Use Effect

```
UseEffect(() => {
    console.log("Use effect");
}, []);
        ↑
    dependancy array
```

1) If no dependancy array use effect is called everytime

2) If empty dependancy array then if is only called once at start.

3) If some thing given inside the array then it will be called ~~when~~ only when value inside that array changes.

Use state ← used to creating local state variables inside our
                    functional component.
  Initialize
1) Never make use state out side the function.

2) Never create it inside a conditional statement. or loops or any other place keep them at top and singal

www.reactrouter.com   date- 15/3/24

React Router Library

① npm : react-router dom

or ~~pnpm~~ yarn add react-router-dom

② Make Routing configuration in root file   [in app.js]
        ↖ Info that will tell ~~the~~ browser what will happen
           on that specific route

③ import {createBrowserRoute} from "react-router-dom"

④   const appRouter = createBrowserRoute([←
                                    ①   ↙ this take a
                                                     list
        {
            path : "/",
            element : <AppLayout/>,
        },

        {
            path : "/about",   ← Path then
            element : <About/>,   load this
        }
    ]);

⑤ import { RouterProvider} from "react-router-dom";

⑥ [root.render(< RouterProvider router={appRouter} />];

⑥ Created a contact us page

when there is an error.

→
```
{
    path : "/",
    element : < Applayout />,
    error Element : < Error />  ← our error
}            ↑                      component
          when
      the above
      components
      throws an
      error.
```
w

do import Export error.

Special Hook from Router Dom

→
→ in your error component Page ②
→ import { use Route Error } from "react-Router-dom

we get
a object
from this
You can see it
by

```
const Error = () => {
    const err = useRouteError();
    return (
        <div>
            <h1> Oops !! </h1>
            <h2> Something went wrong </h2>
            <h3>
                {err.status} : {err.statusText}
            </h3>
        </div>
    );
};
```

when ever a function
starts with use
it is a hook

export default Error;

Children Routes

Children Routes

import { createbrowserRoute, use RouterEnrichy, Outlet} from React-Router-dom

```
const Applayout = () => {,
    return (
        <div className = "app" >
            <Header /> ←
            <Outlet />
        </div>
    );
};
```

Here we will have same headed for all pages

this outlet will automatically selects the as per the path

```
const app Router = create Browser Router([
    { path : "/"
      element : <App Layout />

        * children : [
            {
            path : "/"    ← for this path
            element </Body>       body element will come
            },
            { path : "/about",
              element : <About/>
            },
            { path : "/contact",
              element : <Contact/>
            },
        ],

        errorElement : <Error/>
    },
]);
```

Link from react-router-dom

→ import { Link } from React-Router-dom.

↳ It do not reloades the page and faster than href.

`<Link to = "/contact">...</Link>`

∴ Always use this instead of <a> anchor tag.

∴ [React is called as Single Page Application]

Server Side Routing → A network call is made with requirements and a page is sent to client

Client Side Routing → No network call is made Page is updated on the client side only.

ex - Using (Link) from react-Router-dom

↳ Single page application

Dynamic Routes (for each restaurant)

this part is dynamic

```
{
  Path : "/restaurants/:resId",
  element : <RestaurantMenue/>
}
```

① create a restaurant component
② create a dynamic route
③ import restr component

Getting restaurants data from swiggy api

useEffect Hook

In restaurantMenue js

```
import { useEffect } from "react";
const RestaurantMenu = () => {

    useEffect(() => {
        fetchMenu();
    },[]);
```

↑ empty array so that it makes the api call only one time

```
const fetchMenu = async () => {
    const data = await fetch(".....");
    const json = await data.json();

    setResInfo(json.data);  // after making use state
```

```
import { useState } from "react";
const Restaurant Menue = () => {
    const [resInfo, setResInfo] = useState(null);

    [ useEffect
      Peter

    if (resInfo === null) return <Shimmer/>   ] shimmer
                                                  UI
```

④ Make state variable

⑤ Make shimmer UI.

⑥ Import that from "./Shimmer"

⑦ get the ~~display~~ Menu data from api

⑧ Map over all the dishes

To make dynamic restaurants card we are
using Hook

```
import { useParams} from "react-router-dom"

const RestaurantMenu = () => {
    const { resId } = useParams();
```

Now to make the components clickable use Link component
in ~~the~~ Body.js   ① import { Link} from "react-router-dom"

| Link is nothing just a anchor tag wrapped to keep
  track |

→① So we are getting the restaurant Id from the parameter

② And we adding the ~~th~~ (resId) or we are making
   our each restaurant clickable by wink Link tag

③ It is done while we map the restaurants
   as a clickable link.