- useState hook with Objects -

=> we can't directly update our variable directly so we need to use
-- useState hook because we are using the functional components
(for class based components we use setState hook)

-- if you want to print the brand and the model name then you can make two different states like this and each of them can be handled separately.

```
function Car() {
    const [brand, setBrand] = useState("Ford");
    const [model, setModel] = useState("Mustang");
```

-- but here we have only two variables what if we have multiple variable then for each of them you can't make just state variables like this individually .
-- so we need of objects .

-- like if you are creating the forms then there are multiple values :-

```
App.js > default
import { useState } from "react";
import React from "react";

function Car() {
    const [car, setCar] = useState({
        brand: "Ford",
        model: "Mustang",
    });
    return (
        <>
        <h1>It is a {car.model} of {car.brand}</h1>
        </>
    )
}
export default Car;
```

It is a Mustang of Ford.

Note : updating values of the Object is different like you can't just use setCar and just update the value
Ex : if you want to update the model of the brand like Fiesta
-- so you wanna do ford fiesta
-- then you will do like this :

```
function update(e){
    e.preventDefault();
    setCar({model : "Fiesta"})
}
```

It is a Fiesta of

Update

-- so if you try to change only the model then it's first name automatically get removed .
-- because when you are setting the value then only model is there and no brand is available there .
-- so it will discard the brand property from the object .

-- similarly if you will change the model then brand will be discarded.

Note : so we need the ...spread operator.

Spread Operator

The JavaScript spread operator (`...`) allows us to quickly copy all or part of an existing array or object into another array or object.

-- means we have all (or part of the existing Object/array) value and add some new if you want to add .

```
function update(e){
    e.preventDefault();
    setCar({...car, model : "Fiesta"})
    // spread operator
}
```

-- and finally it is working because it doesnot discard the existing values .

-- spread operator first creates copy of all the existing values and then update (overwrite with the new value if provided).