

11. 리액트 라우터 (1)

Prof. Seunghyun Park (sp@hansung.ac.kr)

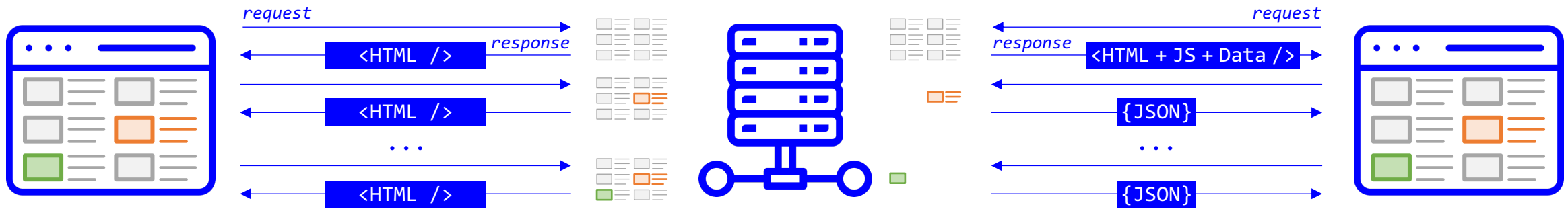
Division of Computer Engineering

학습 목표: 11장. 리액트 라우터 (1)

- Single Page Application
 - Client-side rendering
- React Router
 - `<Routes />`, `<Route />`로 컴포넌트 전환 (주소 직접 입력)
 - `useLocation()` hooks로 주소정보 반환
 - `<Link />`로 라우터 링크 제공
 - 중첩 Route와 `<Outlet />`, `<Navigate />`
 - `useRoutes()` hooks
- 동적 라우팅
 - 라우팅 파라미터
 - 쿼리
- React Context (6장)

Single Page Application

페이지 전환



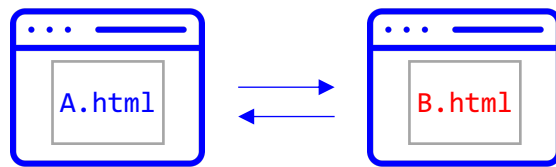
페이지 요청 및 처리

- 매 요청시마다 서버는 렌더링 할 페이지를 반환
 - 변경되지 않는 영역까지 request & response 수행
 - 서버 리소스와 네트워크 트래픽 낭비

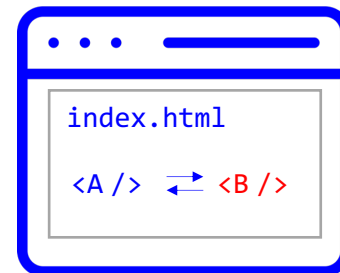
• Single page application

- 뷰 렌더링을 브라우저가 담당 (client-side rendering)
- 하나의 페이지를 로딩한 후, 변경된 부분만 교체

페이지 이동



별도의 페이지로 전환

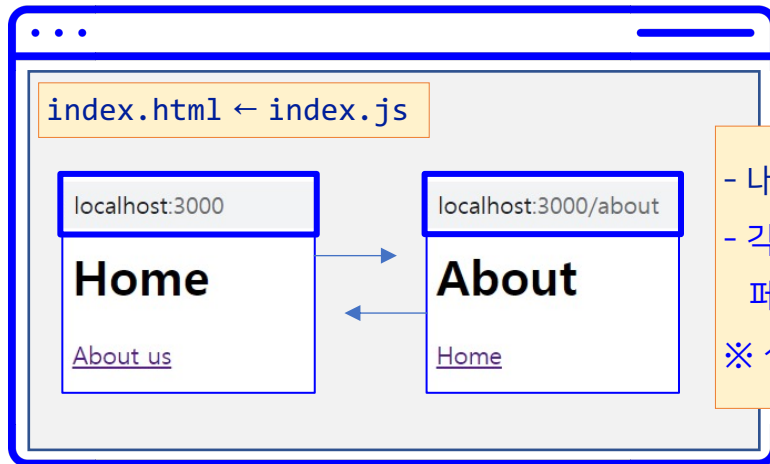


- 페이지 단위의 컴포넌트 반환을 위해 라우팅 필요
 - ※ 라우팅: 클라이언트 요청에 맞는 페이지 반환
- 리액트 라이브러리는 라우팅 기능 제공하지 않으므로 별도의 라우팅 라이브러리 필요

- 리액트 라우터 (react-router-dom)

라우팅 기능을 제공하지 않는 리액트에서
컴포넌트 기반으로 라우팅을 구현하기 위해 사용하는 라이브러리

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



- 내부에 2개의 컴포넌트 구성
- 각 컴포넌트 전환을
페이지 이동처럼 활용

※ 실제로는 하나의 페이지

```
/* ch11/proj/ch11-00/src/index.js */  
import { createRoot } from 'react-dom/client';  
import { RouterProvider, createBrowserRouter, Link }  
  from 'react-router-dom';
```

```
const router = createBrowserRouter([  
  {  
    path: "/",  
    element: (  
      <div>  
        <h1>Home</h1>  
        <Link to="about">About</Link>  
      </div> )  
    },  
  ],  
  {  
    path: "about",  
    element: (  
      <div>  
        <h1>About</h1>  
        <Link to="/">Home</Link>  
      </div> )  
    }  
  ]  
]);
```

```
const element = document.getElementById('root')  
const root = createRoot(element);  
root.render( <RouterProvider router={router} /> );
```

Routes and Route 컴포넌트

```
/* ch11/proj/01-1/src/index.js */
import { createRoot } from 'react-dom/client';
import { BrowserRouter } from 'react-router-dom';
import App from './App';
```

```
const element = document.getElementById('root');
const root = createRoot(element);
```

```
root.render(
```

```
  <BrowserRouter>
```

```
    <App />
```

```
  </BrowserRouter>
```

```
);
```

`<BrowserRouter />`
브라우저의 주소 표시줄에서
현재 주소, History API 활용하여
주소 탐색을 지원하는 컴포넌트

```
<Routes />
```

- 여러 <Route />를 포함
- path가 일치하는 하나의 Route를 렌더링

```
<Route />
```

- 주요 속성은 경로와 엘리먼트
- 중첩된 <Route />도 가능

```
/* ch11/proj/01-1/src/App.js */
```

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

```
import Home from './components/Home';
```

```
import About from './components/About';
```

```
const App = () => (
```

```
  <Routes>
```

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

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

```
  </Routes>
```

```
)
```

```
export default App;
```

주소 창에 입력된 경로로 페이지 이동

```
<App />
```

```
  /
```

```
  <Home />
```

./src/components/Home.js

```
  about
```

```
  <About />
```

./src/components/About.js

```
/* ch11/proj/01-1/src/components/Home.js */
```

```
1 const Home = () => (
```

```
  <div>
```

```
    <h1>Home</h1>
```

```
    <p>Home, the first page.</p>
```

```
  </div>
```

```
);
```

```
export default Home;
```

localhost:3000

Home

Home, the first page.

```
/* ch11/proj/01-1/src/components/About.js */
```

```
2 const About = () => (
```

```
  <div>
```

```
    <h1>About</h1>
```

```
    <p>About, introducing page.</p>
```

```
  </div>
```

```
);
```

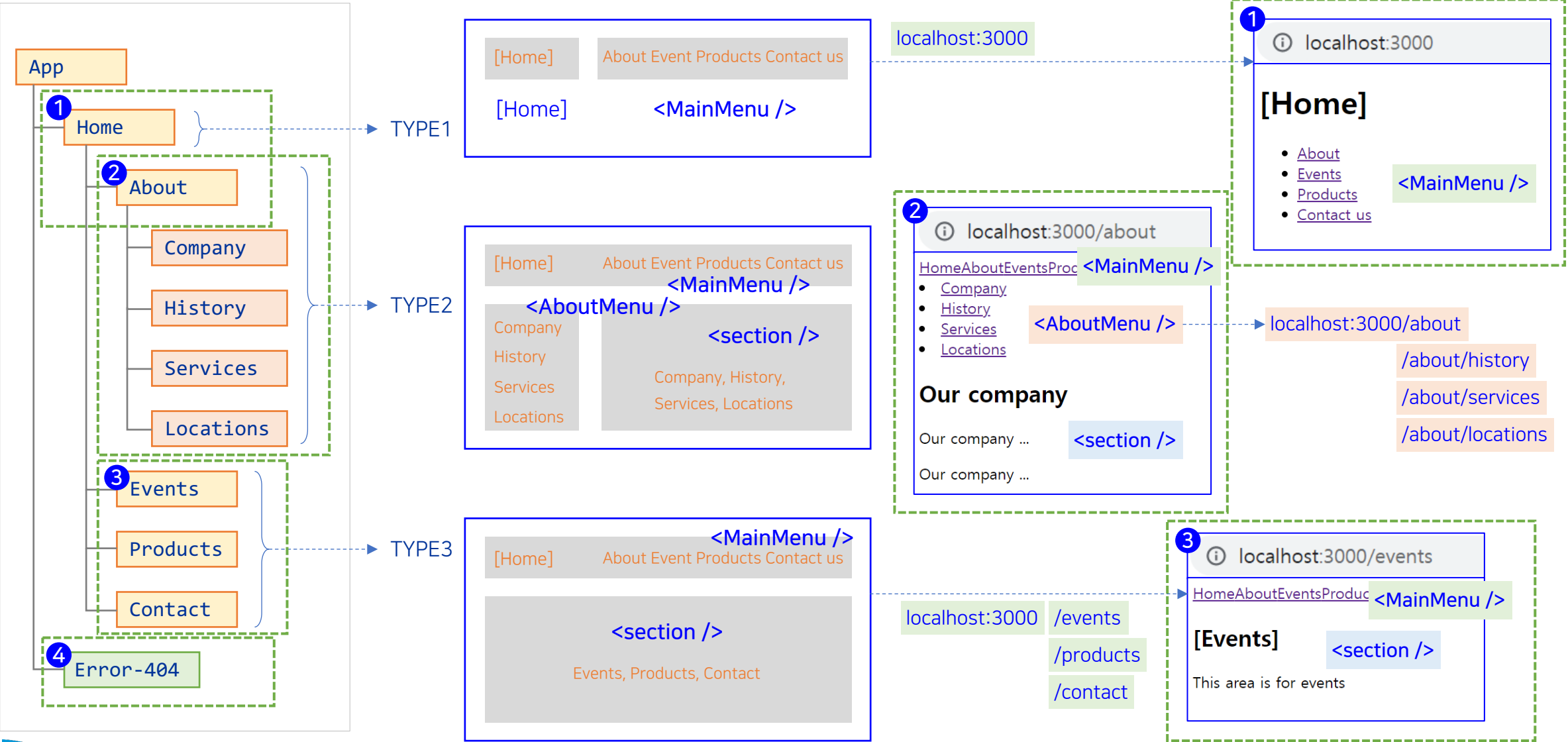
```
export default About;
```

localhost:3000/about

About

About, introducing page.

페이지 전환을 위한 맵



Route로 컴포넌트 전환

<Routes>

<Route path="상대 주소" element={컴포넌트} />

...

</Routes>

```
/* ch11/proj/01-2/src/App.js */
```

```
import { Route, Routes } from 'react-router-dom';
import { Home, About, Events, Products, Contact }
  from './components/Pages';
```

const App = () => (<App />은 하위에 5개의 컴포넌트 포함

```
<Routes>
  <Route path="/" element={<Home />} />
  <Route path="/about" element={<About />} />
  <Route path="/events" element={<Events />} />
  <Route path="/products" element={<Products />} />
  <Route path="/contact" element={<Contact />} />
</Routes>
)
```

```
export default App;
```

```
/* ch11/proj/01-2/src/components/Pages.js */
```

```
export const Home = () => (
  <div>
    <h1>[Home]</h1>
  </div>
);
```

```
export const About = () => (
  <div>
    <h1>[About]</h1>
  </div>
);
```

```
export const Events = () => (
  <div>
    <h1>[Events]</h1>
  </div>
);
```

```
export const Products = () => (
  <div>
    <h1>[Products]</h1>
  </div>
);
```

```
export const Contact = () => (
  <div>
    <h1>[Contact]</h1>
  </div>
);
```

주소 창에 직접 경로를 입력하여 페이지 전환
(컴포넌트 렌더링)

localhost:3000

[Home]

localhost:3000/about

[About]

localhost:3000/events

[Events]

localhost:3000/products

[Products]

localhost:3000/contact

[Contact]

useLocation() hooks로 주소정보 반환

<https://reactrouter.com/en/main/hooks/use-location>

```
/* ch11/proj/01-3/src/App.js */
import { Route, Routes } from 'react-router-dom';
import { Home, About, Events, Products, Contact,
  Whoops404 } from './components/Pages';

const App = () => (
  <Routes>
    <Route path="/" element={<Home />} />
    <Route path="/about" element={<About />} />
    <Route path="/events" element={<Events />} />
    <Route path="/products" element={<Products />} />
    <Route path="/contact" element={<Contact />} />
    <Route path="*" element={<Whoops404 />} />
  </Routes>
)

export default App;
```

에러처리 (404 에러):
정의되지 않은 주소를 요청시 에러페이지로 전환

```
/* ch11/proj/01-3/src/components/Pages.js */
import { useLocation } from "react-router-dom";
export const Home = () => (
  <div>
    <h1>[Home]</h1>
  </div>
);
export const About = () => (...);
export const Events = () => (...);
export const Products = () => (...);
export const Contact = () => (...);
export const Whoops404 = () => {
  let location = useLocation();
  console.log(location);
  return (
    <div>
      <h1>Resource not found at {location.pathname}</h1>
    </div>
  );
};
```

useLocation()

- 현재 URL 주소를 받아오는 hooks
- react-router-dom에 포함

[[Prototype]]: Object
hash: ""
key: "default"
pathname: "/no"
search: ""
state: null

localhost:3000 localhost:3000/events ⓘ localhost:3000/no

[Home] [Events] Resource not found at /no

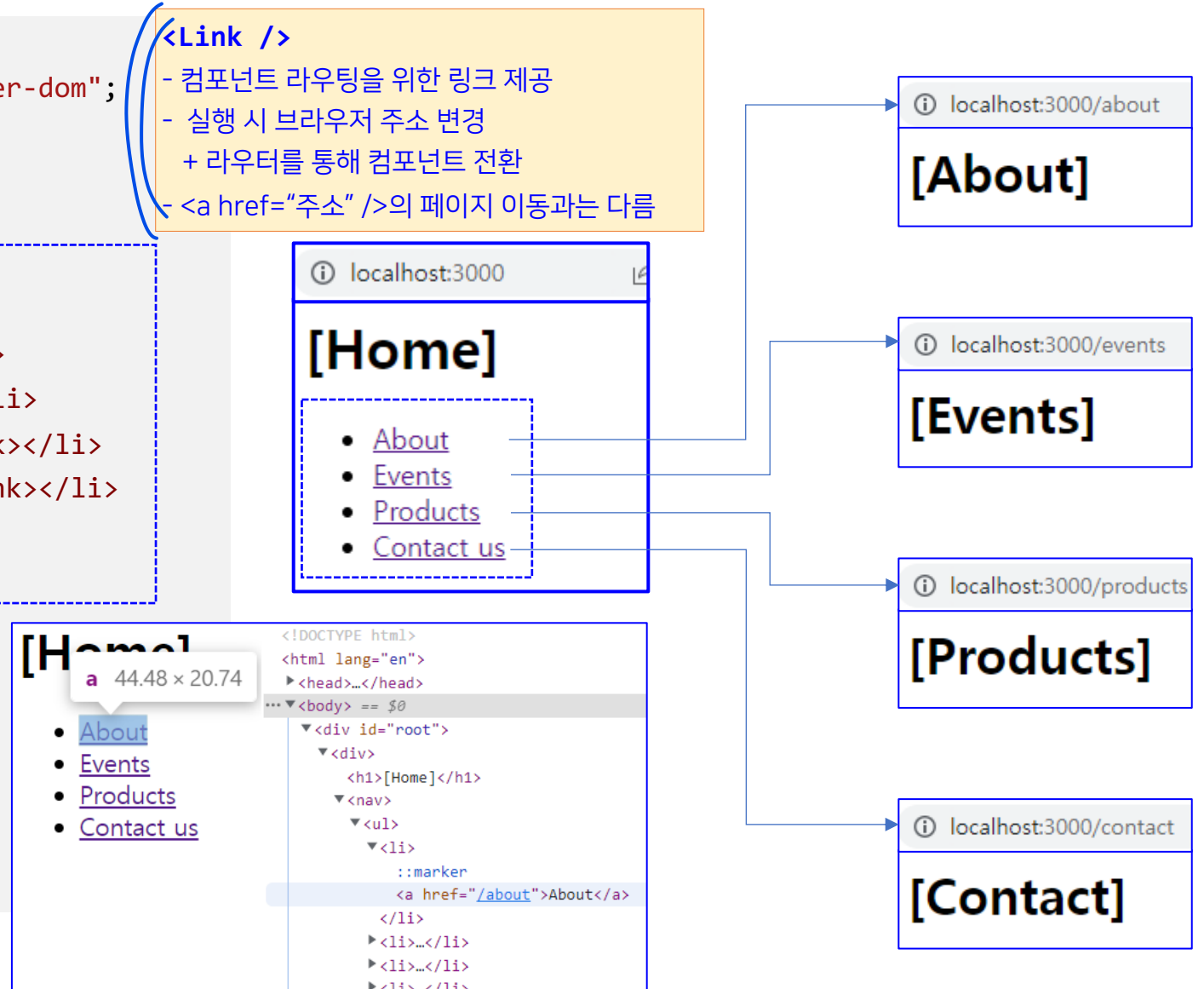
<Link />로 라우터 링크 제공

<https://reactrouter.com/en/main/components/link>

```
/* ch11/proj/01-4/src/components/Pages.js */
import { useLocation, Link } from "react-router-dom";
export const Home = () => (
  <div>
    <h1>[Home]</h1>
    <nav>
      <ul>
        <li><Link to="about">About</Link></li>
        <li><Link to="events">Events</Link></li>
        <li><Link to="products">Products</Link></li>
        <li><Link to="contact">Contact us</Link></li>
      </ul>
    </nav>
  </div>
);
export const About = () => (...);
export const Events = () => (...);
export const Products = () => (...);
export const Contact = () => (...);
export const Whoops404 = () => {...};
```

<Link />

- 컴포넌트 라우팅을 위한 링크 제공
- 실행 시 브라우저 주소 변경
+ 라우터를 통해 컴포넌트 전환
- 의 페이지 이동과는 다름



중첩 Route와 <Outlet />

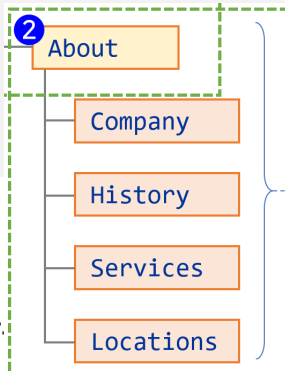
서브 메뉴

<https://reactrouter.com/en/main/components/outlet>

```
/* ch11/proj/01-5/src/App.js */
import { Route, Routes } from 'react-router-dom';
import { Home, About, Events, Products, Contact, Whoops404,
  Services, History, Locations } from './components/Pages';
```

```
const App = () => (
  <Routes>
    <Route path="/" element={<Home />} />
    <Route path="/about/*" element={<About />}>
      <Route path="services" element={<Services />} />
      <Route path="history" element={<History />} />
      <Route path="locations" element={<Locations />} />
    </Route>
    <Route path="/events" element={<Events />} />
    <Route path="/products" element={<Products />} />
    <Route path="/contact" element={<Contact />} />
    <Route path="*" element={<Whoops404 />} />
  </Routes>
)
```

```
export default App;
```



```
/* ch11/proj/01-5/src/components/Pages.js */
import { useLocation, Link, Outlet } from "react-router-dom";
```

```
export const Home = () => (...);
export const About = () => (
  <div>
    <h1>[About]</h1>
    <div>
      <Outlet />
    </div>
  </div>
);
```

상위 컴포넌트 가져와서 함께 보여줌

```
export const Events = () => (...);
export const Products = () => (...);
export const Contact = () => (...);
export const Whoops404 = () => (...);
```

```
export const Services = () => (
  <div>
    <h1>[Services]</h1>
  </div>
);
```

```
export const History = () => (...);
export const Locations = () => (...);
```



<Navigate /> 특정 조건이 발생했을 때 다른 주소로 자동 이동 (리디렉션)

<https://reactrouter.com/en/main/components/navigate>

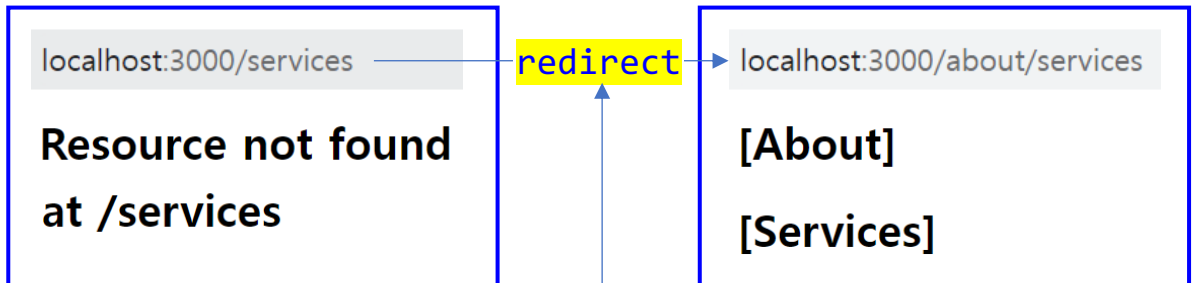
```
/* ch11/proj/01-6/src/App.js */
import { Route, Routes, Navigate } from 'react-router-dom';
import { Home, About, Events, Products, Contact, Whoops404,
Services, History, Locations } from './components/Pages';

const App = () => (
  <Routes>
    <Route path="/" element={<Home />} />
    <Route path="/about/*" element={<About />}>
      <Route path="services" element={<Services />} />
      ...
    </Route>
    ...
    <Route path="/services"
      element={<Navigate replace="true"
        to="/about/services" />} />
    ...
    <Route path="*" element={<Whoops404 />} />
  </Routes>
);

export default App;
```

/services → /Whoops404

/services → /about/services



`<Navigate to={url} replace={boolean} />`

- **to**: 이동할 주소
- **replace**: back을 시도할 때 redirection 직전의 주소로 이동할 것인지 여부

※ `useNavigate()` hooks의 wrapper 컴포넌트로 사용법 동일

useRoutes() hooks

useRoutes() hooks:

- Routes 정보를 객체의 배열로 표현
- 주요 속성: path, element, children

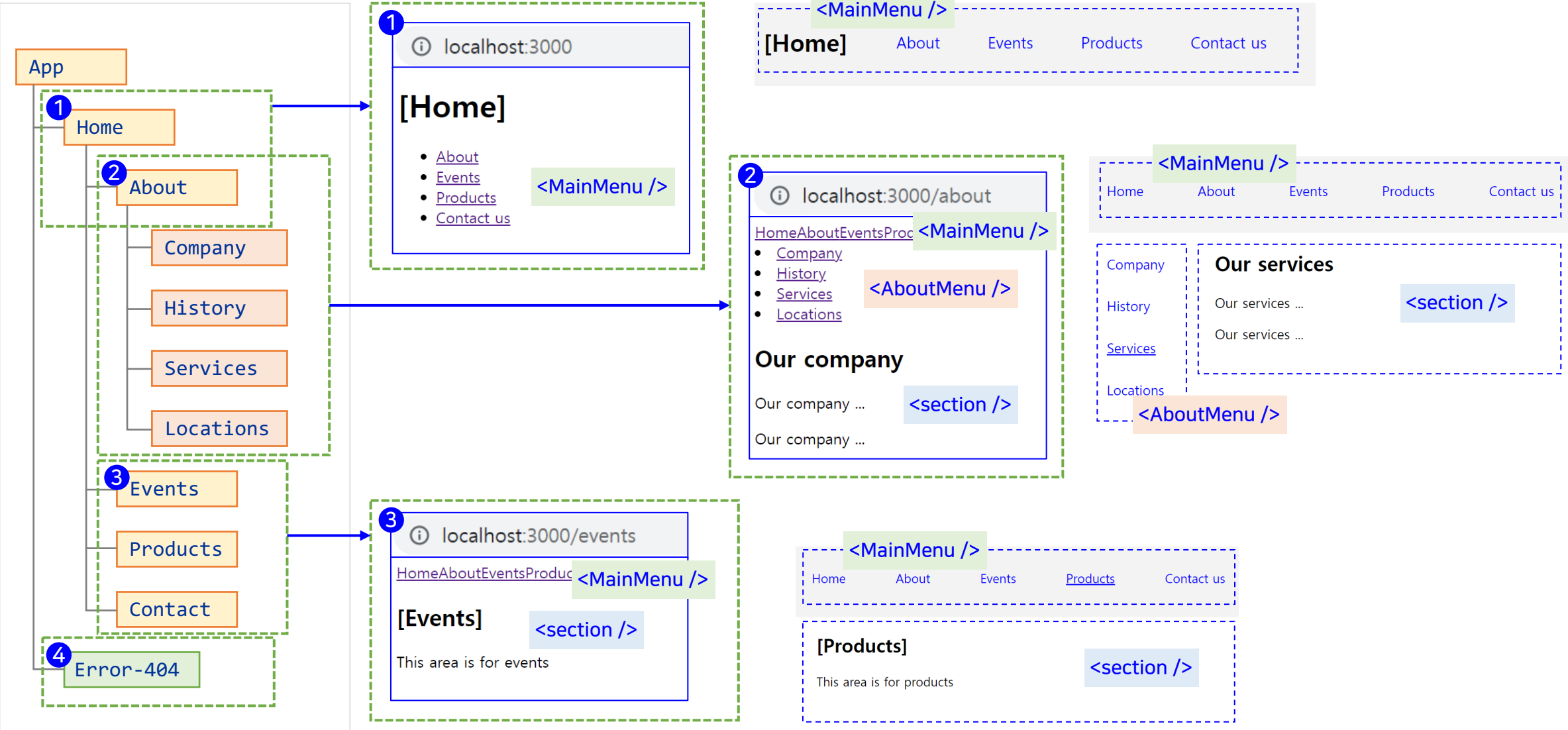
```
/* ch11/proj/01-7/src/App.js */
import { Navigate, useRoutes } from 'react-router-dom';

const App = () => {
  let element = useRoutes([
    { path: "/", element: <Home /> },
    { path: "/about", element: <About />, children: [
      { path: "services", element: <Services /> },
      { path: "history", element: <History /> },
      { path: "locations", element: <Locations /> }
    ] },
    { path: "events", element: <Events /> },
    { path: "products", element: <Products /> },
    { path: "contact", element: <Contact /> },
    { path: "*", element: <Whoops404 /> },
    { path: "services",
      element: <Navigate replace="true" to="/about/services" /> },
    { path: "history",
      element: <Navigate to="/about/history" /> },
    { path: "locations",
      element: <Navigate to="/about/locations" /> }
  ]);
  return element;
};
export default App;
```

```
/* ch11/proj/01-6/src/App.js */
import { Route, Routes, Navigate } from 'react-router-dom';

const App = () => (
  <Routes>
    <Route path="/" element={<Home />} />
    <Route path="/about/*" element={<About />}>
      <Route path="services" element={<Services />} />
      <Route path="history" element={<History />} />
      <Route path="locations" element={<Locations />} />
    </Route>
    <Route path="/events" element={<Events />} />
    <Route path="/products" element={<Products />} />
    <Route path="/contact" element={<Contact />} />
    <Route path="/services"
      element={<Navigate replace="true" to="/about/services" />} />
    <Route path="/history"
      element={<Navigate to="/about/history" />} />
    <Route path="/locations"
      element={<Navigate to="/about/locations" />} />
    <Route path="*" element={<Whoops404 />} />
  </Routes>
);
export default App;
```

페이지 구성 (예: ch11/proj/01-8)

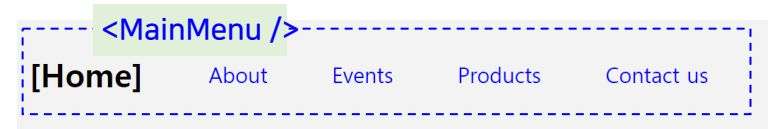


예제코드(ch11/01-8: <App /> - 전체 라우팅 경로 및 계층 정의)

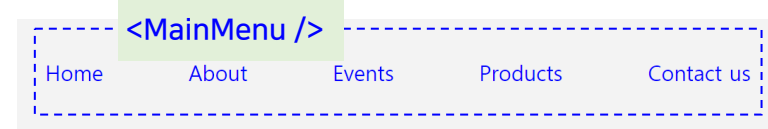
```
/* ch11/proj/01-8/src/App.js */
import { Route, Routes, Navigate } from 'react-router-dom';
import { Home, About, Events, Products, Contact, Whoops404,
Services, History, Locations, Company } from './components/Pages';
```

```
const App = () => (
  <Routes>
    <Route path="/" element={<Home />} />
    <Route path="/about/*" element={<About />} />
    <Route path="/" element={<Company />} />
    <Route path="/services" element={<Services />} />
    <Route path="/history" element={<History />} />
    <Route path="/locations" element={<Locations />} />
  </Route>
  <Route path="/events" element={<Events />} />
  <Route path="/products" element={<Products />} />
  <Route path="/contact" element={<Contact />} />
  <Route path="/services"
    element={<Navigate replace="true" to="/about/services" />} />
  <Route path="/history"
    element={<Navigate replace="true" to="/about/history" />} />
  <Route path="/locations"
    element={<Navigate replace="true" to="/about/locations" />} />
  <Route path="*" element={<Whoops404 />} />
</Routes>
);
export default App;
```

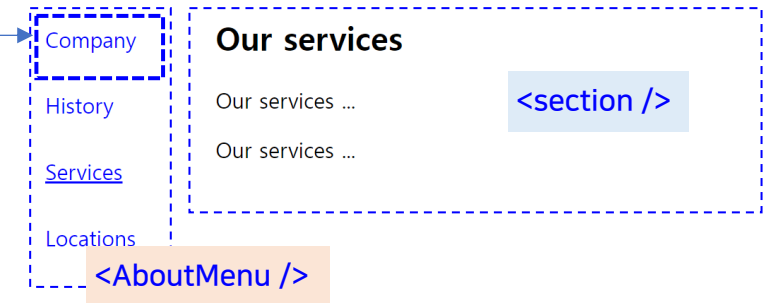
메인메뉴 (<MainMenu />) 중
About 항목의 서브 메뉴에서
default 컴포넌트 <Company /> 생성



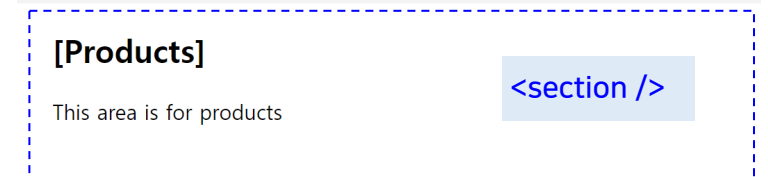
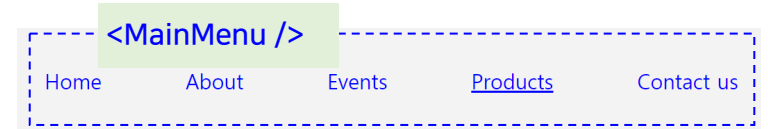
Type 1



Type 2



Type 3



예제코드(ch11/proj/01-8: <MainMenu />와 <AboutMenu />)

```
/* ch11/proj/01-8/src/components/Menus.js */
import { NavLink } from "react-router-dom";

export const MainMenu = () => (
  <nav className="main-menu">
    <NavLink to="/">Home</NavLink>
    <NavLink to="/about">About</NavLink>
    <NavLink to="/events">Events</NavLink>
    <NavLink to="/products">Products</NavLink>
    <NavLink to="/contact">Contact us</NavLink>
  </nav>
);

export const AboutMenu = () => (
  <div className="about-menu">
    <li><NavLink to="/about">Company</NavLink></li>
    <li><NavLink to="/about/history">History</NavLink></li>
    <li><NavLink to="/about/services">Services</NavLink></li>
    <li><NavLink to="/about/locations">Locations</NavLink></li>
  </div>
);
```

메인 메뉴와 서브 메뉴를 컴포넌트로 구성
(<MainMenu />, <AboutMenu />)

MainMenu: Type 2,3 공통



<NavLink />

- 현재 URL과 클릭 후 이동할 URL이 같은 경우
style 속성 추가 가능 (현재 활성화 된 아이템에 표시)

boolean: Props.isActive

<https://reactrouter.com/en/main/components/nav-link>

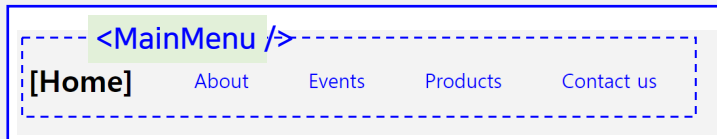
예제코드(ch11/proj/01-8: Pages.js - <PageTemplate />, <Home/>, <About /> 및 기타)

```
/* ch11/proj/01-8/src/components/Pages.js */
import { Link, Outlet } from "react-router-dom";
import { MainMenu, AboutMenu } from '../Menus';
import '../style/style.css'
```

```
const PageTemplate = ({ children }) => (
  <div className="page">
    <MainMenu />
    {children}
  </div>
);
```

컴포넌트 정의: <PageTemplate />
- 메인 메뉴를 상단 고정
- 하위 컴포넌트를 children에 렌더링

```
export const Home = () => (
  <div className="home">
    <h2 className="logo">[Home]</h2>
    <nav className="top-menu-nav">
      <ul className="top-menu">
        <li><Link to="about">About</Link></li>
        <li><Link to="events">Events</Link></li>
        <li><Link to="products">Products</Link></li>
        <li><Link to="contact">Contact us</Link></li>
      </ul>
    </nav>
  </div>
);
```



```
export const About = () => (
  <PageTemplate>
    <section className="about">
      <AboutMenu />
      <Outlet />
    </section>
  </PageTemplate>
);
```

<About />:(Type 2)
- 메인메뉴/서브메뉴 모두 포함
- 컴포넌트 내용은 <Outlet />으로

```
export const Events = () => (
  <PageTemplate>
    <section className="events">
      <h1>[Events]</h1>
      <p>This area is for events</p>
    </section>
  </PageTemplate>
);
export const Products = () => (...);
export const Contact = () => (...);
```

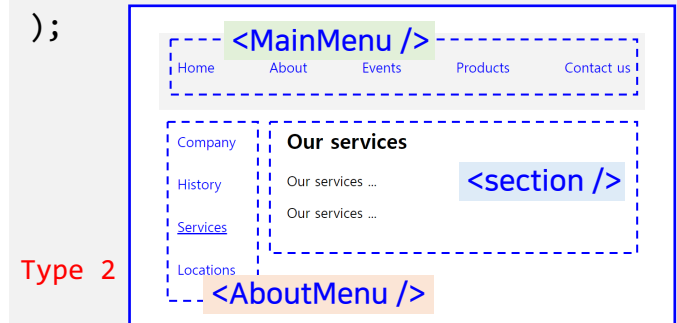
Event, Products, Contact (Type 3)
메인메뉴와 <section />만 포함



```
export const Company = () => (
  <section className="company">
    <h2>Our company</h2>
    <p>Our company ...</p>
    <p>Our company ...</p>
  </section>
);
```

About 하위 서브메뉴 컴포넌트 (Type 2)
- <section>...</section>에 담긴 내용은 <About />의 <Outlet />영역에 렌더링

```
export const Services = () => (...);
export const History = () => (...);
export const Locations = () => (...);
export const Whoops404 =
  ({ location }) => (
    <div className="whoops-404">
      <h1>{location.pathname}</h1>
    </div>
  );
```



학습 정리: 11장. 리액트 라우터

- Single Page Application
 - Client-side rendering
- React Router
 - `<Routes />`, `<Route />`로 컴포넌트 전환 (주소 직접 입력)
 - `useLocation()` hooks로 주소정보 반환
 - `<Link />`로 라우터 링크 제공
 - 중첩 Route와 `<Outlet />`, `<Navigate />`
 - `useRoutes()` hooks
- 동적 라우팅
 - 라우팅 파라미터
 - 쿼리
- React Context (6장)