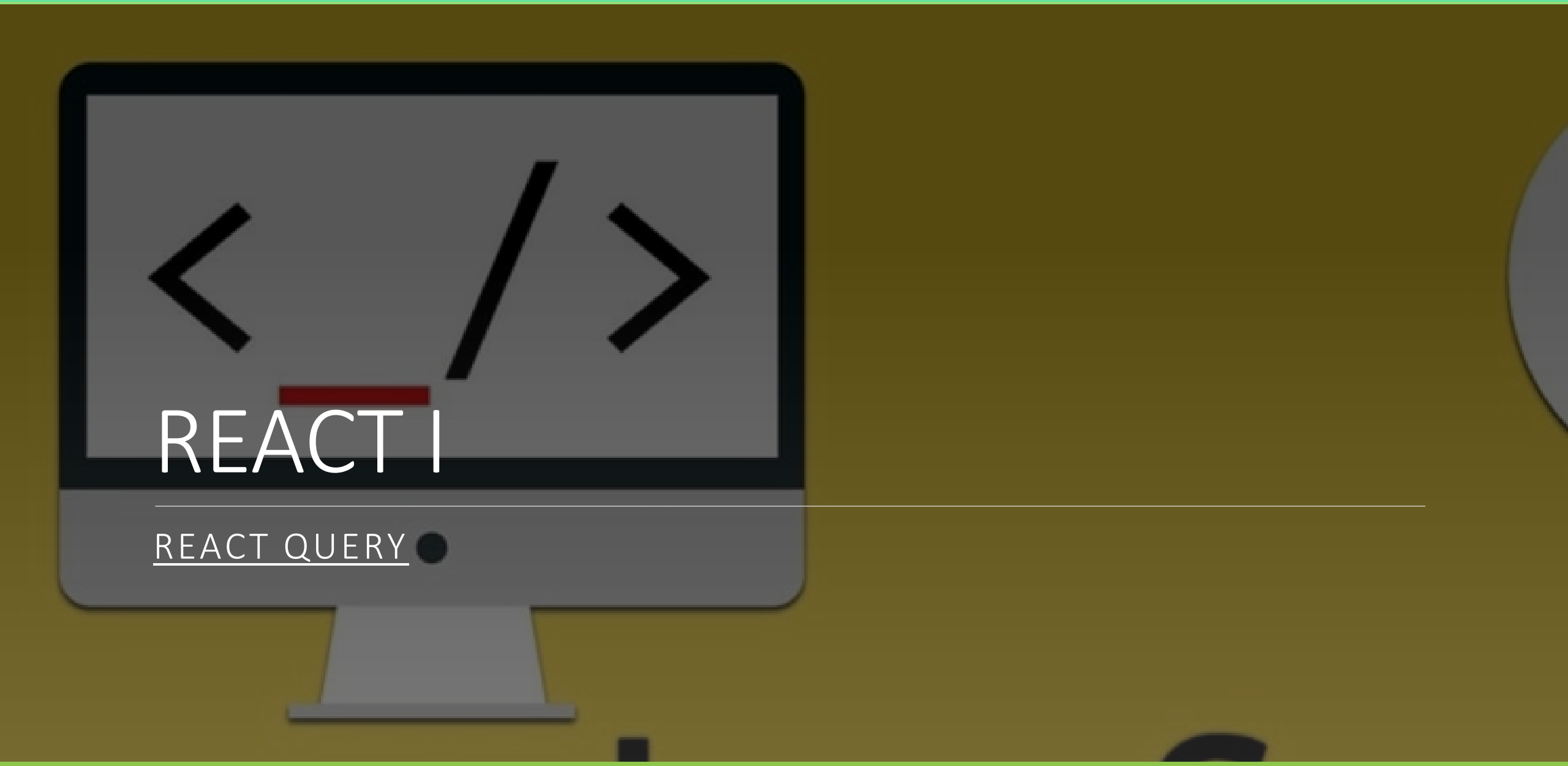

REACT I

REACT QUERY

yarn add react-query

```
import React from 'react'
import ReactDOM from 'react-dom/client'
import App from './App'
import './index.css'
import { BrowserRouter, Routes, Route, Navigate, useParams } from 'react-router-dom'
import { Home } from './Home'
import { Producto } from './Producto'
import { QueryClient, QueryClientProvider } from 'react-query'


const queryClient = new QueryClient()


ReactDOM.createRoot(document.getElementById('root') as HTMLElement).render(
  <QueryClientProvider client={queryClient}>
    <BrowserRouter>
      <Routes>
        <Route path="/" element={<Home />}>
          <Route index element={<p>defecto</p>}></Route>
          <Route path="/productos" element={<p>productos</p>}></Route>
          <Route path="/productos/:id" element={<Producto></Producto>}></Route>
          <Route path="/clientes" element={<p>clientes</p>}></Route>
          <Route path="*" element={<Navigate to="/" replace />} />
        </Route>
      </Routes>
    </BrowserRouter>

  </QueryClientProvider>
)
```

REACT-QUERY

```tsx
import { useParams } from "react-router-dom";
import { useQuery } from "react-query"
interface IProducto {
    pro:number
    nombre:string
}
export function Producto() {

    async function getData() {
        return [
            { "pro": 1, nombre: "n1" },
            { "pro": 2, nombre: "n2" },
            { "pro": 3, nombre: "n3" },
            { "pro": 4, nombre: "n4" }
        ]
    }
    const params = useParams();
     const { data, isLoading } = useQuery(["q1"], getData)

    if (isLoading)
        return <p>Loading</p>
    return (
        <div>

            {JSON.stringify(data)}
            <ul>
                {
                    data.map((item: IProducto, index: number) =>
                        <li key={index}>{item.pro}</li>
                    )
                }
            </ul>
        </div>
    )
}
```

SIMULACIÓN QUERY

```
import { useState } from 'react'
import {useMutation} from 'react-query'
export function Mutacion(){
    const [state, setState] = useState<number[]>([])
    const {mutate, isLoading, isError, isSuccess} = useMutation((newTodo:number) => {
        setState([...state, state.length])
        return 'ok'
      })
    return <div>
        <div>
        {
            state.map((item:number, index:number) => <p key={index}>{item}</p>)
        }
        </div>
        <button onClick={() => mutate(1)}>Add</button>
    </div>
}
```