

```

1  /* ===== Effects, Reducers & Context =====
2
3  What is an "effect" (or a "Side Effect")?
4
5  Main job: Render UI & React to User Input
6  ==>
7      Evaluate & Render JSX
8      Manage State & Props
9      React to (User) Events & Input
10     Re-evaluate Component upon State & props changes
11     .....
12     This all is "baked into" React via the "tools"
13     and features covered in this course (i.e.
14         useState() Hook, Props etc);
15  ==<
16
17  Side Effects: Anythings Else
18  ==>
19     store Data in Browser Storage Send Http Requests to Backed Servers Set & Manage Times
20
21     These tasks must happen outside of the normal compnent evaluation and render cycle especially
22     since they might block/ delay rendering (e.g. Http requests)
23  ==<
24
25  Handling Side Effects with the useEffect() Hook
26
27  useEffect( () => {...}, [ dependencies] );
28
29  ***> A function that sould be executed AFTER every component evaluation IF
30  the speciefied dependencies changed
31
32  ***> Dependencies of this effect the function only runs if the dependencies changed
33
34  ===== can be save time when send HTTP Req =====
35  useEffect( )=>{
36      //setTimeout( () => {}, 500);
37      const identifier = setTimeout( () =>{
38          setFormIsValid(
39              // HTTP requests
40              enteredEmail.include('@') && enteredPassword.trim().length > 6;
41          );
42      }, 500);
43
44      return () => {
45          clearTimeout(identifier);
46      };
47  }, [enteredEmail, enteredPassword]);
48
49
50  // ***** Another Example
51
52  useEffect( )=>{
53      ... All time run this when component Render coz without Dependencey

```

```

54     });
55
56     //***** with Depenedece empty [] run only one times *****
57     useEffect( ()=>{
58         ... run only first times when component render
59     }, [])
60
61
62     // ***** with dependencies [enteredPasswaord] *****
63     useEffect( ()=>{
64         ...whit out function it will run All keyStory
65
66     }, [enteredPassword]);
67
68     // ***** with function and dependecy *****
69     useEffect( ()=>{
70         ... function run only the dependencies changes
71         ... I mean return function
72         return ()=>{
73             ... every change of dependencies it will run...
74         };
75     }, [eneteredPassword]);
76
77     */
78     /* ===== useEffect =====
79         // Dev tools => Application => localStorage
80     import React, { useState, useEffect } from 'react';
81
82     const [isLoggedIn, setIsLoggedIn] = useState(false);
83
84
85
86
87     //useEffect( ()=>{ }, []);
88     useEffect(() => {
89         let storUserLoggedInInfo = localStorage.getItem('isLoggedIn');
90
91         if (storUserLoggedInInfo === '1') {
92             setIsLoggedIn(true);
93         }
94     }, []);
95
96     localStorage.setItem('isLoggedIn', '1');
97     localStorage.removeItem('isLoggedIn');
98
99     // Now side effect can change or update When any keyStory is update
100
101     useEffect( ()=> {
102         ... Inside code run only the dependecy change any time
103
104         setFormIsValid(
105             enteredMail.includes('@') && enteredPassword.trim().length > 6
106         );
107     }, [enteredMail, enteredPassword] );
108
109

```

