

React - Code with Harry

1)

→ Made and maintained by facebook. (meta)

→ Used for single page application

↳ [Application only loads once and rest of the things happen using JS.]

→ Based on components which can be used again & again.

→ Installations needed

→ Node.js (Runtime environment makes it possible to execute JS in your computer).
(without browser).

→ VS Code (as IDE).

→ npm installed with node.js.

→ React dev tools (Chrome extension)

Extensions which are useful } Thunderclient, EST React Snippets, HTML5 boiler plate, Live Server

lec 2

→ A basic folder structure quickly created using create react app.

→ `npx create-react-app app-name`.

`cd my-app`

`npm start`

* `npx` → leading something (without buying & returning).
VS
`npm` → buying & keeping something.
(It takes space & is installed).

→ Library (or) framework.

↳ Is technically a Library.

[but as it is in competition with angular & vue.js which are frameworks. it is usually also called framework.]

↳ Component based Architecture.

→ Files in create - React - App

↳ node_modules → All modules used in project.
can be replaced using package.json & package-lock.json.

↳ Readme & Gitignore used in github upload.

↳ public → only has index.html (which has root dir).

↳ src → has all components (most imp).
class based function based
 States
 Props

Lec 3 (Javascript Refresher)

①

→ language of the web. → 90% of sites worldwide.

Initially made for client side functionality.
but evolved into server side also.

②

→ console.log(a);
var, let, const a;

Arithmetic operations.

→ confirm & alert boxes.

③

→ numbers, obj, boolean, string & data types.
undefined, null

④

→ functions. → function funcName() {

* obj → let a = [

 x: 34;

 m: 64;

 func: ...

]

 | console.log(a);

 Prints everything.

 | console.log(a.x);

 prints x.

 | a.func()

⑤ Event listeners in JS.

Note → many JS string methods.

→ document.addEventListener(...);

 → document.getElementById(...);

 etc.

⑥ Data Structures and Algorithms.

→ Basic methods like swap, reverse, etc.

→ map, filter & reduce

 for, for each & while loop.

 if, else, switch.

→ Date.

⑦ Arrow function

→ lambda func.

→ use strict "strict mode on.

 → this keyword.

⑧ Callback fn → Invoked when something is complete. (work)

setTimeout, etc.

Asynchronous JS. → JS doesn't execute synchro, i.e. if the next fn not ready is called next to next fn. (i.e. not Synchronous).

⑨ Javascript Promises → Promises in when a certain task is complete I will run this fn.

→ Promises

 └─ Resolved.

 └─ Rejected.