

라우터는 뭐 할때 쓰는가?

여러 개 화면을 보여줄 때 사용합니다.

보통 여러분의 화면은 하나의 화면으로 구성되어있지 않다.

예를 들면, 많은 프로젝트에서 예시로 드는 TodoList 는 라우터가 필요없다.

<https://computationalcore.github.io/react-to-do-list/>

모바일로 위 사이트 접속해보면

이것도 필요할수는 있겠네요 탭이 세 개로 나뉘니깐

무슨 말이나면

만약 탭이 하나밖에 없었다고 생각해보자.

이 경우엔 라우터가 필요없어요

라우터가 필요한 경우는, 다른 화면일 경우 필요합니다.

조교도시락 피그마 보면,

1. 로딩 이미지 있죠

이 다음 뭐가있어요?

2. 로그인 화면 있죠

그 다음은?

3. 회원가입

근데 공통된 거 보이죠?

ㄱ. 푸터

필요하네요 밑에 달린거. 이건 별개로 빼겠습니다.

그 다음은?

4. 메인화면

또?

5. 주문화면

주문끝나면 주문완료 나와야죠?

6. 주문완료

그 다음 내 주문 확인하고 싶으면?

7. 나의 주문내역

수정하면 주문화면으로 돌아간다고 치고,

취소하면 다이얼로그 창 하나떠서 확인 받을거고

확인하면 뭐가 나와야되요?

8. 취소화면

나와야되잖아.

총 몇 개의 화면이 필요해? 여덟개의 완전히 새로운 화면이 필요하고
푸터는 상황마다 다르게 필요하니까 하나의 컴포넌트가 나오지.

그럼 이 화면은 어떻게 구분할까?
유알엘로 구분합니다.

예를들면,

<https://lunchbox.com/main>

이러면 메인페이지가 나오는거야.

<https://lunchbox.com/login>

이러면 로그인 나오는거지.

이런 식으로, 총 여덟개의 새로운 화면을 만들겁니다.

라우트는 이렇게, 유알엘로 서로 다른 화면이 나오게하는 것이고
서로다른 분기점을 라우터가 아닌 라우트 라고 합니다.

그러면 총 여덟개의 라우트가 필요할까?
그건 아닙니다. 라우트는 클릭이 일어나는 단위로 구분합니다.
라우터 연습하면서 알게될거예요

클릭을 했을 때 넘어가는 가장 쉬운 예가 뭘까?
네비게이션이 있죠
페이스북 페이지 가보면
탭이 있고
클릭하면 화면 바뀌면서 유알엘도 같이 바뀌는 걸 볼 수 있어요
조교도시락 피그마엔 네비게이션이 없는데?
그렇다고 못 만드는데는 아닙니다.
클릭이라는 기준만 가지고 있으면 만들 수 있어요.

일단, 본격적인 프로젝트 전에 라우터 연습부터 해보고
우리의 프로젝트의 라우터와 컴포넌트 설계도를 그려보고나서
우리의 프로젝트 코딩 시작하도록 하겠습니다.
프로젝트할때는, 코딩보다 코딩 직전까지의 작업이 가장 중요합니다.
이걸 무시하고 그냥 브이에스코드 켜서 코딩 시작하면 안되요

설계도 안해놓고 망치들고 건물짓지 말라는거예요
설계부터 제대로 합시다.

일단 라우터 연습부터 해보죠

먼저 프로젝트 디렉터리부터 만듭시다.

```
$ pwd
```

내 위치가 WSL 우분투에 있는 게 좋습니다.

어떻게 알 수 있나요?

C 나 D 로 시작하면 윈도우고

지금은 루트로 시작해서 홈의 유저이름안에 들어왔죠?

프로젝트할때는 우분투의 유저 디렉터리 안에서 작업하시는게 좋아요

여기에서, create-react-app 이라는 걸 써볼겁니다.

리엑트는 하나의 라이브러리라서, 여러분들이 create-react-app을 사용하지 않으면

귀찮은 환경설정을 일일이 다 해줘야되요

create-react-app 패키지는 귀찮은 환경설정을 제외한

하나의 샘플 프로젝트를 만들어줍니다.

마치, 코드샌드박스에 리엑트 템플릿 켜면

자동으로 당장 코딩 시작할 수 있는 환경 만들어주는것과 같아요

```
$ npx create-react-app 원하는프로젝트이름 --template typescript
```

저는 프로젝트 이름을 router-test 라고 할게요

그럼 엄청나게 많은 명령어가 출력되면서

해피해킹 문구 나왔죠? 성공입니다.

```
$ ls
```

하면, router-test 라는 디렉터리가 생겼죠? 들어가봅시다.

```
$ cd router-test
```

그리고 이걸 vscode 로 열거예요.

현재 디렉터리를 열고싶으면 다음 명령어를 칩니다.

```
$ code .
```

여기서 점은 현재 위치를 의미합니다. 엔터쳐보죠

그러면 브이에스 코드가 켜질겁니다. 프로젝트 디렉터리 한번 살펴보죠

...

우리가 코드샌드박스에서 샌드박스 하나 만들었을때보다 많은 것들이 자동으로 생성되었습니다.

한번 실행해볼까요?

브이에스 코드에서는 터미널도 실행 가능합니다.

ctrl + 백틱 눌러보죠

터미널 열렸구요, 현재 위치가 아니네요. 이동해줍시다.

...

그리고 노드제이에스 서버를 동작시킬거예요.

여러분의 컴퓨터에서, 자바스크립트로 작동되는 서버를 동작시킬겁니다.

```
$ npm start
```

하면 좀 시간이 지났다가 자동으로 새창이 뜨는데

WSL 우분투 쓰는 경우엔 다음과 같이 아무것도 안 뜰수가 있어요

이 경우엔 다음과 같이 하면 됩니다.

로컬호스트 밑에 172 아이피로 시작하는거있죠?

컨트롤키 누른채로 마우스 클릭해봅시다.

다음과 같이 정상적으로 작동되네요

에프십이 눌러서 개발자도구 켜주고, 콘솔에 선택해주도록 하겠습니다.

여러분이 코드를 수정하고 저장하면, 코드샌드박스에서 했던것처럼 자동으로 반영될겁니다

해볼까요? 에스알씨 - 앱닷티에스엑스

뱅크뱅크 돌아가는 리액트 아래의 문구를 바꿔보죠

조교행님의 리액트

그러면 다음과 같이 바뀐것을 볼 수 있습니다.

이제 좀 쓸모없는것들을 지워주도록 할게요.

우리는 딱 기본적인 것들에서만 시작할거거든요

.....

그리고 항상 새로고침할때는 개발자모드 켜 상태에서
새로고침 버튼을 오른쪽마우스 클릭하고
캐시 비우기 및 강력 새로고침을 해줘야합니다.
캐시에 대한 설명은 지금 당장 하지는 않을게요

그리고, 브이에스코드 추천하는 익스텐션 봅시다.
익스텐션이라는게 뭐냐, 한국어 브이에스코드 쓰시는분들은 확장이라고도 하는데,
브이에스코드에서 기본적으로 제공하지 않는, 다른사람이 만들어놓아서 브이에스코드에 설
치할 수 있는 프로그램을 말합니다.
테트리스같이생긴 버튼 클릭해볼게요
둘로 나뉘져있을거예요 로컬은 윈도우즈 브이에스코드에 깔린것들
그리고 더블유에스엘 우분투에 깔린 것들이 따로 나옵니다.
즉, 같은 이름의 익스텐션이라도 둘 다 깔아줘야 작동되는것들이 있어요
예를들면 퓨리티어 같은것들

일단은 더블유에스엘쪽에만 깔아주도록 할게요
만약 더블유에스엘에 안깔리는건 자동으로 로컬에 깔릴거예요

vscode-icons

이건 여러분들의 아이콘을 예쁘게 바꿔줄겁니다.

eslint

린트는, 코딩의 맞춤법검사기같은겁니다. 여러분들이 실수하는것들에 잔소리를 해줄거예요

prettier

이건 코드 포매터입니다. 여러분이 스페이스나 엔터, 탭을 예쁘지않게 작성했을때 예쁘게
바꿔주는 역할을 합니다. 단축키는 알트 시프트 에프

typescript extension pack

타입스크립트 쓰는 사람들이 깔면 좋을만한 익스텐션 모음입니다.

이것 정도만 깔아주도록 할게요.

다음 영상에서부터 본격적으로 라우터 연습해보도록 하겠습니다.

글자가 작게 보인다는 얘기가 있어서 좀 키워드릴게요

일단 src 아래에 디렉터리 두 개를 만들거예요

하나를 routes 라우트 복수 에스 라고 이름짓고

하나를 components 복수 에스 라고 이름짓겠습니다.

라우츠에는 화면이 되면서 유알엘을 가지는 것들이 들어갈 것이고,

컴포넌츠에는 화면이 될 수는 없는 자주 쓰이는 것들을 모아놓겠습니다.

맨 첫 화면을 먼저 만들어볼게요

Home.tsx 라고 이름짓고

원래 앱에 있던것들을 싹 긁어서

홈에 붙여넣고 이름을 App 에서 Home 으로 수정하고

에이치원 홈이라고 쓰고

조교행님의 메인페이지 라고 쓰겠습니다.

App.tsx는 빈 함수컴포넌트를 만들어서

홈을 임포트하고

태그를 붙여서 렌더링하겠습니다.

이게 우리의 첫번째 화면이에요

이제 앱 컴포넌트는 라우터를 담당하는 역할만 할겁니다.

여기서 스테이트 데이터도 관리하나요?

아니요 이제 안할거예요

라우터를 쓰면서, 앱은 오로지 라우터만 관리할겁니다.

우리가 할 건,

하나의 네비게이션에 두개의 버튼을 다는거예요.

하나를 Home, 다른 하나는 About

홈을 누르면 맨 처음 화면이 나오고,

어바웃을 누르면 조교행님 소개 페이지가 나온다고 해보죠

네비게이션은 하나의 화면이 아니니깐 컴포넌츠 디렉터리 안에 작성하겠습니다.

Nav.tsx 파일 하나 만들고,

빈 함수 컴포넌트 하나 만들어줍니다.

그리고

ul.navList>li*2 탭

엘아이 태그 안에 홈과 어바웃이라고 써놓읍시다

그리고 어바웃 컴포넌트도 만들겁니다.

에이치원 어바웃 쓰고

나이는 28, 직업은 조교

이렇게 써놓을게요

그리고 앱에서 네브 컴포넌트와 어바웃 라우트를 임포트할게요

그리고 Nav 태그와 어바웃태그를 써서 화면에 보이도록 하겠습니다.

잘 보이죠?

제가 뭐 한다고 했어요?

홈과 어바웃을 클릭하면 위에 유알엘이 바뀌면서

화면이 바뀌었으면 한다고했죠

이제 리액트 라우터 돔이라는 것을 쓸겁니다.

리액트와는 다른 패키지이니깐 설치를 해줘야해요

터미널에서 `npm install react-router-dom`

임포트해봅시다

임포트 세미콜론 찍고 리액트 라우터 돔부터 먼저 써줄게요.

원래 프롬 해서 가져와야되는데 이런식으로

가져오고자 하는 패키지를 먼저 써서 잘 가져와지는지 확인해야되요

근데 에러뜨죠? 그런 패키지 없다는 뜻인데

분명 설치를 했잖아요? 왜 인식을 못할까?

우린 타입스크립트를 쓰고 있기 때문입니다.

리액트 라우터 돔은 타입스크립트를 쓰지 않기 때문에 인식을 못하는거예요

그래서 번역기가 필요합니다.

가끔 설치할 때 이런 경우가 있어요

그래서 하나 더 깔아주도록 할건데,

보통은 이런 식으로 시작합니다.

@types/

여기에서 쓰고싶은 패키지 입력해주면 되는데

하기전에 엔피엠에서 진짜 있는지 찾아봅시다.

진짜 있죠? 버전은 오점 일점 팔

다운로드 수가 백팔십만으로 준수하죠?

이 맨 앞에 있는 버전이 원래 패키지하고 맞아떨어지는데

디펜던시 보면

오 버전 맞죠? 설치해주면 되겠네요

`npm install @types/react-router-dom`

하면 설치가 되면서

에러로 밀쳐진게 사라졌을거예요

이제 패키지를 정상적으로 가져올 수 있다는 뜻입니다.

뭘 가져올것이나

두 개를 가져올거예요

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

하나는 라우터를 말하고, 다른 하나는 라우트를 말합니다.

브라우저 라우터 이름 길면, 별명을 지어줄 수 있어요

`as Router`

라고 하면, 라우터라는 이름으로 우리 프로젝트에서 사용할겁니다.

앱에서는 라우터만 관리할것이라고 했죠?

그래서, 프래그먼트 바로 아래에 라우터라고 써주고, 모든 것을 라우터 안에 두겠습니다.
이제 홈과 어바웃을 지워버리고, 다음과 같이 두 개의 라우트를 네비게이션 밑에 둘게요

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

라우트 안에는 항상 두 개의 프로퍼티가 필요한데
패스와 컴포넌트입니다.

자, 무슨말이냐, 유알엘 보세요

맨 처음 로컬호스트 삼천번으로 들어가면 홈 컴포넌트가 나오고

뒤에 어바웃이라고 쓰면 어바웃 컴포넌트가 나오라 이말입니다.

근데 좀 이상해. 어바웃이라고 쓰면 어바웃만 나와야지 왜 홈까지 같이 있지?

이건 앞에 있는 슬래시때문에 그래요.

홈에도 슬래시가 있고, 어바웃에도 앞에 슬래시가 있잖아요

그래서 두 개가 같이 나오는거예요

만약 홈만 보여주고 싶다 하면 뭘 넣어야할까?

홈에 exact 라고 추가합시다

그러면 홈만 보여질겁니다.

딱, 그 주소에 해당될때만 보여주라는거예요

이그젝트를 붙일지 말지는 개발자의 상황에 따라 결정해야합니다.

홈 밑에 어바웃을 나오게하고싶을수도 있잖아요.

코딩하면서 적절한 상황에 활용해보세요

근데 네비게이션 부분 보면, 아직 텍스트일 뿐이고,

눌렀을 때 링크로 안 빠지죠?

이제 링크 달아주도록 할겁니다.

네브 컴포넌트로 갑시다

이건 왜 라우트가 아니겠어요? 하나의 화면 자격이 없어요

그래서 컴포넌츠 안에 있는겁니다

들어가서, 역시 리액트 라우터 돔을 임포트할건데

뭘 포함시켜줄거냐, 링크라는 놈을 포함시킬거예요

대문자로 시작하죠? 에이치티엠엘의 링크하곤 다른겁니다.

그리고 홈과 어바웃을 링크로 감싸주도록 하겠습니다.

알트하고 마우스 클릭해서

링크로 감싸주고

중요한 프로퍼티 하나 있어요

투 입니다.

to="/"

그리고 어바웃에는

to="/about"

확인해보면 링크가 생겼고

클릭하면 화면이 바뀌고 유알엘도 바뀌죠?

즉, 링크를 ‘클릭’ 하면 앱에 있는 라우터의 해당 주소로 향하는겁니다.
이게 핵심입니다. 라우터를 변경시키는 핵심은 뭐다?
클릭이 이루어져야되는겁니다.
여기까지가 라우터의 기본이고,
다음 영상부터 라우터의 심화내용을 알아보도록 하죠.

그 다음에 해볼 건, 데이터 전달입니다.
프롭스를 사용할 것이고
두 개의 라우트를 만들어서
링크를 누르면 연결된 라우트로
데이터를 보내버릴거예요
일단 두 개의 라우트 만들어야겠죠?
라우터프롭테스트 닷 티에스엑스
라우터프롭테스트 리저트 닷 티에스엑스
그리고 둘 다 텅 빈 함수 컴포넌트로 세팅하고
앱에다가 이 두개를 임포트해서
라우트를 등록할게요
네비게이션에도 링크를 달건데, 하나만 달겠습니다.
결과를 보여주는 페이지는 클릭하면 넘어갈테니깐
라우터프롭테스트 하나만 달게요.
클릭을 할 때마다 화면은 바뀌고
유알엘도 바뀔겁니다.
우리가 할 일은, 라우터프롭테스트에서
라우터프롭테스트 리저트 닷 티에스엑스로
데이터를 보내는거예요
일단 라우터프롭테스트에 데이터가 있어야겠네요
콘스트 메뉴 라고 쓰고
네임 짜장면
프라이스 5,000원
이라고 쓰겠습니다.
화면 바뀌는 기준은 뭐라고 그랬어요?
클릭이죠. 클릭이 일어나려면?
링크가 있어야겠죠. 이 링크에 투를 써주고 전달하고싶은 주소 쓰겠습니다.
그리고 링크는 “데이터 보내버리기” 라고 쓸게요
그리고 링크에 이제 패스 말고도 데이터를 전달하는 부분을 추가할건데,
이렇게 쓰면 됩니다.
to={{} } 괄호 두개 치고

첫번째 프로퍼티는 패스네임 하고 경로를 써주고

두번째 프로퍼티는 스테이트라고 쓸 것이고,

```
state: {  
  menu  
}
```

라고 써줄게요

여기서 스테이트는, 리액트 라우터 돔에서 정해놓은 이름입니다.

우리가 맘대로 바꿀수는 없어요

이 안에 우리가 보내고싶은 데이터를 넣어주는거예요

그러면 주소가 정상적으로 연결되어있으니깐

해당 주소에 연결되어있는 라우트에 데이터가 전달되어야겠네요?

잘 갔는지 확인해봅시다. 콘솔로그 찍어보죠

어떻게 확인할까? 프롭스 확인하듯이 하면됩니다.

파라미터에 프롭스라고 쓰고

콘솔로그로 찍어내고싶은데

우린 타입스크립트 쓰기때문에 인터페이스 써야되죠?

근데 지금 당장은 데이터타입을 알기가 힘드니깐

먼저 애니로 쓰세요

애니는 어떤 타입이든지 다 받겠다는뜻입니다.

콘솔로그로 데이터 확인하고

인터페이스 작성해서 인터페이스 타입으로 바꿀거예요

확인해보면

로케이션 안에

스테이트 안에

샘플데이터가 있고 그 안에보면 우리가 보낸 데이터가 있어요

그럼 어떻게 인터페이스 쓰면 될까?

```
interface propTypes {  
  location: {  
    state: {  
      menu: {  
        name: string,  
        price: string  
      }  
    }  
  }  
}
```

이렇게 쓰고 애니타입을 없애버리고

프롭타입스라고 쓰면 됩니다.

인터페이스 어떻게 작성해야되는지 스트레스받지말고

일단 애니로 받아서 콘솔로그 찍어본다음에
바꾸면 됩니다.

그럼 잘 받아왔으니깐 붙이면 되겠네요

일단 콘솔로그를 바꿔줄게요

const menu 라고 선언하고, props.location.state.menu
를 집어넣습니다.

그리고 디브태그 두 개를 달아서,

```
{menu.name}
```

```
{menu.price}
```

써주면,

테스트해보죠

처음으로 돌아간다음

라우터프롭테스트 클릭하고

아무것도 없죠?

유알엘보면 우리 위치도 라우터프롭테스트

클릭하면 화면바뀌고 짜장면 오천원

유알엘이 바뀌었죠? 뒤에 리저트가 붙어서, 라우터프롭테스트리저트로 이동했어요
이런식으로 데이터 전달을 합니다.

근데 개발하다보면 상황에 따라서,

데이터를 직접 전달하지 않고

‘어떤 데이터를 가져오는지’에 대한 정보만 넘겨준다음

데이터소스에 접근해 ‘알아서’ 가져오게 할수도 있어요

그건 다음 시간에 알아보죠

먼저 샘플데이터를 만들겠습니다. src 아래에 제이슨파일을 만들거구요, 데이터닷컴제이슨이
라고 합니다.

그리고 제이슨은 이렇게 만들거예요

```
{  
  "data": {  
    "students": [  
      {  
        "id": 0,  
        "name": "조교행님",  
        "age": 28,  
        "isGirlfriend": true  
      },  
      {
```

```

      "id": 1,
      "name": "지수",
      "age": 22,
      "isGirlfriend": true
    },
    {
      "id": 2,
      "name": "펑수",
      "age": 5,
      "isGirlfriend": false
    }
  ]
}
}

```

보시면, 중괄호로 시작했고 그 안에 데이터가 있고
 그 안에 스튜던츠인데 배열로 들어올 겁니다
 배열 엘리먼트는 총 세개, 각각의 배열은 객체네요.
 객체 안엔 아이디, 네임, 에이지, 이즈걸프렌드가 있어요
 여기서 우리 이즈걸프렌드는 안 쓸 겁니다.
 일단 이런식의 샘플데이터를 만들고, 백엔드로부터 들어온다고 가정할게요
 그리고 StudentList.tsx 라는 라우트 하나 만들고
 빈 함수 컴포넌트로 두겠습니다.
 앱에다가 импорт할게요 그리고 라우트 하나 달겠습니다.
 패스는 studentlist 컴포넌트는 {StudentList}
 그리고 네비게이션에 등록해주겠습니다.
 링크 두 스튜던트리스트, 링크 안에는 스튜던트 리스트라고 쓸게요
 클릭을 하면 화면 변경되죠?
 이제 스튜던트리스트 라우트에서, 샘플데이터를 импорт할게요
`import { data } from "../data.json";`
 그리고 콘솔로그로 찍어볼게요
 보니까, 우리가 필요한건 스튜던츠라서
 스튜던츠만 가져오도록 하죠
`const students = data.students;`
 그리고 반복해야되니까 맵함수 씁시다.
`<div>`

```

    {students.map((student) => (
      <h4>
        <a>
          {student.name}
        </a>

```

```
    </h4>
  )})
</div>
```

조교행님 지수 펍수 잘 뜨죠?

근데 워닝 떴네요. 키값이 없어서 그렇죠?

h4 안에 키 프로퍼티를 달고,

key={student.id} 로 정해서 각 요소를 유일하게 만들겠습니다.

제가 원하는건, 눌렀을 때 해당 학생의 디테일한 정보가 나오게하는것인데

데이터 자체를 프롭스로 전달하는게 아니라

힌트만 전달하고싶다는겁니다.

일단 링크 사용해야되니깐

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

그리고 에이태그를 링크로 바꿔주겠습니다.

근데 어디로 가야할지를 아직 안 만들었죠?

StudentDetail.tsx 라는걸 하나 따로 만들게요

앱에 라우트 등록하는데, 경로를 /studentlist/detail 로 하겠습니다.

이 경로는 스튜던트리스트 라우트에도 등록을 해줄게요.

네비게이션엔 등록하지 않습니다.

왜냐, 이 라우트는 오로지 스튜던트 리스트에 들어가서 클릭을 할때만 보여지거든요

그리고 화면에서 적용이 잘 되었는지 볼까요?

클릭하면, 넘어갑니다. 근데 스튜던트 리스트는 그대로 남아있죠?

왜 그래요? 앱으로 돌아가서 보면

스튜던트리스트 까지 둘 다 똑같고

디테일은 하위항목이잖아요. 만약 이게 싫으면 어떻게 하라했어요?

이그젝트 붙이면 된다고했죠.

클릭 클릭 클릭 클릭

이제 무엇을 할것이냐, 각 학생의 이름을 누르면, 이름과 나이가 보이도록 만들것인데

이름과 나이 그 데이터 자체를 넘겨주는연습은 우린 이미 지난 영상에서 했고,

그렇게 하지 않고, 접근할 데이터의 인덱스만 넘겨주도록 하겠습니다.

스튜던트리스트 라우트에서 to 에서, 스튜던트리스트 뒤에 뭘 더 추가해줄건데

백틱을 써보도록하죠. 백틱으로 감싸줍시다.

왜 백틱을 썼을까? 변수를 집어넣을거거든요

```
<Link to={`/${studentlist}/${student.id}`}>{student.name}</Link>
```

달려표시하고 중괄호로 감싸고, 스튜던트아이디를 넣읍시다.

이거 왜 이렇게 쓰는지 모르면 저의 언어강의인 자바스크립트, 타입스크립트의 백틱부분을 복습하셔야됩니다.

그럼 앱닷컴에스엑스도 바꿔줘야겠죠?

여기선 백틱을 쓰지 않고, 디테일 지우고 다음과 같이 쓸겁니다.

```
/:id
```

이건, 받아들이는거예요. 왜 이렇게 되는지가 아니라 리액트라우터돔에서 이렇게 하라고 했

어요.

디테일 라우트에선 어떻게 썼었죠?

확인해보면 달려 쓰고 원하는 인덱스를 가져왔고,

앱닷티에스엑스엔 어떻게했어요?

콜론 붙이고 아이디라고 썼다구요.

이름을 꼭 아이디로 해야되니까?

상황에 따라 여러분 원하시는대로 지어도 되겠죠.

그리고, 저장하고,

다시 학생 리스트에서 클릭을 하면 어디로 갈까?

디테일을 출력하는건 똑같은데 유알엘 잘 봅시다

제가 누르는 학생 번호에 맞춰서 유알엘이 바뀌죠?

즉, 우리가 만든 스튜던트 디테일 라우트는 하나인데,

유알엘 뒤에 붙는 인덱스에 따라서 서로 다른 화면을 보여주도록 할거라구요.

그러면 저 유알엘에 있는 인덱스를 가져오면 되겠네요?

어떻게 가져올까. 프롭스로 가져올겁니다.

스튜던트 디테일 라우트에서

프롭스를 쓰고 애니 써서 콘솔로그 찍어보죠.

다음과 같은 것들이 있네요

매치 들어가서 파람스 들어가면

짜잔. 아이디가 있잖아요.

그럼 이것만 뽑아내봅시다.

프롭스 전부 다 가져오기싫고 매치만 가져오고 싶으면

{match} 라고 쓰면 됩니다. 인터페이스 써줍시다.

```
interface propTypes {  
  match: {  
    params: {  
      id: any;  
    };  
  };  
}
```

그리고 애니 떼버리고 프롭타입스

그리고

```
const id = match.params.id;
```

```
console.log(id);
```

하면 아이디 잘 뜨죠.

이제 우린 아이디를 알고있으니깐

데이터소스에 직접 접근해서 원하는걸 가져올수있겠네?

여기도 제이슨 импорт해줍시다.

```
import { data } from "../data.json";
```

그리고 어떻게 쓰면 되겠어요?

```
const name = data.students[id].name;
받은 아이디를 인덱스로 사용했다구요
console.log(name);
이름 잘 나오죠?
우린 나이도 필요하니까
const age = data.students[id].age;
console.log(age);
```

잘 나옵니다. 이제 콘솔로그 다 지워버리고, 가져다붙이면 되겠네요
div 태그 두개 쓰고
이름은 {name}
나이는 {age}
하면 잘 뜹니다.
이번 영상에선 뭘 보여준거예요?
저번에는 데이터를 직접 넘겨줬는데,
이번엔 힌트만 줘서 받는 라우트에서 데이터소스에 직접 접근해서 가져왔죠.
여러분이 코딩하는 상황에 따라서, 둘을 적절히 활용하면 됩니다.

라우터 이론배운다고 생각보다 시간을 많이 썼는데,
이제 진짜 우리 프로젝트로 가봅시다.
조교도시락 만든다고했죠?
근데 코딩시작하기전에 할 일이 하나 더 있어요
우린 유아이 만들었고, 이알디 짰고, 샘플데이터까지 넣었지만,
라우트-컴포넌트 계통 및 props 흐름을 그림으로 정리하진 않았어요
아니 언제코딩할겁니까? 코딩 안할거예요? 묻는사람들 있는데
이 과정까지 마치면 여러분들 전체 코딩하는 삶이 편해질겁니다.
왜? 설계도가 이미 있기때문에,
코딩은 설계도만 따라가면 된다구요.
여러분들이 이 설계를 계율리하고 키보드에 코드부터 두들기면
나중에 구조 자체가 개판이되요.
데이터 흐름이 엉망이에요.
그래서 그걸 미리 정해놓고
코딩하다가 뭘 해야될지 모르겠으면
그 설계도 보면서 작업하라구요.
아시겠습니까?
우선 피그마를 두 개 켜게요.

첫번째 화면, 조교도시락 로딩화면이죠

이 화면의 경우, 자동으로 넘어가기때문에 클릭이 일어나지 않아요

그래서 로그인 컴포넌트안에 이 로딩 페이지도 있어야되는겁니다

왜? 라우트 이동은 링크를 클릭함으로써 이루어지는데 여긴 클릭이 아무것도 없잖아요.

근데 궁금한사람 있을수있어요

말도안돼. 세상에 이벤트가 클릭만 있는게 아닌데

꼭 링크버튼을 달아야 라우트가 바뀐다고?

물론 다른 방법 있어요. 히스토리에 푸시를 한다던가

근데 안 가르칠거예요.

제가 지금 진행하는 프로젝트는 일부러

여지를 상당히 남겨두는 미니프로젝트로 만들겁니다.

나머지 궁금한거,

여러분들이 구글링하고, 찾아보고, 고민해보고 만들어봐야되는겁니다.

일부러 많이 부족한 실전을 해보는거예요.

알겠죠? 우리가 할 수 있는것만 일단 활용합시다.

어쨌든, 첫번째 라우트는 뭐라고? 로그인

다른 피그마에 등록해줍시다.

회원가입은 로그인창에서 클릭을 해야 넘어가죠?

그러니깐 라우트가 되는겁니다.

두번째 라우트 뭐라고? 회원가입.

그리고 컴포넌트를 두 개 달건데, 우리 앱 홍보하는 로딩화면을 담당하는

로딩 컴포넌트가 있고

그리고 진짜 로그인 화면을 담당하는 로그인컴포넌트가 있어요

그리고 로그인이든 회원가입이든 클릭을해서, 성공하면 뭐가뚝니까?

메인페이지 뜨죠.

세번째 라우트 뭐라고? 메인.

그리고 주문하기 버튼 클릭하면 주문화면이 뜨겠죠?

오더라고 영어로 이름지으면

네번째 라우트 뭐라고? 오더.

그리고 오더에서 주문하기 클릭하면, 내일봐요 화면 뜨죠?

오더 컴플리트라고 합시다.

근데 이 경우에는 문제점이 있어. 주문완료에서 메인화면으로 자동으로 넘어가기때문에

클릭을 못합니다. 그래서, 이건 라우트로 못 빼요. 컴포넌트로 뺄겁니다. 라우트로는 실격

근데 주문 완료하면 내일봐요 화면 뜨면서 잠시 시간이 지났다가 메인화면으로 간다고 했잖

아요.

그러면 이 내일봐요, 즉 오더컴플릿트 컴포넌트를 주문화면에 두는게 맞을까 메인화면에 두는게 맞을까

여러분이 링크 클릭뿐만 아니라, 라우터의 다른 기능을 이용해서 이 내일봐요를 독립된 라우트로 뺄 수 있다면 그렇게 하시면되는데 지금은 그렇게 하는 법 모르니깐, 일단 메인화면 아래에다가 내일봐요를 두겠습니다.

클릭이 일어난 시점에서 라우트를 탈출한다고 가정하면 메인 아래에 두는게 맞겠죠.

그리고 메인화면에서 마이 버튼 클릭하면 나의 주문내역 나오죠
마이오더라고 이름붙일게요
다섯번째 라우트 뭐라고? 마이오더

그리고 수정하기 누르면 주문화면으로 넘어갈것이고, 이것은 이미 라우트가 있습니다.
취소하기 누르면? 취소되었습니다 유유 로 넘어가죠
근데 이것도 자동으로 메인화면으로 빠질거예요. 버튼이나 링크가 없으니깐.
그래서 라우트가 아니라 컴포넌트로 뺄거예요
라우트로는 실격.
이 컴포넌트 역시 메인화면 아래에 따로 두도록 하겠습니다.

이제 라우트와 컴포넌트 색깔을 구분해주도록 하겠습니다.

총 몇개? 다섯개 라우트가 필요하고

컴포넌트는 총 몇개? 실격된거 두개에 푸터까지 해서
지금 당장은 총 세 개가 필요하네요.
라우트 밑에 세부적으로 컴포넌트를 만들 수 있겠죠?
특히 메인화면은, 헤더와 콘텐츠, 푸터로 나눌 수 있겠네요.
나머지엔 푸터를 달아줄것이고.

이제 프롭스 파악해봅시다.

우선, 우리의 디비가 있겠죠.

따로 동떨어진곳에 두도록 합시다.

로그인부터가 디비에 접근해서 일치하는지 확인하죠?

회원가입도 마찬가지로? 아이디와 일치하는지 확인하고 가입 진행할것이고

로그인 성공시에 메인화면에서도 각종 데이터를 디비로부터 가져올겁니다.

근데 메인은 세 개의 하위컴포넌트로 나뉘잖아요

다른건 몰라도, 콘텐츠 컴포넌트엔 우리 데이터 중에 메뉴에 있는 데이터를 프롭스로 내려 보내야겠네요.

그리고 주문하기 누르면 어디로 향합니까? 주문화면으로 갈건데

사용자가 어떤 걸 선택했는지 알아야하니깐

컨텐츠 컴포넌트에서 주문화면 라우트로 프롭스로 데이터가 전송되어야 할 것이고

주문하기 누르면 그 정보가 어디로 가겠어요?

디비로 가겠죠.

만약에, 아까 메인화면에서 마이 버튼을 누르면 내가 이제까지 주문한 내역을 보여주게 될 텐데

그것도 어디서 가져올까? 디비에서 가져오겠죠.

거기서 수정하기 누르면 지금 기록되어있는 데이터를 어디로 넘겨줄까요?

주문화면으로 프롭스 전달할것이고

취소하기 누르면 디비에 삭제 요청을 할 것입니다.

정리해서 그려보면 다음과 같은 설계도가 그려졌겠네요.

이렇게 만들어놓으면 뭐가 좋다? 길 잃을 걱정이 없다.

물론, 중간중간에 바뀌는 항목이 있다면 설계도도 수정될수 있지만

우리가 어디로 가야할지 확실하 알고있는것과

그렇지 않고 키보드부터 두드리면서 코딩 시작하는것은 천지차이입니다.

여러분은 전체 개발과정의 40% 를 마쳤다고 할 수 있는거예요.

다음시간부터, 프로젝트 디렉터리 만들고, 본격적인 실전 코딩 시작하겠습니다.