L-4

Talking is Cheap
Show me Code!!

Page No.
Date

Precab

Babel - ~~it is~~ Compiler.

Normal variable - small letter
L React Element

functional/Component - Capital letter

→ return → use → multiple lines of code
L single line of code written in multiple lines - no need to use return
ex -

```
<h1 id='title'>
Food
</h1>
```

} no need to use return

→ Comment in JSX
L like in JS { } inside this

→ React never makes mandatory to use ES6, JSX or Typescript

→ we can seporatel use React for any particular section of web ex- footer

NOTE - Third way to write component
<Title></Title> ~~Not gone~~

Advice → First plan & then Do coding
L(Not Everything)

3hrs:15 backup at 94%

→ JSX only have one parent

9:14 / 1:16 time

JSX Component need one parent
for more than one tag

JSX need parent so but what if
we don't give extra parent

then → React fragment is component
which exported by React

n:11
4:11

```
<React.Fragment>
 // your code
</React.Fragment>
```

not provide any ~~extra~~ parent
~~this is power~~ of Reactfragment

React fragment is like empty tag
So, it look ugly to write length <React.fragment>
So, Shorthand <> </>

```
<React fragment>    ⇒   < >
</React fragment    ⇒   </ >
```

Do not ~~need to~~ provide parent

→ Inline CSS in JSX

```
<div style={ { background="blue", color="red";
                                          }
</div            }
```

we have to provide object

[illegible] this

3 | ss-color

Const StyobJ = {

backgro : "red" ;

}.

~~& . In JSX~~ In JSX

<div style={ StylobJ }

</div>

Q Can I use React fragment Inside React fragment

→

—

→ . ~~Joint, y~~ to provide Comm

v.v.I — Config Driven UI

find optional chaning

1:57

5tb

—

Props :- properties

- passing attribute
- passing datas properies in to component

basic — we pass argument & recieve parameter

- Props is normal parameter function

- Probs → not mandatory to use but conventional

Code

Probs — ~~reciving~~ parameter

```
Const List =(Probs)=> (
   <div>
   <H1> This {Probs. objec} </H1>
   <H1> This { Probs. obj} </H1>
   </div>
   )
```

```
<Body>
   </List menu={obj[0]}/>
   </List menu={obj[1]}/>
   </List men= {obj[2]}/>
</Body>
```

These all here passing argument

2nd way

we can also write
{List (obj[0]) }

We can also pass multible probs like

```
</List mono={obj[0]} name="Ankit"/>
```

same thing by 2nd - way

Comonly ~~we~~ develobers use probs word of parameter

but we can also do destructuring

~~Not~~
Probs
↓
destructing
↓
{Aynone}

ex -

~~Cons~~

```
Const List = ({ Anynone
                ex- item }) => (
    <H1> This { item } </H1>
)
```

H/w Read about destructing -

```
const probs = {
    restront : { datafr
           data { nome
                  age
                  tin } } } }

const { restront } = probs

Const restront = {
  data.dat
    nome : ''     ''
    age  : ''     ''
    tie  : ''     ''
}

→ const {nome, age, time} = restront.data.data
```

Q:15 deep dive in to destoutsing of Q:15 L-4

gn real Abbrication

map is best way to do functional brogromming

Q: 93

— what is differens b/w map & forEach

virtual Dom

its a representation of dom ~~in code~~ in our code.

we keep respraction of dom with us

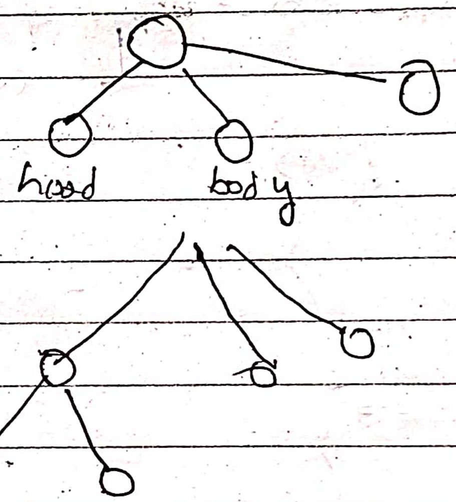
why need it

- we need it for reconciliation
  search

Dom

<head>
<body>
~~<[illegible]~~
<nest>
<nesq>
<bodyl>
</head>

Virtual-Dom



Reconciliation :- it is an algorithm uses by react to find diff from one tree to another tree and it determining what ~~[illegible]~~ changes or not.

- ~~ff find~~

it find out differnce b/w Dom & virtual dom & changes accordingly. & it just render that part only not whole page.

Conclusion:- it is only re-render that position of app which found by diff

This process is called Reconsidliation. use div algorithm.

key ~~if their is~~
Let assume there is

1st case

<div> <div> <div> <div>
① ③ ② ④

└ So here is 4 div with same attribut
~~rea~~ if the position of ~~react ch~~ div changes then React is unable to know it.

2nd case

but ~~if~~ when there is <div> & <img> only & its poisition is changes as:-

<div> </div>
<img>

after changing position
<img>
<div> </div>

but this time react know this which & what had ~~be~~ changed. ~~So~~ & it is identified by react as it has different tag name So, by its tag name react knows it

more ex -

1 2 3 4 5 | 6 1 2 3 4 5

~~render everything~~ | re-render everything

but if key is given

~~then it re-render as it know what & which d is part is changes~~