



React 'Router' {

[A fully-featured routing solution
for react apps]

< Routing is the capacity to show different pages
to the user >

}

Client Side 'Routing';

Client side routing allows your app to update the URL from a link click without making another request for another document from the server. Instead, your app can immediately render some new UI and make data requests with `fetch` to update the page with new information.

This enables faster user experiences because the browser doesn't need to request an entirely new document or re-evaluate CSS and JavaScript assets for the next page. It also enables more dynamic user experiences with things like animation.

Client side routing is enabled by creating a Router and linking/submitting to pages with `Link` and `<Form>`.



How to add 'react-router' {

01 Install react-router-dom

< It offers pre-developed components, Hooks and utility functions to create modern routing strategies >

02 Import Browser Router, Routes & Route

< The Root Component - To set path & element >

03 Import Link & NavLink & Outlet

< To set the URL and keep track of browsing history and render the current route selected >

}

01 {

[Install react-router-dom]

< This tutorial uses React Router v6
If you are upgrading from v5, you will need to use
the @latest flag >

}

Basic React Router Setup; {

<p>

The first thing that you'll need to do is
install React Router DOM using npm (or yarn):

</p>

Npm i react-router-dom

}

02 {

[Import Browser Router, Routes & Route]

< We wrap our content first with <BrowserRouter>

Then we define our <Routes>. An application can have

Multiple nested <Route> s >

}

React Router 'Components' {

Create and render Browser router, Routes and Route in `App.js`:

```
import {BrowserRouter , Routes , Route} from "react-router-dom";
```

```
}
```

Set the 'props' {

We declare `<Route>`s within the `<Routes>` component as children. We can declare as many `<Route>`s as we like and we need to provide at least two props to each `<Route>`, `path` and `element`:

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

```
}
```



Path and Element {

< The path prop specifies on what path of our app a given route is located. >

< The element prop is used to display a specific component for our path >

}

03 Import Link & NavLink ; }

We actually want to create some links so we can move around our application more easily instead of having to change the URL manually in the browser. There is 2 components for this purpose:

```
import {Link , NavLink , Outlet} from 'react-router-dom'
```

```
}
```



Link Component {

`<Link>` is used to set the URL and keep track of browsing history.

Anytime we link to an internal path, we will use `<Link>` instead of ``.

It accepts the ``to`` prop instead of ``href`` prop, which specifies where we want the link to navigate our user to.

```
<Link to="/About">About</Link>
```

}

NavLink Component {

Additionally, React Router DOM gives us a NavLink component which is helpful in the event that we want to apply some special styles.

If we are on the current path that the link points to, this allows us to create some **active** link styles to tell our users, by looking at our link, what page they're on.

Notice: The className in bellow command is a property.

NavLinkProps.className?: string

```
<NavLink to="/Tutorials" className={styles['nav-link']}>
  Tutorials
</NavLink>
```

}

Active class of NavLink {

<p>In a css component we style the active class of NavLink.

We use ``:global`` pseudo class</p>

```
.nav-link:global.active
```

```
{
```

```
background-color: orange;
```

```
}
```

```
}
```

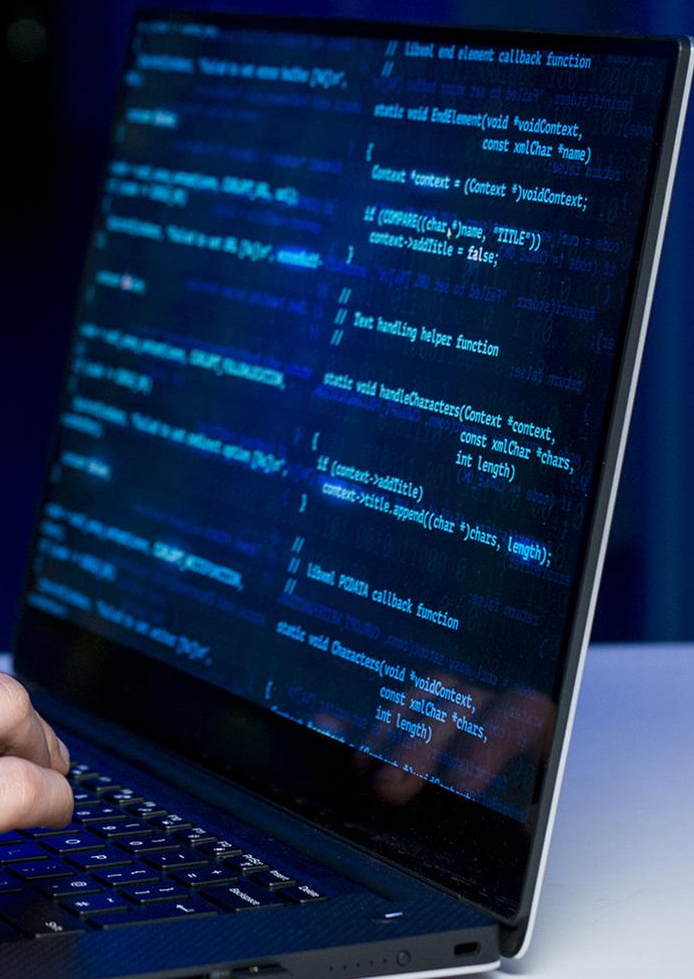
💎 Outlet component; {

The `<Outlet>` renders the current route selected.
When you have nested `<route>`, you should use `<Outlet/>` component to submit where you want to insert this nested route component.

```
    <div className={styles['content']}>  
      <Outlet/>  
    </div>  
  }
```

You can see an
'Example' in the
next slide {

}



```
1  import './App.css';
2
3  import {BrowserRouter ,Route, Routes} from 'react-router-dom'
4  import GrayCntAbout from './components/gray-cnt-about';
5  import MainLayout from './layout/main-layout';
6  import {W3schoolsPage} from './pages'
7  import { TutorialPage } from './pages';
8  import {HtmlTutorialPage} from './pages'
9
10
11 function App() {
12   return (
13     <BrowserRouter>
14       <MainLayout>
15         <GrayCntAbout/>
16         <W3schoolsPage/>
17
18         <Routes>
19           <Route path="/" element={<W3schoolsPage/>}/>
20           <Route path='Tutorials' element={<TutorialPage/>}>
21             <Route path='HtmlTutorial' element={<HtmlTutorialPage/>}/>
22           </Route>
23         </Routes>
24       </MainLayout>
25     </BrowserRouter>
26   );
27 }
28
29
30 export default App;
31
```



```
1 Thanks; {
```



```
2  
3 'Do you have any questions?'
```



```
4  
5 homeira2010@gmail.com
```



```
6  
7 https://www.linkedin.com/in/homeira-haddad-792ab226b
```

```
8  
9  
10 CREDITS: This presentation template was  
11 created by Slidesgo, including icons by  
12 Flaticon, and infographics & images by Freepik
```

```
13 < Please keep this slide for attribution >
```

```
14 }
```



Resources {

<https://www.reactrouter.com/>

<https://www.freecodecamp.org/>

<https://www.w3schools.com/>

}