useState

State as Object:

The state is an object that holds information about a certain component.

Two Factor:

State change is one of the two things that make a React component re-render(the other is a change in props)

Behavior

The state stores information about the component and controls its behavior.

Updater Function

The setState() function is asynchronous .

Time Taken by Re-render

The time it takes for a component to render after a call to setState() depends upon the various factors, including the size and complexity of the component, the speed of the Computer or device and the amount of data being processed .

Optimization

It reduces unnecessary re-renders and updates only the components that have changed.

useEffect

Side Effects

Actions which are Unpredictable because they are performed with the outside world.

Meaning

Side Effects basically means something that happens after Some other specific thing happens.

Common Side Effects

useEffect (cont)

1. Making a request to an API for data on back end server.

2.Interact with browser API e.g. use document or windows directly.

3.Using unpredictable timing functions like setTimeOut() or setInterval().

How

If we need to perform side effects it should be after the component renders.

useEffect

useEffect has one function(callback) and an array(dependency array)

Callback

Call back function will be called after the component renders. And this is the function which is used to perform our side effects .

Dependency Array

The dependencies array should include all of the values that our side Effect relies upon.

1. The array will cause the useState() to execute again if one of its variables changes

2.If provided incorrectly the array with the local state. this may cause an infinite loop.

CleanUp Function

When component unmounts this functions cleans up any changes made by effect.

useEffect and useState

```
import React, { useState,
useEffect } from 'react';
function Example() {
    const [data, setData] =
useSta te([]);
    use Eff ect(() => {
        fet ch( 'ht tps :// -
jso npl ace hol der.ty pic -
ode.co m/p osts')
        .th en( res ponse =>
respon se.j son())
        .th en(json =>
setDat a(j son))
        .ca tch (error =>
consol e.e rro r(e rror));
    }, []);
    return (
        <di v>
        {da ta.m ap (post =>
(
            <div key={p -
ost.id }>
            <h2 >{p -
ost.ti tle }</ h2>
            <p> {po -
st.b od y}< /p>
            </d iv>
        ))}
        </d iv>
    );
}
```