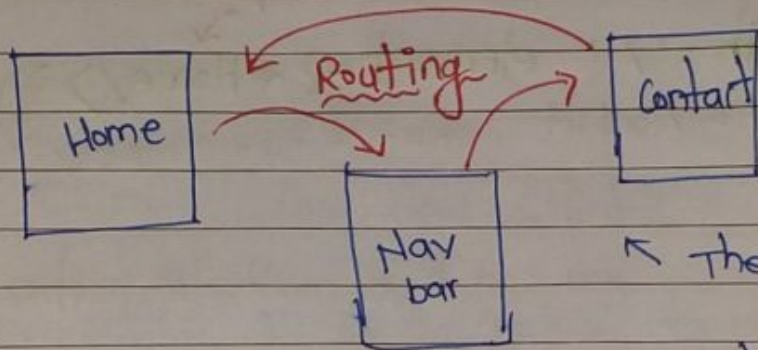# # React Router → SPA approach

avoids page from reload!

single HTML file ∴ single page

dynamically changing the UI

← these all are not "pages"
They're components

One line → "React is all about components" ☺

↳ we can navigate betⁿ multiple pages without refreshing the page..

install → (npm install react-router-dom)

now, to make ur app ready for routing, wrap your <APP> component inside < BrowserRouter) tag.

inside index.js ↴   → import { BrowserRouter } from "react-router-dom"

```
<BrowserRouter>
    <APP/>
</BrowserRouter>
```

Now, we will be defining routes / paths..

this is nothing but routing.

Creating your own routes

↳ you need to define ⟨Routes⟩ tag
& inside this tag you can create a single single Route

```
<Routes>
    <Route path="/" element={<Home/>} />    ← component
    <Route path="/contact" element={<Contact/>} />
</Routes>
                    ↑
            you can have multiple routes..
```

edge cases

① path="/" → this is your home page / by default

② path="/contact" → this is also valid
        ↑
    any component

③ path="*" → when any of the route is not matched
        ↓
    basically page not found..

        ↓

user can't type URL of your component all the time
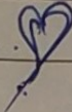that is the reason we create navbar..

now, we connect our routes with the components.
        ↓
& that work is done by ⟨Link to="">⟩ tag
It is same like ⟨a⟩ tag

i.e. `<Link to="/"> Home </Link>`
`<Link to="/about"> About </Link>`
:

& so on.. ♡
:

What is NavLink ?

it behaves same like `<Link>`
+
also add's an (active) class to the current route

● what does that mean ????
↳ Home   About   Contact
in normal case (`<Link>`) when you click on
any button you won't see any changes in your
HTML
⟶ to see the changes (inspect)
_____

but, `<NavLink>` -- Home
About
Contact

● now, when you click on any button then,
(class = "active")
is automatically added to the current route only.

∴ now in your index.css you can define

.active {
color: aqua;
}   Home About Contact

                                    ↗
                          This is the benefit of
                                NavLink

Kya baat # samaj mein aati hai ??
↳ yes ☺

we know

    "/" → Home

    /about → About

    /Contact → Contact

/ ⇒ Parent

/ \ / \ \ ⇒ childs

about ...

∴ you can create Home as parent for other components..

```
<Routes>
    <Route path="/" element={<Home />}>
        <Route path="/about" element={<About />}
        <Route path="         "
                                              }
    </Route>
</Routes>
```

but this is going to create one problem..
i.e Every time Home component is displayed..

✓ solution
write <outlet /> tag in Home component...
    ↳ gives permission to render ~~defined~~ child <u>Routes</u>
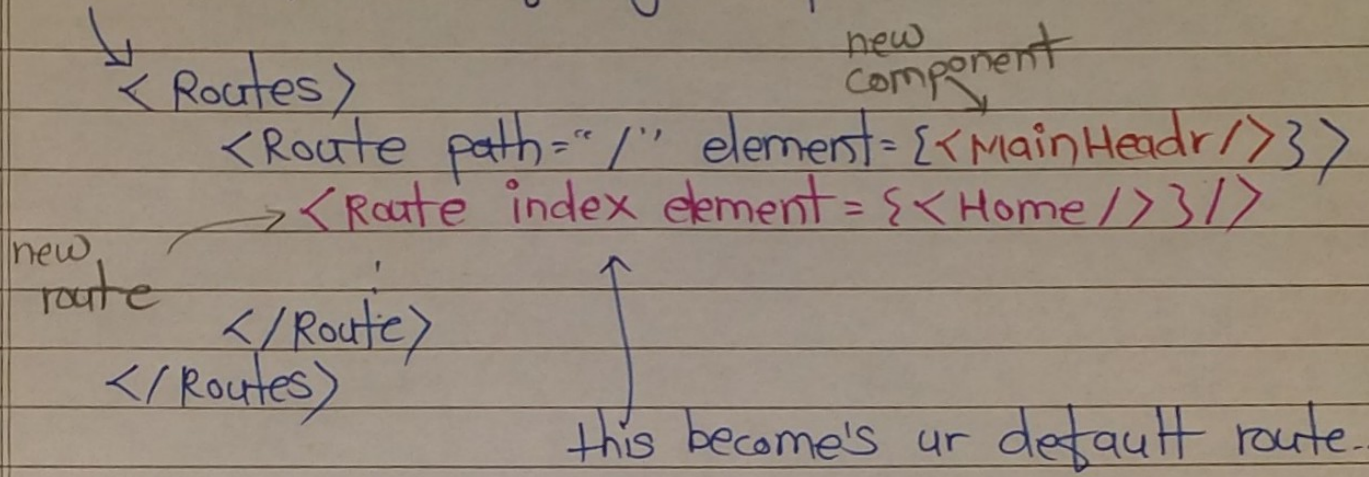
abhi bhi problem hai...
↳ two components are rendered at a time..
    i.e. Home & other everytime..

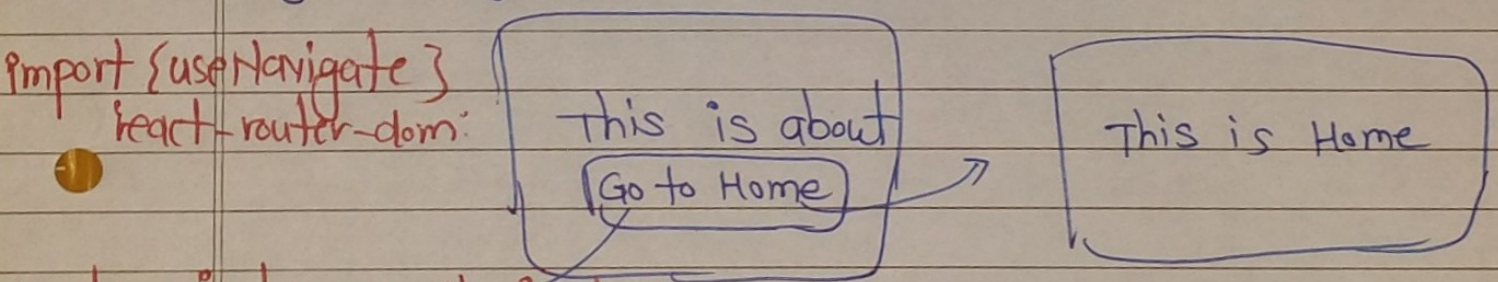∴. Now, you define a new route which will render your Home component

&

new component for your parent route..

↓

```
< Routes >
    <Route path="/" element={<MainHeadr/>}>    ← new component
        <Route index element={<Home/>}/>
new
route
    </Route>
</Routes>
```

this become's ur default route.

use Navigate

↳ you can switch from one page to another by clicking a button.

e.g. if you are on `about` page.

import {useNavigate}
   react-router-dom:

| This is about |
| Go to Home |

→ | This is Home |

const navigate = useNavigate();
∴.   onClick = { pageHandler }

```
function pageHandler()
{
    navigate('/');                    navigate(-1);
}                                                    ↓
                                         one page back
                                              or
```

That's all 😊

previous page