

Day 38

Array destructuring:

It is a way to get individual items from an array of items and save those individual items as separate component.

Object destructuring:

When structuring object we have to destructure a property of an object using that exact properties name as the name of the destructured variable.

Working with useState

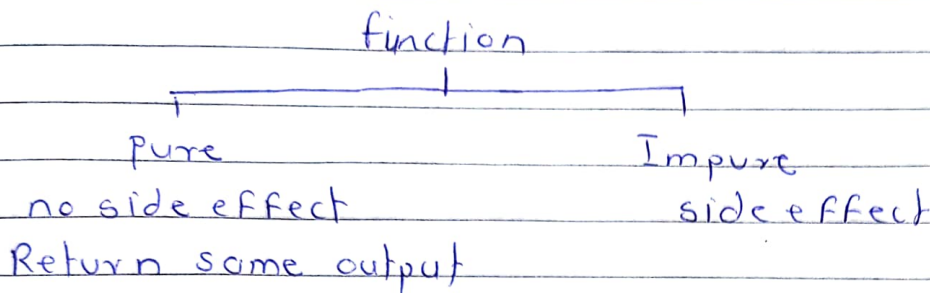
eg

```
import { useState } from "react";
```

```
export default function App() {  
  const [greeting, setGreeting] = useState({  
    greet: "Hello, World" });  
  console.log (greeting, setGreeting);
```

```
  function updateGreeting() {  
    greeting.greet = "Hello, World-wide web";  
    setGreeting (greeting);  
  }
```

Side effect - Makes a function impure



Pure function example

```

function Dob(props) {
  return <h1> Dob : { props.year } </h1>
}

function App() {
  return <Dob = {2001} />
}

export default App;
  
```

Impure function example

```

function ShoppingCart(props) {
  const total = props.count * props.price;
  console.log(total)
  return <h1> Total: {total} </h1>
}

export default function App() {
  return (
    <shoppingCart items = {5} priceperItem = {10} />
  )
}
  
```

useEffect hook - Use to deal side effects

eg

```
import {useEffect} from "react";
```

```
function shoppingCart (props) {
  const total = props.count * props.price;
```

```
  useEffect (function () {
    console.log (total);
```

```
  }, [])
```

```
  return <h1> Total : {total} </h1>
```

```
  }
```

using arrow function -

```
function shoppingCart (props) {
  const total = props.count * props.price;
```

```
  useEffect (() => console.log (total), []);
```

```
  return <h1> Total : {total} </h1>
```

```
  }
```