LEARN REACT JS

Full Course for Beginners - Tutorial 2019

PLATFORM: Youtube

TUTOR: Bob Ziroll

STARTED ON:
6th May 2022

DURATION:

FINISHED ON:
13th May 2022

▷ Creating Our React Application

- Open VS Code & pick gitbash for your terminal

→ node -v (If you don't have node. Install it)

v12.11.0

→ npm -v (Globally)

8.5.5

npx -v (Locally)

8.5.5

→ npx create-react-app . # Creates react app in an empty folder

- After app is created we can see in the terminal

npm start

This allows us to start up our development environment

npm run build

This is going to build our application, minify all of our files and give us a production version of our application that we can deploy to a server.

npm test

Starts the test runner

npm run eject

Let's Build a Really Simple Todo List Application

App.Js

```
import React from 'react';

function App() {
  return (
    <TodoList />        # This is going to be a seperate Component
  )
}
```

Note: This isn't normal HTML as you can see. It is JSX

What is JSX?

it is a Javascript language that allows us to write HTML in React

Let's create our ~~Kk~~ TodoList Component in src

TodoList.Js # Create this

```
import React from 'react'

export default function TodoList () {
  return (
    <div>
      Hello World
    </div>
  )
}
```

≡ Now let's head to our App.Js and import this Component

☐ App.Js

2 import TodoList from './TodoList' * Imported ☑

```
function App() {
  return (
    <TodoList />
    <input type="text" * />        # this will give us an error
  )
}

export default App;
```

Because we can't return 2 JSX Elements
it has to be wrapped in
<> ... </> (a fragment)

```
return (
  <>
    <TodoList />
    <input type="text">
  </>
)
```

Let's Add

☑ This will now work

```
<button> Add Todo </button>
<button> Clear Completed Todos </button>
<div> 0 left to do </div>
```

Hello World

[ ] Add Todo

Clear Complete

0 left to do

useState

Now we want to store all of our todos inside of a state so we can actually render those todos and everytime we change, delete or add a todo, it will actually re-render our entire component for us.

This is made possible using the "useState"

App.js

```
import React, { useState } from 'react';   # useState Imported

function App() {
  useState([])
  const [todos, setTodos] = useState(['Todo 1', 'Todo 2'])
  return (
    <>
      <TodoList todos={todos} />
      ....
    </>
```

Passed in →

TodoList.js

```
export default function TodoList({ todos }) {
  return (
    <div>
      {todos.length}
    </div>
  )
}
```

2 Add Todo

≡ Now we want to print out our todos individually. This can be done easily using a loop (a map loop) and a new Component we will be calling "Todo.Js"

☐ Todo.Js {* Shortcut rfc (generate function boilerplate)} * Create this

≡ Next: Let's Import our Todo Component and use it in our TodoList.Js

☐ TodoList.Js

```
2 import Todo from './Todo'
... ...          function Todo({ todos}) {
    return (
        todos.map(todo => {
            return <Todo todo={todo} />
        })
    )
```

≡ Back to our Todo.Js

☐ Todo.Js

...

```
return (
    <div>
        { todo }
    </div>
)
```

Todo 1	
Todo 2	☑

If you check the Console you will see an error telling you a key is needed for every single todo in other to re-render only the todos that change and not every single todo.

So let's add an id for our todos

App.Js

```
function App() {
  const [todos, setTodos] = useState([{ id: 1, name: 'Todo 1',
                                        Complete: false }])
                                                  * updated
  ...
}
```

Now we can add a key

TodoList.Js

```
...

return (
  todos.map(todo => {
    return <Todo key={todo.id} todo={todo} />   # updated
  })
```

Todo 1

Todo.Js

```
return (
  <div>
    {todo.name}
  </div>
)
}
```

#updated

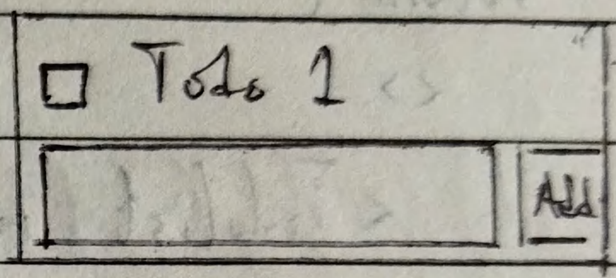
NEXT: We want a check box that works

Todo.Js

```
...

  return (
    <div>
      <label>
        <input type="checkbox" checked={todo.complete} />
        {todo.name}
      </label>
    </div>
  )
```

todo 1



Now we want to be able to add todos. To do this we need to first of all ~~do~~ 2 things

- Install random id Generator
  - npm i uuid
- import another hook called useRef to be able to access the values typed out in the input field

App.Js

```
1  import React, { useState, useRef } from 'react';   # updated
3  ... import { v4 as uuidv4 } from 'uuid'
   function App() {
```

```
Const [todos, setTodos] = useState([])
Const todoNameRef = useRef()                    # Added

function handleAddTodo(){                       # Added
    * This function will handle the adding of Todos
}
```

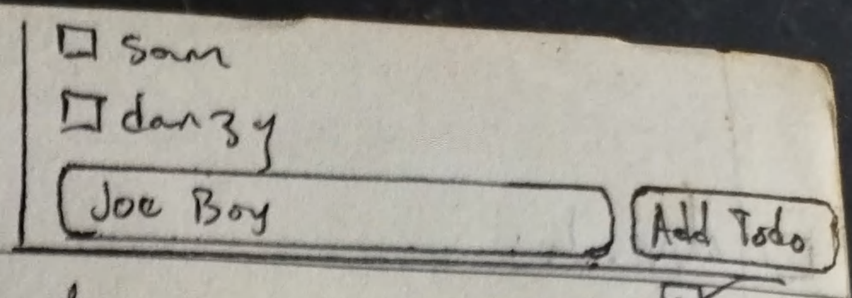
```
return (
  <>
    <Todolist todos={todos} />
    <input ref={todoNameRef} type="text" /> # updated
    <button onClick={handleAddTodo}>
    ...
  </>
```

≡ Now Let's Complete our handleAddTodo() Function

```
function handleAddTodo(){
    Const todoname = todoNameRef.current.value
    if (todoname === "") return

    setTodos(prevTodos => {
        return [...prevTodos, {id: uuidv4(), name: name,
                               Complete: false }]
    })
    todoNameRef.current.value = null
}
```

Next: Now whenever we reload our page. All our todos disapper. We can fix this using a hook called "useEffect"



App.Js

```
import React, { useState, useRef, useEffect } from 'react'
...

const LOCAL_STORAGE_KEY = "todoApp.todos"        * Added

function App() {
  const [todos, setTodos] = useState([])
  const todoNameRef = useRef()

  useEffect(() => {                              * Added
    localStorage.setItem(LOCAL_STORAGE_KEY,
      JSON.stringify(todos))
  }, [todos])
```

-----> Now this useEffect is called on any change of state of our todos

....

Now on page reload lets check our localStorage to see if there is any todo for us to add into our todos.

- Let's do this outside of our App() function

App.Js

...

```
const LOCAL_STORAGE_KEY = "todoApp.todos"
const storedTodos = JSON.parse(localStorage.getItem(
                                LOCAL_STORAGE_KEY))

if (storedTodos.length === 0) {
  storedTodos = []
}
```

```
function App() {
  const [todos, setTodos] = useState(storedTodos)   # updated
                                     passed in
  const ...
  const ...
  ...
}
```