**# Working with props : -**

When using class based component, it’s **this.props**

**e.g** class person extends component{

render(){

return(

<h1>my name is {this.props}</h1>

)

}}

**# Understanding children property : -**

* **App.js:-**

render(){

return(

<div>

<Person name=”ABC” age=”21”>

<Person name=”XYZ” age=”21”>

<Person name=”cBC” age=”21”> ABC is my name</Person>

</div>

)

}

* **Person.js :-**

Render(){

return(

<p>my name is {props.name} and age is {props.age}</p>

<p>{props.children}</p>

)

}

**#Handling events with methods :-**

eventHandler = () =>{

console.log(“clicked”);

}

Render(){

Return(

<button onClick={this.eventHandler}>Click</button> **right method**

<button onClick={this.eventHandler()}>Click</button> **wrong method**

this will executes when react render method is executed.

**#Manipulating State :-**

Component method has **setsState({})**  which allows us to update state property take object as an argument

**#Using the useState() Hook for state Manipulation :-**

* **useState :-**

It is a hook that allows to manage state in a functional component

Syntax :-  **const Statearr = useState(**

Person:[{name:’Max’ ,age:’10’}

{name:’Manu’ ,age:’12’}

],

Otherstate:’some other value’

**)**

It returns array with exactly to elements