

Migrating Your Existing Apps to React Hooks



Peter Kellner

DEVELOPER, CONSULTANT AND AUTHOR

@pkellner [linkedin.com/in/peterkellner99](https://www.linkedin.com/in/peterkellner99) PeterKellner.net

React Hooks only available
in React Functional
Components.

React Hooks can manage
state and lifecycle events.

Three Most Common Hooks



useState



useRef



useEffect

Four More Hooks



useContext



useReducer



useCallback



useMemo

Echo input demonstration
with history.

State

React Hooks

```
const [inputText, setInputText] =  
  useState('');  
  
onChange={e => {  
  setInputText(e.target.value);  
}}
```

Class Components

```
state = {  
  inputText: ''  
}  
  
handleChange = event => {  
  this.setState({  
    inputText: event.target.value  
  });  
};
```

Ref

React Hooks

```
const MyFunct = () => {  
  const imageRef = useRef(null);  
  return (  
      
  );  
};
```

Class Components

```
class MyComp  
  extends React.Component {  
  render() {  
    const imageRef = React.createRef();  
    return ;  
  }  
}
```


Lifecycle Events

React Hooks

```
const MyFunct = () => {  
  useEffect(() => {  
    console.log(  
      "mounting")  
    return () => {  
      console.log(  
        "dismounting")  
    };  
  }, [isLoading]);  
}
```

Class Components

```
class MyComp extends React.Component {  
  componentDidMount() {  
    console.log("mounting");  
  }  
  componentDidUpdate(prevProps,  
                        prevState) {  
    console.log("dependency-chk");  
  }  
  
  componentWillUnmount() {  
    console.log("dismounting");  
  }  
}
```

useEffect Second Parameter, Dependency Array

Null

useEffect run on every invocation of component

Empty Array

useEffect run only on first invocation of component

Values

useEffect runs only when any of array values change