

Types of exporting the code in reactjs ?

ANS:

In React.js, there are primarily two types of exports: **named exports** and **default exports**. Here's a brief overview of each:

Default exports: basically the purpose of exporting the code is to use it in another file .

example

Creating a File with Default Export

File: `Greeting.js`

```
javascript Copy code  
  
// Define a function  
function Greeting() {  
  return "Hello, World!";  
}  
  
// Export this function as the default export  
export default Greeting;
```

export default Greeting; → this is the syntax to export the default export . if you want to export Use these keywords **export default** and mention what you want to export to use it somewhere

Using the Default Export in Another File

File: `App.js`

```
javascript Copy code  
  
// Import the default export from Greeting.js  
import WelcomeMessage from './Greeting';  
  
function App() {  
  return (  
    <div>  
      <p>{WelcomeMessage()}</p> {/* Use the imported function */}  
    </div>  
  );  
}  
  
export default App;
```

You can only have one default export per file. When importing, you don't need to use the same name as the exported item. You can choose any name you like.

Incorrect way of using default exports:

Example: Incorrect Use of Multiple Default Exports

Let's say you mistakenly try to have two default exports in the same file:

File: `Shapes.js`

```
javascript

// Define a function
function Circle() {
  return "This is a circle";
}

// Export Circle as the default export
export default Circle;

// Define another function
function Square() {
  return "This is a square";
}

// Trying to export Square as the default export too (This will cause an error)
export default Square;
```

This code will cause an error because you're trying to have two default exports (`Circle` and `Square`) in the same file.

Names exports in react:

Named exports allow you to export multiple things (like functions, variables, or classes) from a file. Each thing you export has a specific name, and you can export as many as you need. When you import them into another file, you must use the exact names you exported.

Key Points about Named Exports

1. **Multiple Exports:** You can export multiple items from a file. Each item has a specific name.
2. **Exact Naming:** When importing, you have to use the exact name of the item as it was exported.
3. **Curly Braces:** Named exports are imported using curly braces `{}`.

Example of Named Exports

Creating Named Exports

File: `mathUtils.js`

```
javascript Copy code  
  
// Export a function to add numbers  
export function add(a, b) {  
  return a + b;  
}  
  
// Export a function to subtract numbers  
export function subtract(a, b) {  
  return a - b;  
}  
  
// Export a constant value  
export const PI = 3.14159;
```

In this file, we have three named exports: `add`, `subtract`, and `PI`. Each one can be used individually in other files.

To import them to use in another file we have to use the exact names which are exported in the file. Otherwise it throws an error .

File: `App.js`

```
javascript Copy code

// Import the named exports using their exact names
import { add, subtract, PI } from './mathUtils';

function App() {
  return (
    <div>
      <p>Addition: {add(5, 3)}</p> {/* Calls the add function */}
      <p>Subtraction: {subtract(5, 3)}</p> {/* Calls the subtract function */}
      <p>PI Value: {PI}</p> {/* Uses the constant PI */}
    </div>
  );
}

export default App;
```

Mixing Named and Default Exports

You can have both named exports and a default export in the same file. But remember, you can only have one default export per file.

File: `shapes.js`

```
javascript Copy code

// Named exports
export function Square() {
  return "This is a square";
}

export function Triangle() {
  return "This is a triangle";
}

// Default export
export default function Circle() {
  return "This is a circle";
}
```

Using Mixed Exports

```
// Import the default export (Circle) - no need for curly braces
import Circle from './shapes';

// Import the named exports (Square and Triangle) - use curly braces
import { Square, Triangle } from './shapes';


function App() {
  return (
    <div>
      <p>{Circle()}</p> {/* Calls the Circle function */}
      <p>{Square()}</p> {/* Calls the Square function */}
      <p>{Triangle()}</p> {/* Calls the Triangle function */}
    </div>
  );
}

export default App;
```



When you have both a default export and named exports from the same file, you can import them together in a single line.

javascript

 Copy code

```
import Circle, { Square, Triangle } from './shapes';
```

Explanation:

- `Circle`: This is the default export from `shapes.js`. You import it without curly braces.
- `{ Square, Triangle }`: These are named exports from the same `shapes.js` file. You import them using curly braces.