

Week retrospective

What did you **achieve last week**?

Try to remember the main things (projects, tools, new methods)

What **worked well**? What did **not**?

- What should you start doing?
- Stop doing?
- Continue doing?

Reactjs

week 6

Margit Tennosaar

This session

- React router

Conditional rendering

Conditional rendering in React means showing different parts of your app depending on certain conditions. It's like using "if" statements to decide what to show or hide in the app's interface, making the app interactive and responsive to user actions or changes in data.

Various approaches, including ternary operators, logical `&&` operator, and `if-else` statements, can be used to implement conditional rendering.

```
function Greeting(props) {  
  if (props.user === 'Margit') {  
    return <p> Welcome {props.user}</p>  
  }  
  return <p>Please log in</p>  
}
```

```
function Greeting({ user }) {  
  return (user === 'Margit') ? <p> Welcome {user}</p> : <p>Please log  
in</p>  
}
```

```
<Greeting user='Margit'/>;
```

```
function Greeting(props) {  
  if ({props.isLoggedIn}) {  
    return <UserGreeting />;  
  }  
  return <GuestGreeting />;  
}
```

```
<Greeting isLoggedIn={false}/>;
```

```
function WelcomeMessage() {  
  const [isLoggedIn, setIsLoggedIn] = useState(false);  
  
  return (  
    <div>  
      <button onClick={() => setIsLoggedIn(!isLoggedIn)}>  
        {isLoggedIn ? 'Log Out' : 'Log In'}  
      </button>  
      {isLoggedIn && <p>Welcome back, user!</p>}  
    </div>  
  );  
}
```

```
<span className="material-icons">  
  {likes >= 0 ? 'favorite' : 'heart_broken'}  
</span>
```


Multiple pages in SPA

Traditional webpage

/



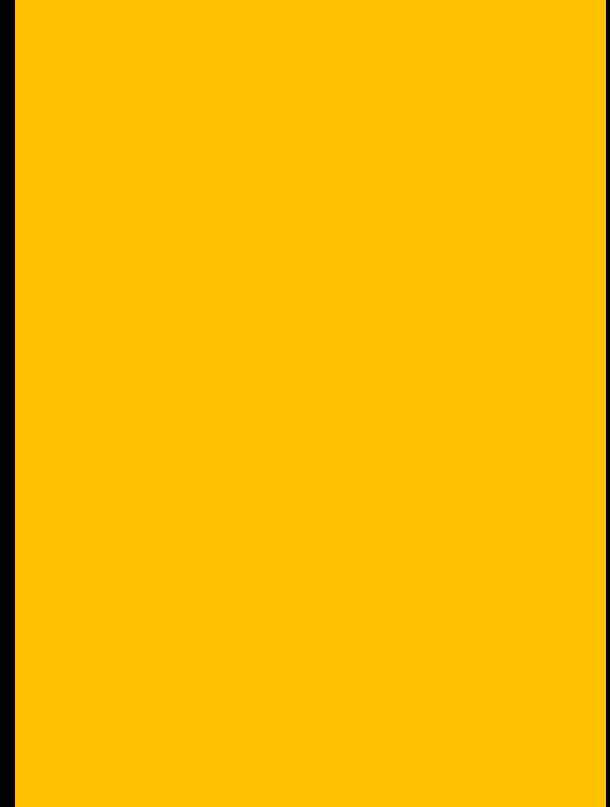
index.html

/persons



persons.html

/about



about.html

React Router is a library for handling navigation in React applications, especially in the context of Single Page Applications (SPAs).

SPAs are web applications that load a single HTML page and dynamically update the content as the user interacts with the app, without requiring a full page reload.

Routing means **conditional rendering** of components based on the url in the browser.

When the URL in the address bar changes, the content of the page is only manipulated using Javascript, and the **browser will not load new content form the server**. Using the back and forward actions, as well as making bookmarks, is still logical like on a traditional web page.

How to use

```
npm install react-router-dom
```

- createBrowserRouter
- RouterProvider
- Link
- NavLink
- Outlet

Setting up the routes

```
import { createBrowserRouter, RouterProvider } from 'react-router-dom';

const router = createBrowserRouter([
  {path: '/', element:<Home/>},
  {path: '/persons', element:<Persons/>},
]);

ReactDOM.createRoot(document.getElementById('root')).render(
  <React.StrictMode>
    <RouterProvider router={router} />
  </React.StrictMode>
);
```

Setting up the routes (older approach)

```
import { createBrowserRouter, createRoutesFromElements, Route, RouterProvider }  
from 'react-router-dom';  
  
const routesDefinitions = createRoutesFromElements(  
  <Route>  
    <Route path='/' element={<Home/>}/>  
    <Route path='/products' element={<Persons/>}/>  
  </Route>  
)  
  
const router = createBrowserRouter(routesDefinitions)  
  
ReactDOM.createRoot(document.getElementById('root')).render(  
  <React.StrictMode>  
    <RouterProvider router={router} />  
  </React.StrictMode>  
);
```

Creating links

```
import { Link } from 'react-router-dom';

function Home() {
  return (
    <>
      <h1>Home will be here</h1>
      <a href="/persons">Persons</a> // DON'T DO LIKE THAT
      <Link to="/persons">Persons</Link> // correct way
    </>
  );
}

export default Home;
```


Navigation (or Route Changers)

React Router provides a `<Link>` component to create links in your application. Wherever you render a `<Link>`, an anchor (`<a>`) will be rendered in your HTML document.

The `<NavLink>` is a special type of `<Link>` that can style itself as "active" when its `to` prop matches the current location.

Nested routes

```
const router = createBrowserRouter([
  {
    path: '/',
    element: <Root />,
    errorElement: <ErrorPage />,
    children: [
      { path: '/', element: <Home /> },
      {
        path: '/persons',
        element: <Persons />,
      },
    ],
  },
]);
```

```
export default function Root() {
  return (
    <>
      <Header/>
      <Outlet/>
      <Footer/>
    </>
  );
}
```

Task

Refactor your zoo app:

- Add links to:
 - **Home** // only two big images – links to animals and bird pages
 - **Animals** // listing all animals, have animals search
 - **Birds** // listing all birds, have birds search
 - **About** // page about this project and summary of the features and what you learned with this project