  
  return (  
    <div>  
      <button onClick = {this.handleClick}>Add Item</button>  
    </div>  
  );  
}
```

```
<ReactCSSTransitionGroup transitionName = "example"
  transitionEnterTimeout = {500} transitionLeaveTimeout = {500}>
  {items}
</ReactCSSTransitionGroup>
</div>
);
}
}
export default App;
```

## main.js

```
import React from 'react'
import ReactDOM from 'react-dom';
import App from './App.jsx';

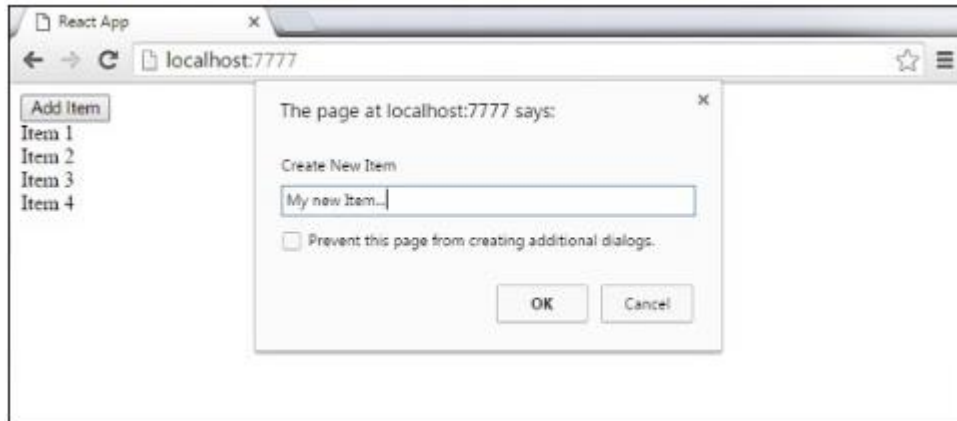
ReactDOM.render(<App />, document.getElementById('app'));
```

## css/style.css

```
.example-enter {
  opacity: 0.04;
}
.example-enter.example-enter-active {
  opacity: 5;
  transition: opacity 50s ease-in;
}
.example-leave {
  opacity: 1;
}
```

```
.example-leave.example-leave-active {  
  
  opacity: 0.04;  
  
  transition: opacity 50s ease-in;  
  
}
```

پس از باز کردن برنامه و کلیک کردن بر روی دکمه ی Add Item پیام زیر نمایش داده می شود.



بعد از وارد کردن اسم و OK کردن، عنصر جدید در حالت وارد شدن محو می شود.



حالا می توانیم با کلیک کردن بر روی آیتم ها، برخی از آن ها (Item 3...) را حذف کنیم. این کار باعث می شود این آیتم در حال خروج از لیست محو شود.

