

React - ES6 Variables

Topics : [React JS](#)

Written on [January 02, 2024](#)

In React, you can use ES6 (ECMAScript 2015) syntax for declaring variables. Here are some common ways to declare variables in React using ES6:

1. Using const and let:

- `const` is used to declare constants (variables that cannot be reassigned).
- `let` is used to declare variables that can be reassigned.

javascript code

```
const myConstant = 10;
let myVariable = 'Hello';

// Example of reassigning a variable
myVariable = 'World';
```

2. Destructuring Assignment:

- You can use destructuring assignment to extract values from objects or arrays.

javascript code

```
// Destructuring assignment with objects
const { name, age } = person;

// Destructuring assignment with arrays
const [first, second] = myArray;
```

3. Arrow Functions:

- Arrow functions are often used in React components.

javascript code

```
// Traditional function
function myFunction() {
```

```
// function body
}  
  
// Arrow function  
const myArrowFunction = () => {  
  // function body  
};
```

4. Template Literals:

- Template literals allow you to embed expressions inside string literals.

javascript code

```
const name = 'John';  
const greeting = `Hello, ${name}!`;
```

5. Spread Operator:

- The spread operator (...) can be used to copy elements from one array or object into another.

javascript code

```
const originalArray = [1, 2, 3];  
const newArray = [...originalArray, 4, 5];
```

6. Class Properties:

- When working with class components, you can use class properties to define state or other properties.

javascript code

```
class MyClass extends React.Component {  
  state = {  
    value: 42  
  };  
  
  // other class methods  
}
```

Remember that React components often use class components or functional components with hooks like `useState`. Here's an example of using `useState` in a functional component:

javascript code

```
import React, { useState } from 'react';
```

```
function MyComponent() {
  const [count, setCount] = useState(0);

  const increment = () => {
    setCount(count + 1);
  };

  return (
    <div>
      <p>Count: {count}</p>
      <button onClick={increment}>Increment</button>
    </div>
  );
}

export default MyComponent;
```

© Copyright **Aryatechno**. All Rights Reserved. Written tutorials and materials by [Aryatechno](#)