To **export multiple functional components**, we do either of the below 2 :-

```
export { Title, Header };
```

```
export const Title = () => (
    <a href="/">
        <img
            className="logo"
            alt="logo"
            src="https://yt3.
        </img>
    </a>
);

export const Header = () => (
    <div className="header">
```

However, during importing them, we cannot just say :-

import Obj from "./components/Title";

Because as I said earlier, during named exports, we are not exporting components as objects, so we do any one of the below 2:-

```
import * as Obj from "./components/Title";
```

and use it like :- <Obj.Header />

```
import { Title, Header } from "./components/Title";
```

and use it like :- <Header />

#NOTE :-

We can also export by default and name in the same file, and import them in the same line BUT the default import should be placed before the named import.

```
export default Header;
export { Title };
```

```
import Header, { Title } from "./components/Title";
```

ONE-WAY DATA BINDING IN REACT

*
```
Const Body = () => {
  const SearchTxt = "KFC";
  return (
    <>
    <div>
    <input type="text"
           placeholder="Search"
           value={searchTxt}
    </>      />
    );};
```

I have a variable 'SearchTxt' and if I put that here, Then, the value "KFC" will go inside my input box.

→ I'm not able to edit the value "KFC" because it is a hardcoded value.

→ To change the value in the input box, we need to modify the variable SearchTxt.

But in input box, If we write something, it won't change SearchTxt.

This is called One-way data binding.

* How will I change the value of SearchTxt ?

Ans) :- Write an onChange method

→ In react, If I want to create a local variable like SearchTxt, I will create it using useState Hook

useState Hook

→ New way of creating variables

```
Const [searchTxt] = useState();
```

If we have to create local variables in react, you need to use state variables.

→ State variables are created using useState hook

What is Hooks ?

- Hooks are normal functions.
- I get useState hook from 'react' library. (imported using named import)

What is the function of useState ?

- It's to create state variables.

```
const [SearchTxt] = useState();
```

(Second element is a set function to update the variable)

This function returns an array

The first element of this array is variable name.

'SearchTxt' is a local state variable.

→ To give a default value to my useState variable, do this ↓

```
Const [searchTxt] = useState("KFC");
```

→ This is how we create a local state variable in react.

> In javascript, we create a variable searchTxt having a value "KFC" like this ↓
>
> ```
> const searchTxt = "KFC";
> ```

→ In react, to modify the variable 'searchTxt' I have to use function.

useState() gives us that function.

Let us call that function 'setSearchTxt()'

```
const [searchTxt, setSearchTxt] = useState("KFC");
```

from this event property I can read whatever I'm typing

```
onChange = {(e) => {
    setSearchTxt(e.target.value);
}}
```

```
* const Body = () => {
    const [searchTxt, setSearchTxt] = useState("KFC");
    return (
      <>
        <div classname = "search-container">
        <input type = "text"
               className = "search-input"
               placeholder = "Search"
               value = {searchTxt}
               onchange = {(e) => {
                  setSearchTxt(e.target.value);
               }}
        />
        <button className = "search-btn">
          Search - {searchTxt}
        </button>
      </>
    );};
```

TWO-WAY BINDING

Here, I'm reading as well as writing searchTxt. Both