**React JS Tutorial**

Contents

[React JS Tutorial 1](#_Toc142218334)

[Reference 4](#_Toc142218335)

[Section 1: Getting Started 5](#_Toc142218336)

[1. About ReactJS 5](#_Toc142218337)

[Section 2: JavaScript Refresher 5](#_Toc142218338)

[2. The Spread and Rest Operator 5](#_Toc142218339)

[3. De structuring 5](#_Toc142218340)

[4. References and Primitive Types 6](#_Toc142218341)

[5. Array Function 6](#_Toc142218342)

[Section 3: React Basic & Working with Components 7](#_Toc142218343)

[6. Module Introduction 7](#_Toc142218344)

[7. What are Components & why React is all about them. 7](#_Toc142218345)

[8. React code is written in declarative way! 9](#_Toc142218346)

[9. Create new React project. 10](#_Toc142218347)

[10. Analysing the standard React Project 11](#_Toc142218348)

[11. Introducing JSX 12](#_Toc142218349)

[12. How React Works 12](#_Toc142218350)

[13. Building the First Custom Component 12](#_Toc142218351)

[14. Writing More Complex JSX code 12](#_Toc142218352)

[15. Adding Basic CSS Styling 12](#_Toc142218353)

[16. Outputting Dynamic Data and Working with Expressions in JSX 13](#_Toc142218354)

[17. Passing data via props 13](#_Toc142218355)

[Section 4: React State & Working with Events 14](#_Toc142218356)

[36. Module Introduction 14](#_Toc142218357)

[37. Listening to Events and Working with Event Handlers. 14](#_Toc142218358)

[38. How Components Functions Are Executed 14](#_Toc142218359)

[39. Working with Stats 14](#_Toc142218360)

[40. A Closer Look At “useState” hook 16](#_Toc142218361)

[41. States can Be updated in many ways 16](#_Toc142218362)

[42. Adding Form Inputs 17](#_Toc142218363)

[43. Listening to User Inputs 17](#_Toc142218364)

[44. Working With Multiple States 17](#_Toc142218365)

[45. Using One State Instead 17](#_Toc142218366)

[46. Updating State That Depend on Previous State 17](#_Toc142218367)

[47. Handing Form Submissions 17](#_Toc142218368)

[48. Adding Two-Way Binding 17](#_Toc142218369)

[49. Child-to-Parent Component Communication (Bottom-Up) 17](#_Toc142218370)

[50. Lifting The State Up 17](#_Toc142218371)

[51. 17](#_Toc142218372)

[Appendix 18](#_Toc142218373)

[TO DO 18](#_Toc142218374)

# Reference

|  |  |
| --- | --- |
| **Reference** | **URL** |
| Tutorial | [React Documentation](https://legacy.reactjs.org/docs/getting-started.html) |
| Udemy Course | [React - The Complete Guide 2023 (incl. React Router & Redux)](https://www.udemy.com/course/react-the-complete-guide-incl-redux/?kw=React+the+complete&src=sac) |
| [Udemy GitHub](https://github.com/academind/react-complete-guide-code) |
| Coursera | [meta-front-end-developer](https://www.coursera.org/professional-certificates/meta-front-end-developer) |

# ReactJS My GitHub

|  |  |
| --- | --- |
| **Item** | **URL** |
| GitHub URL | <https://github.com/Mansoor-R/ReactJs.git> |
|  |  |
|  |  |

# Section 1: Getting Started

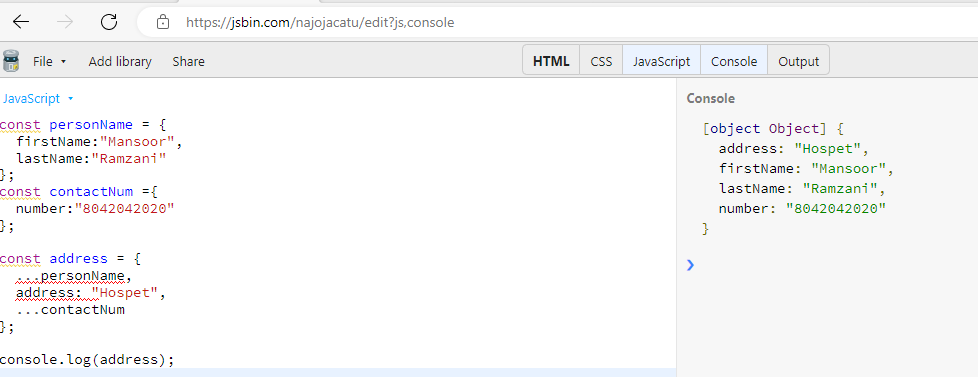
### About ReactJS

### 

# Section 2: JavaScript Refresher

Resource: <https://github.com/academind/react-complete-guide-code/tree/03-react-basics-working-with-components>

### The Spread and Rest Operator



### De structuring

Easily extract array elements or object properties and store them on variables.

|  |
| --- |
| **Array De-structuring** |
| [a, b] = [‘Hello’, ’Mansoor’] |
| Console.log(a) // Hello  Console.log(b) // Mansoor |
|  |

|  |
| --- |
| **Object De-structuring** |
| {name} = [name: ‘Mansoor’, age: 28] |
| Console.log(name) // Mansoor  Console.log(age) // undefined |

### References and Primitive Types

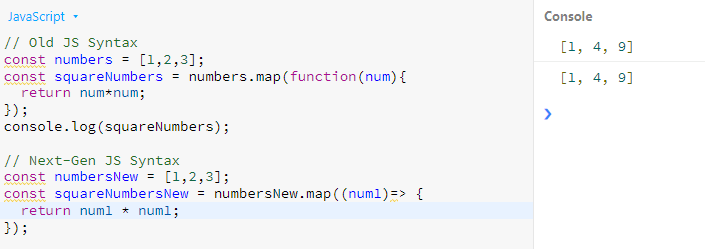
Graphical user interface, text

Description automatically generated with medium confidence

Text

Description automatically generated

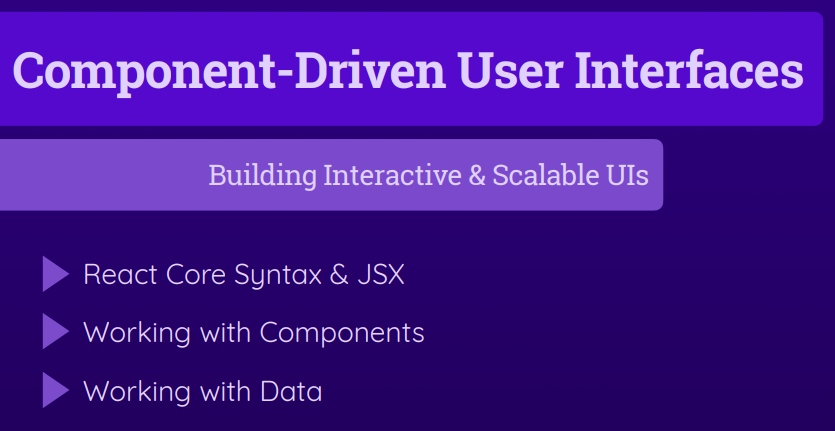
### Array Function



# Section 3: React Basic & Working with Components

### Module Introduction

We are going to learn following concepts along with examples.



### What are Components & why React is all about them.

React makes building complex, interactive and reactive user interface simpler.

Components are reusable building blocks of HTML, CSS, or JavaScript. Components allow reusability and separation of Concerns.





A blue symbol with yellow text

Description automatically generated

A blue and yellow symbol with a circle

Description automatically generated

A blue and yellow symbol with a circle

Description automatically generated

A blue and yellow symbol with white text

Description automatically generated

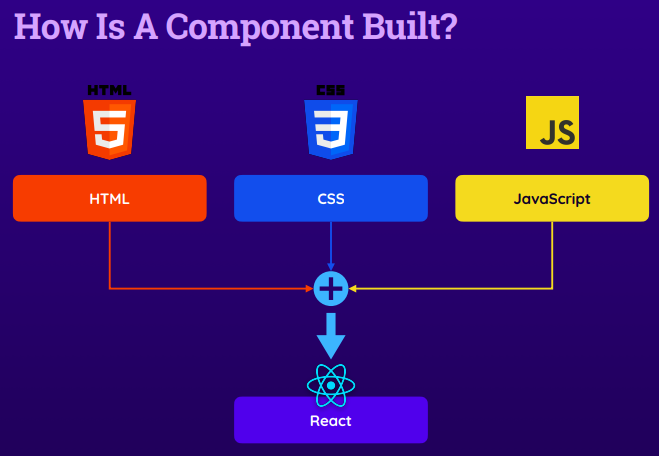
A screenshot of a computer

Description automatically generated

### React code is written in declarative way!

React allow you to create re-usable and interactive components consists of HTML, JavaScript (and CSS).

**Declarative** – Define all the desired state(s) and let React figure out the actual JavaScript DOM instruction. (You declare HTML and let React render it on DOM unlike Vanilla JavaScript where you define DOM).



A screenshot of a computer

Description automatically generated

### Create new React project.

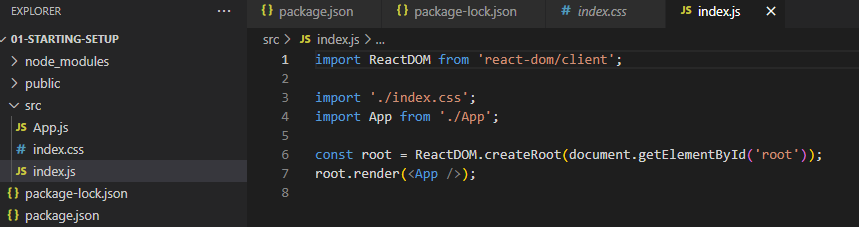
Create-react-app is used to create new React application this command requires NodeJS. So install NodeJS before creating the react project.

|  |
| --- |
| **Syntax**:  npx create-react-app my-app // Create React project  cd my-app // change directory  npm start // start the application on local development environment  URL: http:localhost/3000  CTRL + C to stop the local server |

Note: Create project name can’t contain capital letters.

### Analysing the standard React Project

Index.js is the first executable file here…



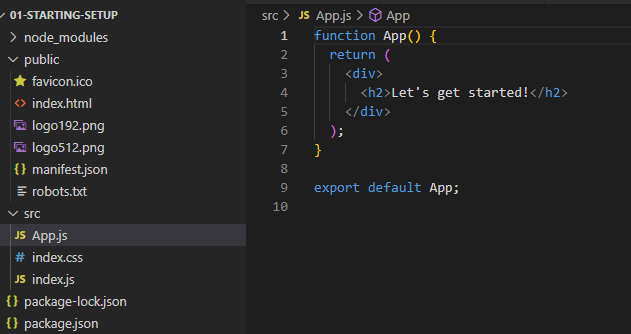
Index.html is the single html file in the project. Observe the div with id=”root” which does not hold any content but to this div react populate the UI. The same root div is selected in the react code above. On the root object the app is rendered.

Now what is that <App />. In the port we import app.js.

Text

Description automatically generated

In App.js function, a html code is returned.



### Introducing JSX

The code that we write is transformed behind the screen.

### How React Works

### Building the First Custom Component

1. Build one component in one file under components folder
2. File name must tell the logic, always start with capital letter and capital letter for each of the words without space.
3. Lower case compounds are basic html components and Uppercase (First charact and first chart of following words) components are custom components.
4. A component in React is just a JavaScript function. It returns JSX code.
5. In each component there must be only one root element.

### Writing More Complex JSX code

1. CSS files need to be imported using import command.
2. Key word class need to replace with className.
3. It is standard not to put complex JavaScript code in JSX rather write this code outside.

### Adding Basic CSS Styling

* CSS file need to be imported using import command.
* Keywork class is replaced with className.

### Outputting Dynamic Data and Working with Expressions in JSX

* {} can be used to evaluate JavaScript expression during compilation.
* Using $ for Variables: Sometimes, developers use the dollar sign as a prefix for variable names to indicate that the variable holds a reference to a DOM element or a result of a jQuery selection.



Figure : Dynamic Data & CSS class

### Passing data via props

## 

# Section 4: React State & Working with Events

## Module Introduction

In this section we learn about user interactions and handling user events. Also, most important concept called state.

## Listening to Events and Working with Event Handlers.

All the HTML controls can be added within JSX code, and all the events handlers are also can be used for these controls like onclick events etc.

Text

Description automatically generated

## How Components Functions Are Executed

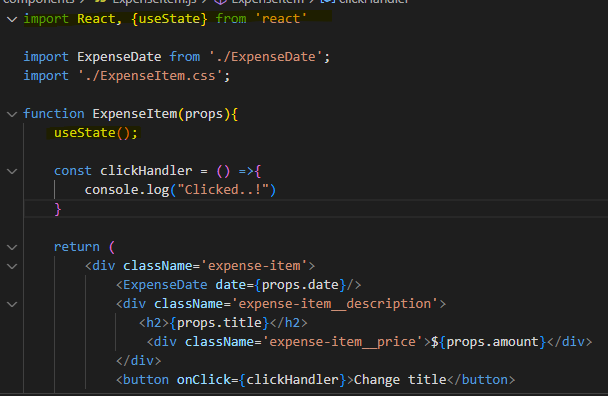
React executes all the components one after the other serials. If a parent has nested component, then all the child components are executed first before going to next components. One component is not called 2nd time as call back. To update the state of any components once it is loaded React uses State.

## Working with Stats

To use state import useState library from React.

* useState is a react hook.
* This should be called inside react component function. And also they should not be called inside any nested function.
* useState has as specific format to follow as shown in 2nd screen.

|  |
| --- |
| function ConstructorFunction(prop) {  const [title, setTitle] = useState(prop.title); // Set Default value  setTitle(‘Updated’); // set new value.  } |



UseState set the values of variable title when function setTitle executed. And setTitle is executed on button click event.

Text

Description automatically generated

## A Closer Look At “useState” hook

useState register some state for specific component instance.

Every Item of a component has its own registered state. Change in one state will not impact the other state.

Text

Description automatically generated

## States can Be updated in many ways

## Adding Form Inputs

## Listening to User Inputs

## Working With Multiple States

## Using One State Instead

## Updating State That Depend on Previous State

## Handing Form Submissions

## Adding Two-Way Binding

## Child-to-Parent Component Communication (Bottom-Up)

## Lifting The State Up

## 

# Appendix

|  |  |
| --- | --- |
| Js bin | [JS Bin - JS Bin](https://jsbin.com/najojacatu/edit?js,console) |
|  |  |

# TO DO

1. Dashboards
   1. Chart JS
   2. React integrate with Power BI

* By MS Power BI <https://www.youtube.com/watch?v=A5KFY5Jh1Uc>
* <https://github.com/amit-shuster/PowerBI-Developer-Videos/blob/master/Embedding%20Power%20BI%20in%20a%20React%20application/my-app/src/App.js>
* <https://www.npmjs.com/package/powerbi-client-react>

1. Calculator
2. Saving Plan