

Javascript !!!!!

```
F1(num1,num2 )
{
    //We want to use num1 and num2

    var p = num1 * num2 //Line x
    var p1 = num1 + num2 //Line y
}

//different ways of calling
F1("hi", "hello") --- error on line x , concat will happen on liney
F1("12","34" ) ---- extract by parseInt( JS attempts to complete operation) and multiply ,
concat will happen on liney
F1(12,34 ) ---- multiply , addition
```

Tag Opening	Tag Body	Tag Closing
<mytag name="pp" age={12} >	children	</mytag>

props.name , props.age	Accessing attributes of the tag	
props.children	Accessing body of the tag	

Ex1 --- Write a react component that takes some content in the Tag body
Output ---- show the content along with number of chars in the tag body

```
<CharCount> This is a wonderful morning </CharCount>
```

```
||
||
||
```

Output Text : This is a wonderful morning
 chars : 27

WE also added a CSS file and the class on div using JSXAttribute **className** (camel case)

Event Handling in React !! Events are similar to Javascript event , JSX Event name is in CAMEL CASE

onclick -----> onClick
onchange -----> onChange

```
<button onclick="validate()" /> -----> <button onClick={validate} />
```

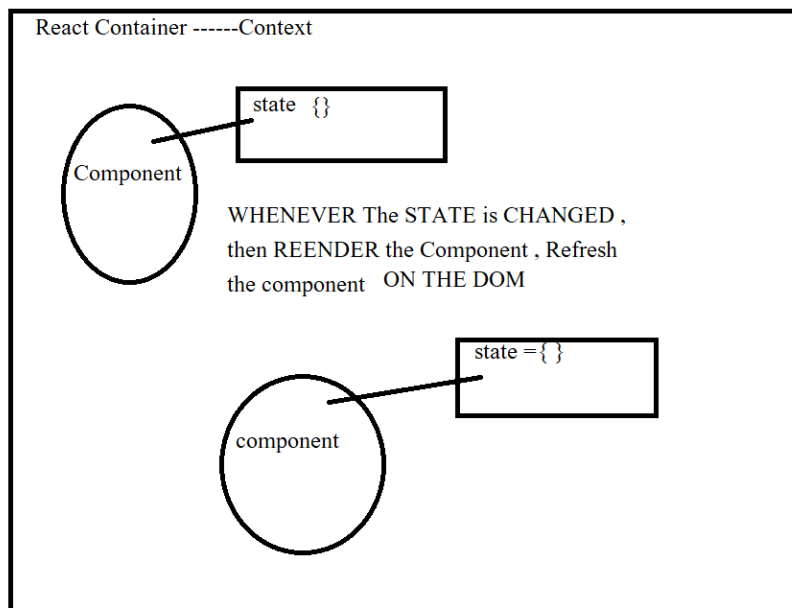
Ex2 --- show an alert when a button is clicked .

1. We never call a function in the `onClick = { }` , we always register a callback . It could be a function name or it could be arrow function .
2. Whenever the event occurs on the target (html element on the screen) , the **target info** is sent as **event** object to the callback method.
3. This event object can tell us about the target on which the event occurred using `event.target.value` !!!

Ex 3 ---- user enters a number in the input field , show the number in a `` tag below .

React context REFRESHES the DOM partially !!!

Only the changes STATE is refreshed !!! Changed local variables or changed properties are not refreshed on the DOM !!!!!



property1

```
setProperty(newval)
{
  property1=newval
  CODE TO RERENDER THE COMPONENT IS EXECUTED
}
```

In the class component

The **setState** method modifies the state object and re-renders the component

The property name MUST be `state` (accessed as `this.state`)

The setter method MUST be `setState({ })`

Create a STATE variable num with initial value 0!!

When we click a button INCR the value of num should be incremented and displayed in a span !!

Ex3 ---

write a class component that accepts a name from the input field and shows the name in Upper case after clicking on the button !!!

1. in a span
2. in the text field itself

TF : _____ button

UPPERCASE

```
class UpperEx extends React.Component
{
  state={inputdata:"",uc:''}

  dosomething= ()=>{

    Let v = this.state.inputdata
    Let uk = v.toUpperCase()
    this.setState({uc:uk})
  }

  Render()
  {
    Return (
      <div>
        <input type="text" onBlur={(e)=>{ this.setState({inputdata:e.target.value}) }} />
        <button onClick={this.dosomething}>OK</button>
        <span>{this.state.uc}</span>
      </div>
    )
  }
}
```

```
function UpperExF()
{
  const [inputdata,setInputData] = useState("")
  const [uc,setUC] = useState("")

  func()
  {
    Var uk = inputdata.toUpperCase()
    setUC(uk)
  }

  Return(
    <div>
      <input type="text" onBlur={(e)=>{setInputData(e.target.value)}} />
    </div>
  )
}
```

```
<button onClick={func}>OK</button>
<span>{uc}</span>
</div>

)

}
```