

🚩 Current Skill React Hooks: Overview

Introduction to React Hooks

The React **16.8** update was a major revolution in the way that React treats and interacts with components and this all thanks to hooks.

Hooks is making big waves in the React community because it reduces the complexity of state management.

In this skill, we are going to see:

- What are React Hooks?
- What is useState hook?
- What is useEffect hook?
- What is useRef hook?
- How to make your own custom hooks?

useState

While describing the use of state in the previous Super Skill, we have seen that the state can only be updated using the `this.setState` method.

The state is characterized by two major things: a **value** (aka **getter**) and a method to set this value (aka **setter**).

The same logic still stands with the `useState()` method.

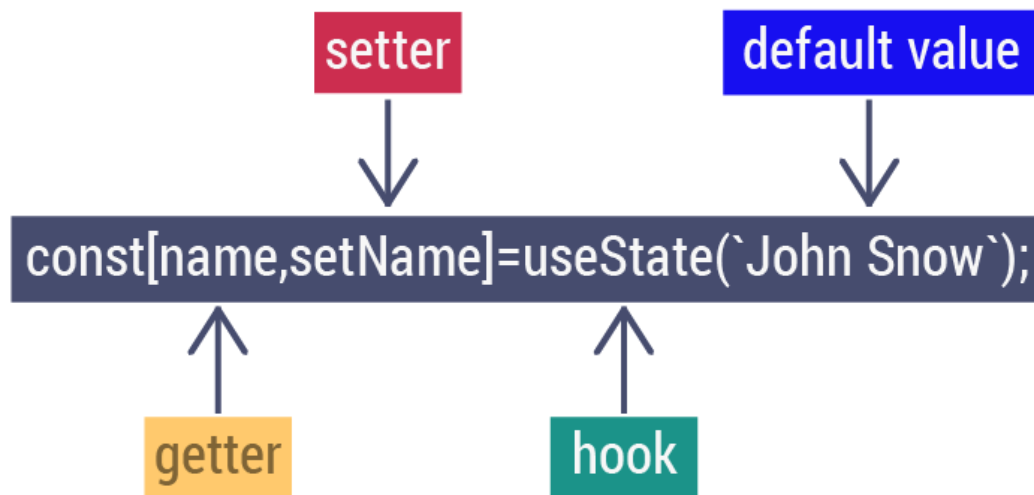
Declaring a state variable:

First of all, we need to learn how to declare a state variable in our component:

1. Import the hook `useState`.
2. Declare your state variable in an array (the getter and the setter)
3. Initialize your state variable with `useState()`.

This example will help you understand more:

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



Accessing The State

Now that we have already declared our component state, we need to be able to access the value of that state.

Like any other variable in JavaScript, if we want the value of a state, we invoke the state's name like it's shown in the example below:

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

const MyFunctionComponent = props => {
  // setting the state hooks
  const [name, setName] = useState("Arya Stark");
  return (
    <div>
      {/* here we use the getter to get the state value */}
      <p>hello my name is {name}</p>
    </div>
  );
};
```

Set the state's value:

To change the state's value, we have to use the setter that we implemented in the state declaration. In this example, the setName method acts as the setter (or the modifier):

```

const MyFunctionComponent = props => {

  // declaring the state hooks

  const [name, setName] = useState("Arya Stark");

  // here we use the setter to change the content of the name state

  const handleClick = () => setName('Tyron Lanyster')

  return (

    <div>

      {/* here we use the getter to get the state value */}

      <p>hello my name is {name}</p>

      <button onClick={handleClick}>Click if you want to give the crown to Tyron Lany:

    </div>

  );
};

```

Keypoints to keep in mind:

Now that we know what the useState() Hook does and once we have our component initial state, we need to be able to access the value of that state.

Like any other variable in JavaScript, if we want the value of a state, all we have to do is invoke the state's name, like shown in the example below:



```

import React, { useState } from "react";

function Welcome() {

  const [name, setName] = useState("Jane")

  return (

    <div>

      <h1> Welcome {name} </h1>

    </div>

  )

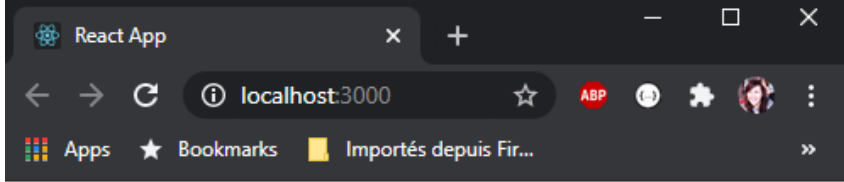
}

export default Welcome;

```




Output:



Welcome Jane



< Previous

next >

