React.useSatate()

React 16.8 and onward, it enables developers to add stateful logic to functional component .useState is a hook that lets us add state to our functional component. It returns a array with two elements

* the current state value
* a function that allows us to update the state value

import React, {useState} from "react"

const feature = () => {

// declare state in functional component

const [state, setState] = useState(initialState)

  return (

    <div>feature</div>

  )

}

export default feature

**state**

The current state value

**setState**

a function to update the state value.

**initialState**

the initial state value, which can be any type

Problem before hooks

Before hooks, state management in react was exclusively handled by the class component. This posed several issue

* boilerplate code

Class component require more boilerplate code, such as constructor and method bindings

* component complexity
* reusability

Step by step execution process

Component initialization

When a function component with a useState hook is rendered for the first

Component definition

The component function is invoked. During this invocation, the useState hook is called

useState hook call

initial state setup

when useState is called for the first, it sets up the initial state. useState return array with two elements

the current state value

a function to update the state

virtual dom creation

virtual dom

react creates a virtual dom representation of the component using the current state value

reconciliation

react compares the virtual dom to actual dom and updates the actual dom to reflect the virtual dom

component rendering

the component render using the initial state

state update

when the user interacts the component setState is called.

Rerendering

After state update

Reinvoking the component

React reinvoke the component function with the new state value.

Recoumputing the virtual dom

The virtual dom is recreated based on the new state

Reconciliation

React compares the new virtual dom with the previous virtual dom and updates the actual dom to reflect any changes

Internal mechanism of useState