

```

1 import { useState } from 'react';
2 const ToggleExample = () => {
3
4   const [show, setShow] = useState(false)
5   return (
6     <div>
7       <h4>toggle custom hook</h4>
8       <button className='btn' onClick={()=>setShow(!show)}>
9         toggle
10      </button>
11      {show && <h4>some stuff</h4>}
12    </div>
13  );
14 };
15 export default ToggleExample;
16

```

The above code has a toggling functionality. Every time we clicked the button the browser toggles “Some Stuff”.

If we have the same functionality(toggling) in various components in our application, we can create our own hook, let's say useToggle(in above case), and use it every-time we have the toggling functionality in different components.

```

03-advanced-react - useToggle.js
1 import { useState } from "react";
2
3 const useToggle = (defaultValue) =>{
4   const [show, setShow] = useState(defaultValue);
5
6   const toggle = () =>{
7     setShow(!show);
8   }
9
10  return {show, toggle}
11
12 }
13
14 export default useToggle;

```

In useToggle.js, we are creating useToggle hook that returns an object with show and toggle properties. (Note: we can return array as well (pretty much everything)). We will supply true as the default value when we call the useToggle hook. show property will hold the default value (in this case, either true or false) and the toggle property will hold the toggle functionality. Now let's import it in our ToggleExample component.

```
03-advanced-react - 01-toggle.jsx

1 import { useState } from 'react';
2 import useToggle from './useToggle';
3
4 const ToggleExample = () => {
5   const {show, toggle} = useToggle(true);
6   return (
7     <div>
8       <h4>toggle custom hook</h4>
9       <button className='btn' onClick={toggle}>
10         toggle
11       </button>
12       {show && <h4>some stuff</h4>}
13     </div>
14   );
15 };
16 export default ToggleExample;
17
```

We can now use the useToggle hook in various components in our application.